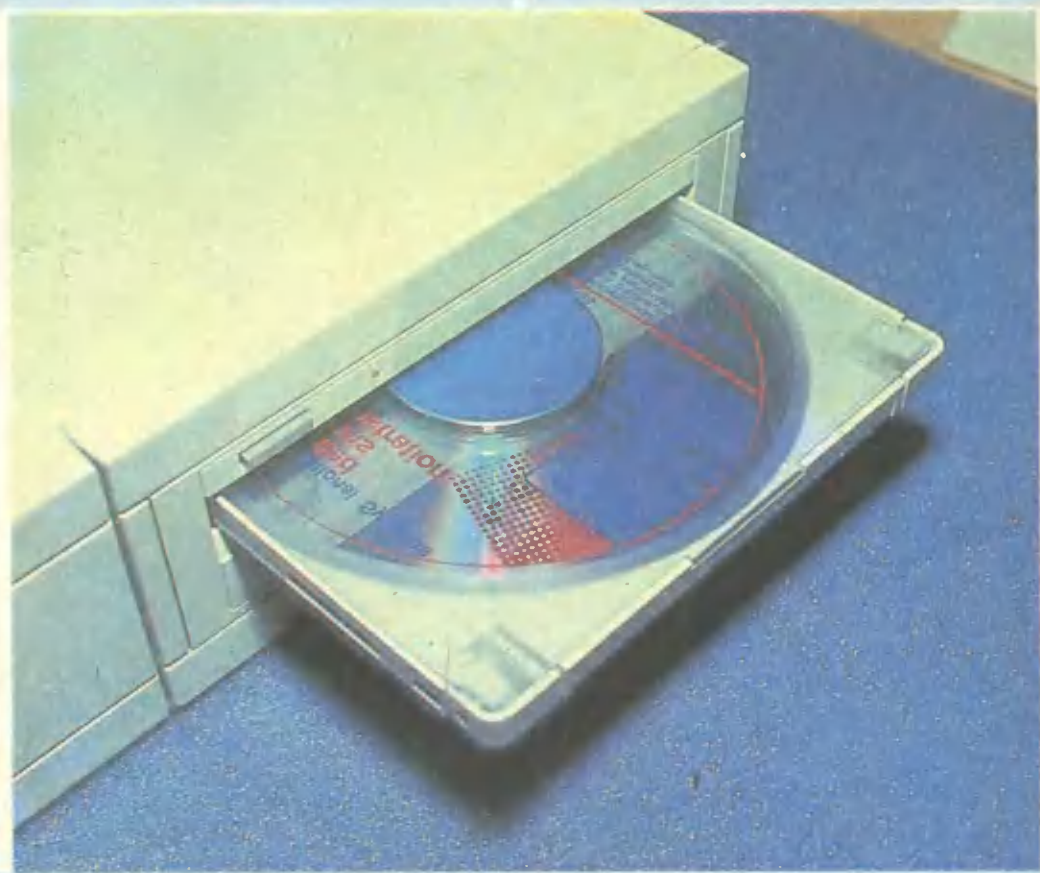


# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

5 1993



*Не будет ли мы, взрослые,  
гужими в их мире?*



*«...Что происходит с ребенком, который постоянно работает с компьютером?*

*Ежегодно в летней школе под г. Красноярском мы наблюдаем несколько учеников, которым не нужен футбол, которых невозможно «вытащить» на концерт, в театр, в кино. И их интересуют не компьютерные игры. Эти дети готовы программировать сутками, и им никто не нужен! Лично для меня — это сложная как в философском, так и в психологическом отношении проблема...».*

*Интервью с Первым заместителем Министра образования Российской Федерации Виктором Александровичем Болотовым читайте в рубрике «Общие вопросы».*

**ИНФ**  
**5'93**

# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

Научно-методический журнал  
Учрежден Министерством  
образования РФ  
и коллективом редакции

Издается с августа 1986 г.  
Выходит шесть раз в год

## РЕДАКЦИОННАЯ КОЛЛЕГИЯ

**БОЛОТОВ В. А.**  
**БРЕДИХИН Г. А.**  
**ВАСИЛЬЕВ Б. М.**  
(зам. гл. ред.)  
**ЗАЙДЕЛЬМАН Я.Н.**  
**ЗУБЧЕНКО А. А.**  
**КИСЕЛЕВ Б. Г.**  
**КОРОЛЕВ В. А.**  
**КРАВЦОВА А. Ю.**  
(первый зам. гл. ред.)  
**КРАСНОВ А. Я.**  
**КУЗЯКИН А. П.**  
**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.**

## СОДЕРЖАНИЕ

### ОБЩИЕ ВОПРОСЫ

Болотов В.А. Программы разные — стандарт общий 3

### МЕТОДИКА

Александров Г.Н. Программированное обучение и новые информационные технологии обучения 7  
Ерохина Е.А. От логики к программированию 25  
Юдина А.Г. Начала алгоритмизации и программирования в 9 классе 37  
Антипов И.Н., Боковнев О.А., Степанов М.Е. О преподавании информатики в младших классах 46  
Пронина С.Е. Информатика в начальной школе 51

### НОВЫЕ ПРОЕКТЫ

Роберт И.В. Виртуальная реальность 53

### ПЕДАГОГИЧЕСКИЙ ОПЫТ

Булитко В.В., Ермилов А.Л. Наблюдение числовых галактик 57  
Сафьянников А.И. «PRIVATE TUTORS» 60

### ЗАДАЧИ

Бадин Н.М., Волченков С.Г. Изучение параллельных алгоритмов 63

### ТОЧКА ЗРЕНИЯ

Щеголев А.Г. Для чего нам учебник? 67

## РЕДАКЦИЯ

*Отдел методики*

Луцкая Н.А.

*Отдел ВТ*

Усенков Д.Ю.

*Отдел рекламы*

Васильева Н.А.

*Отдел подписки и*

*распространения*

Виленкина Е.Н.

*Художественный редактор*

Иванова Т.В.

*Экономический отдел*

Бородаева З.В.

*Технический редактор*

Луговская Т.В.

*Корректор*

Мацюк Т.В.

## ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

Васильев Б.М., Частиков А.П. Микропроцессоры (история, применение, технология)	71
Буцкий Ю.И. Учебные программы интерполяции кубическими сплайнами	84
Гершензон В.Е., Зейналова Е.Ю., Кошевой В.А., Левченко И.В., Тараканова О.Н. Проект «ПОДСОЛНУХ» — образ новой обучающей среды	91
Платонов Н.Е. Средства мультимедиа в образовании	94
Очков В.Ф. Химеры программирования	98

## КЛУБ УКНЦ

Сулханов В.И. «Электроника БК-0010» и УКНЦ	105
Акулин А.И. УКНЦ: повышение эффективности работы	109

## КЛУБ БК

Котов Ю.В. Экономное программирование для БК-0010.01 и БК-0011М	111
Андронов И.Л. Ода Форту	113

## ЗАРУБЕЖНЫЙ ОПЫТ

Керр С. Новые информационные технологии и реформа школы	117
---	-----

## СПРАВОЧНЫЙ ЛИСТОК

О размерах доплат учителям информатики	123
О порядке аттестации	124

## ИНФОРМАЦИЯ

Принтеры производственного объединения «Радий»	126
--	-----

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

142300, Чехов, Московской обл.

Подписано в печать с оригинал-макета 10.02.94. Формат 70x100 1/16. Бумага офсетная. Печать офсетная. Усл.печ.л. 10,40. Усл.кр.-отт. 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.

# ВИКТОР БОЛОТОВ:

## «ПРОГРАММЫ РАЗНЫЕ — СТАНДАРТ ОБЩИЙ»

В ноябре 1993 г. состоялась коллегия Министерства образования РФ, на которой была обсуждена Программа информатизации образования Российской Федерации на 1994-1995 гг. Чем вызвано появление этого документа? Какие новые проекты разрабатываются в министерстве? Какие изменения происходят в системе образования в связи с ее информатизацией? На эти и другие вопросы зам. главного редактора журнала «Информатика и образование» А. Ю. Кравцовой отвечает Первый заместитель Министра образования Российской Федерации В. А. Болотов.

— Виктор Александрович, какие проблемы интересуют Вас в тех процессах, которые мы объединяем словами «информатизация образования»?

— Спор о словах — излюбленное занятие схоластов. Сегодня нет жесткой дефиниции для этих процессов, и это нормально. Ведь мы только входим в информационную эру. Активная работа с большими информационными потоками — удел в лучшем случае 5% населения, в том числе и в странах с развитыми технологиями. Ученые, бизнесмены, управленцы — да, а обычный человек отнюдь не является активным субъектом этих процессов. Он использует информационные технологии ровно настолько, насколько ему это нужно. Поэтому цель школьных уроков информатики, на мой взгляд, как раз и состоит в том, чтобы снять психологические барьеры во взаимодействии человека с компьютером. Человек должен понимать, что при определенных усилиях он способен использовать информационные технологии как в профессиональной, так и в повседневной деятельности. У детей это происходит гораздо быстрее.

Но одних уроков информатики для формирования устойчивых пользовательских навыков мало. Здесь, мне кажется, очень важно, чтобы и на других уроках: физики, химии, географии, истории и т.д. ученики могли бы использовать разнообразные программно-методические комплексы, созданные на основе этих технологий. В педагогике хорошо известен прием непрямого освоения материала. В данном случае, когда ученик изучает компьютер не как цель, а как средство для решения другой задачи, он незаметно для себя постигает его возможности, не изучая их специально. Пользовательские навыки на уровне автоматизма формируются более эффективно. Поэтому разработку компьютерной поддержки различных курсов я считаю актуальной и интересной проблемой.

Но лично для меня наиболее интересно то, что происходит с ребенком, который взаимодействует с компьютером длительное время.

Ежегодно в летней школе под Красноярском мы наблюдаем несколько учеников, которым не нужен футбол, которых невозможно «вытащить» на концерт, в театр, в кино. И их интересуют не компьютерные игры. Эти дети готовы программировать сутками, и им никто не нужен!

У Гегеля есть высказывание, смысл которого состоит в том, что когда человек учится писать, у него формируются определенные структуры мышления. Какие структуры

мышления образуются, какие изменения происходят в психике ребенка, если его развитие и обучение включает использование компьютера и современных интерактивных средств? Не получим ли мы, образно говоря, вместо «homo sapiens» «homo computericus»? И не будем ли мы, взрослые, чужими в их мире? Это скорее философские вопросы, но для меня они очень важны. А трудности технического характера будут преодолены как в аспекте аппаратного, так и программного обеспечения – это вопрос времени.

— *Вы говорили о личности ученика, а как, по Вашему мнению, влияют информационные технологии на систему образования в целом?*

— Еще в начале двадцатого века годы жизни профессии были сопоставимы с длиной человеческой жизни. В наше время сравнительно быстро отмирают одни профессии и возникают другие. Поэтому одна из задач, стоящих перед современной системой образования – развить и сформировать у человека такие способности, которые позволяли бы ему комфортно адаптироваться к быстро изменяющимся внешним условиям. Когда стали обсуждать, какие это способности, то лишний раз убедились в старом тезисе: человеку нельзя дать образование принудительно, можно только создать условия для обучения.

В такой постановке существующая классно-урочная система Я. А. Коменского уже мало кого устраивает. Ситуация: один учитель на 20-40 учеников, неизбежно заставляет его работать, ориентируясь на некоего «обобщенного среднего ученика». Я убежден, что формирование и развитие тех способностей, о которых мы говорили выше, может происходить только в интерактивной деятельности, когда и учитель, и ученик действуют совместно, идут навстречу друг другу, находятся в поисковой зоне, когда в учебный процесс включены элементы коллективной деятельности. Мне кажется, здесь компьютер незаменим. В частности, С. Пейперт в своих работах как раз и предлагает именно технологический инструментарий, который позволяет реализовывать подходы, связанные с развивающим обучением.

Использование информационных технологий в учебном процессе – стержень реформирования системы образования.

— *В нашем журнале были опубликованы материалы, связанные с Государственной научно-технической программой «Развитие образования в России» (направление III «Информатизация образования») на 1994 год. На коллегии Министерства образования РФ, состоявшейся 10 ноября 1993 года, была рассмотрена и одобрена Программа информатизации образования в Российской Федерации на 1994-1995 гг. В чем разница между этими программами?*

— Главное отличие – это документы разного предназначения. Цель первой программы: получение научных результатов по заранее определенным направлениям, представляющим на данный момент интерес в государственном масштабе. Назначение Программы информатизации: координация работы различных организаций, учреждений, органов управления образованием разного уровня в решении проблем, связанных с информатизацией образования. Это прежде всего «управленческая» программа, т. е. некая рамка с описанием процедуры механизмов согласования управленческих воздействий. Конечно, Программа информатизации во многом базируется на научных разработках, которые были выполнены в ходе реализации научной программы, но между ними есть и принципиальные отличия. В частности, по механизму финансирования: научная программа финансируется по принципу грантов, а Программа информатизации – по принципу целевых заказов.

– В рамках той или иной программы расходуются бюджетные средства. Авторский коллектив использует их для создания определенного продукта: учебника, программно-методического комплекса, научного отчета и т.д. Кто является собственником такого продукта, и на какой основе он должен распространяться?

– Я считаю, что вопрос о том, кто является собственником, в каждом конкретном случае, должен фиксироваться в договоре и в принципе может быть решен по-разному. Ведь очень часто министерство финансирует итоговую часть работы, которая изначально была сделана либо в инициативном порядке, либо на средства разработчиков или как-то иначе. И каждый раз это является предметом обсуждения. Интеллектуальная составляющая этого продукта, безусловно, должна стать достоянием всей системы образования, а вот его материальная оболочка (стоимость издания, если это учебник, стоимость дискет и т.п.) должна быть предметом купли-продажи. Например, если тот или иной продукт был создан целиком на государственные средства, то покупая его, нужно оплачивать только стоимость тиражирования. К сожалению, нередки случаи, когда за одну и ту же работу разработчик пытается получить дважды: от государства и от покупателя. Как правило, узнав об этом, мы исключаем такой коллектив или автора из числа партнеров.

– Какие функции в становлении современной системы образования России делегируются регионам, а какие остаются за министерством?

– Сначала давайте определимся, какие задачи нужно решать на федеральном уровне, а какие – на региональном. Например, систематизация и выработка перспективных направлений развития системы образования, определение содержания региональной компоненты образования, выбор альтернативных форм образовательных учреждений с учетом особенностей региона, численность кадров на местах, содержание и формы системы повышения квалификации – это прерогативы регионов. Хорошим примером в этом отношении является Пермская область, где скоординированная политика органов управления образованием и руководителей образовательных учреждений позволила успешно решить многие проблемы информатизации образования именно на региональном уровне. Прошедшая в Перми в декабре 1993 г. Республиканская научно-практическая конференция «Региональные проблемы информатизации образования» это отчетливо показала.

Укажу лишь некоторые проблемы, которые должны решаться на федеральном уровне.

Создание, усиление и поддержка единого информационного пространства в системе образования России. Свою роль в этом должна сыграть и Программа информатизации образования, так как информатизация образования – стратегическое направление создания и удержания этого пространства. В этом отношении научно-методический журнал «Информатика и образование» незаменим как орган, который позволяет такое пространство удерживать. Но здесь прежде всего должна быть разработана идеология, общее поле понимания, по крайней мере. Поэтому по решению коллегии мы организуем Координационный комитет по реализации Программы информатизации образования при Министерстве образования РФ, в который намерены пригласить людей, референтных для разных сообществ. При сложившемся плюрализме в преподавании информатики, альтернативных учебниках, сформировавшихся и зарекомендовавших себя авторских коллективах очень важно, чтобы на вопрос: «А судьи – кто?» – мы могли бы ответить, что судьи – референтные люди, признанные сообществом.

Разработка, согласование и удержание стандартов. Это, безусловно, проблема федеральных органов управления. Ее курирует со стороны министерства ведущий отдел информатизации образования (начальник отдела – Д.О. Смекалин).

Экспертиза и лицензирование программных средств для системы образования. Эта работа будет проводиться в институте информатизации образования МО РФ (директор института – Я.А. Ваграменко). Координационный комитет должен сформировать предварительные списки экспертов, которые затем будут обсуждаться и уточняться на разных уровнях.

Широкое внедрение в систему образования средств телекоммуникации – это ближайшая задача министерства. В связи с этим встает вопрос организации информационных потоков. Какие сети использовать, какая информация необходима тем или иным пользователям и т.д.? Мы начинаем такую работу на федеральном уровне и считаем ее очень актуальной.

Подготовка педагогических кадров – стратегический фактор удержания единого образовательного пространства России. Координация этого направления возложена на Учебно-методическое объединение педагогических учебных заведений Министерства образования Российской Федерации по информатизации педагогического образования (председатель УМО – М.П. Лапчик).

Я хотел бы, пользуясь случаем, обратиться через Ваш журнал к учителям информатики.

Сегодня очень важно не замыкаться в рамках кабинета информатики. Необходимо взять на себя роль миссионеров и идти к учителям-предметникам. Кооперация учителя информатики и учителя-предметника – это дальнейшее развитие информатики в школе.

Идите к управленцам. Для того, чтобы совершенствовать систему образования, нужно использовать информационные технологии и в сфере управления. Предлагайте управленческому аппарату решения их проблем, и тогда они, может быть, поймут ваши.

– *Какие проекты, в том числе совместные, планируются Министерством образования РФ?*

– Планы есть, предварительная проработка таких проектов ведется в отделе информатизации образования, но говорить о реализации какого-либо из них можно только после утверждения бюджета на 1994 год.

Инвестиции западных стран, в частности, в систему образования, в действительности строятся по одному принципу: «Мы даем вам кредиты при условии, что 80% этого кредита будет истрачено в нашей стране». Реально – это закупка вычислительной техники. Мы пока от этих проектов отказываемся. Речь идет о крупных суммах и не надо забывать о процентах.

Есть гранты, но они даются, как правило, на научные разработки. Западные фирмы готовы привлекать наших специалистов для разработки программного обеспечения, методик. Они выражают желание финансировать и адаптацию своих программ для наших школ.

Но надо отчетливо понимать, что проблемы системы образования мы не решим за счет Запада никоим образом. Поэтому Программа информатизации, о которой мы сегодня говорили, рассчитана на бюджетное финансирование. Это реальный шаг на пути к решению наших проблем собственными силами.



**Г. Н. Александров,**

д-р пед. наук, профессор,

заслуженный деятель науки СО АССР, БашАССР

г. Владикавказ, Северо-Осетинский государственный университет

## ПРОГРАММИРОВАННОЕ ОБУЧЕНИЕ И НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ОБУЧЕНИЯ

### Методологические и теоретические проблемы

Исходная позиция автора состоит в том, что программированное обучение представляет дидактическую систему, возникшую на стыке педагогики и кибернетики, а его теория составляет «органическое сочетание общей теории управления с психолого-педагогической теорией обучения» (Н. Ф. Талызина, 1969).

Возникновение программированного обучения было обусловлено необходимостью решения важных педагогических задач, а именно: упорядочение содержания образования, его логической структуры; глубокая индивидуализация обучения с наиболее полным учетом психологических механизмов учения и на этой основе — формирование творческих возможностей обучаемых: повышение общей эффективности обучения с опорой на закономерности процессов управления и использование современных ЭВМ.

Создание качественных программных педагогических средств для ПЭВМ, эффективных компьютерных технологий обучения возможно только на основе хорошо разработанной соответствующей теории, которой и является теория программированного обучения.

Необходимо подчеркнуть, что программированное обучение несло в систему образования идеи научно-технического прогресса, связанные с автоматизацией человеческой деятельности и внедрением электронно-вычислительной техники.

В развитии программированного обучения представляется целесообразным выделить следующие этапы:

1. Формализация и логизация процесса обучения — до возникновения общей теории управления и появления ЭВМ.
2. «Скрытый» период накопления информации об управлении процессом обучения, зарождение теории программированного обучения — от середины 40-х до конца 50-х годов.
3. Период бурного и вместе с тем скачкообразного развития программированного обучения в его первоначальной форме — от конца 50-х начала 60-х годов до появления ПЭВМ.
4. Второе «рождение» программированного обучения: от появления компьютера — до настоящего времени.

Последний этап характеризуется бурным развитием ЭВТ и появлением у ПЭВМ принципиально новых качеств, возможностей, связанных с интерактивными дисками и цифровыми интерактивными видеосистемами, электронными планшетами, оптическими картами, возможностями электронной почты, созданием сетей (Network), в том числе сетей интегрированной цифровой связи (ISDN). В педагогическом плане особое значение приобретают графические возможности современных компьютеров, а также реализация с их помощью диалоговых форм общения.

Прежде чем рассматривать содержание современной теории программированного обучения в нашем понимании, хотелось бы высказать соображения о соотношении и

взаимопроникновении общей теории обучения и теории программированного обучения, а также о соотношении понятий: «программированное обучение», «новые информационные технологии обучения», «компьютеризированное обучение» и другие аналогичные.

Будем называть ту систему обучения, которая существовала задолго до возникновения программированного обучения, традиционной. Этой традиционной системе обучения соответствует общая теория обучения. В недрах традиционной системы обучения возникло относительно давно направление, получившее название «развивающее обучение» (оно особенно выражено в работах И. Песталоцци, А. Дистервега, П. Ф. Каптерева и других выдающихся педагогов).

Вначале ему как бы противостояло программированное обучение, которое в основе своей реализации имело обучающие алгоритмы.

И то, и другое направление развиваются (см. рис.1), как развивается и общая те-

нию единой общей теории обучения, которая станет и теоретической основой различных компьютерных и других технологий обучения. Границами такого скачка, связанного с развитием ЭВТ, являются возможности ПЭВМ: распознавание не только образов, но и смысла объектов, различных ситуаций и соответственно принятие машинной решения. Пока же мы должны всемерно развивать программированное обучение, опирающееся на общую теорию обучения, соответствующие психологические концепции учения, закономерности управления сложными динамическими системами, несомненно при этом учитывая реальные и перспективные возможности ПЭВМ.

Теория программированного обучения сохраняет общедидактические принципы, опирается на общие закономерности процесса обучения, однако она содержит и такие элементы, которые выходят за пределы общей теории обучения. Это элементы, связанные с закономерностями процессов управления, с необходимостью при решении ряда дидактических задач

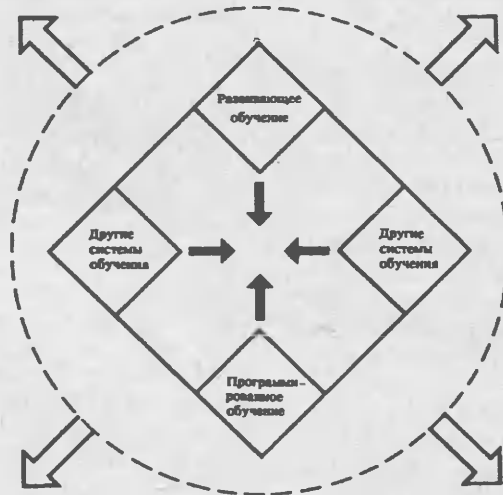


Рис.1

ория обучения, прежде всего, в результате глубокой психологизации обучения, взаимодействия названных выше направлений, наиболее полного учета возможностей развивающейся электронно-вычислительной техники.

Таким образом, педагогическое знание в своем развитии движется к созда-

использовать жестко алгоритмизированные процедуры, с необходимостью учитывать технические возможности ПЭВМ и других средств ЭВТ.

Нельзя учить, используя какие бы то ни было средства обучения, какими бы совершенными они не были, не следуя закономерностям процесса обучения, не

проникая в его сущность, точно также как нельзя хорошо программировать, не зная логических, информационных, технических основ программирования, особенностей языка и технических возможностей того или иного класса компьютеров. Здравый смысл важен везде, он просто необходим, но совершенно недостаточен для решения тех или иных задач. Поэтому просто приходится удивляться, когда хороший программист, либо человек, хорошо знающий предмет, либо человек, обладающий теми и другими качествами, но слабо разбирающийся в психолого-педагогических основах обучения, пытается создать эффективные программные педагогические средства, особенно обучающие программы.

В статье описывается разработанная нами общая концептуальная модель программированного обучения. Ее методологические основы вкратце таковы. Мы исходим из четырех универсально абстрактных структур, которые рассматриваем в данной области как субстанции, с соответствующими субстанциональными свойствами, отношениями, законами. Это категории: «личность», «деятельность», «познание», «управление». В качестве проявления этих субстанций их модусами выступают виды познавательной деятельности в обучении (репродуктивная, репродуктивно-преобразовательная, продуктивная), структуры интеллектуальных процессов (от алгоритмических до эвристических), психологические механизмы усвоения знаний и формирования интеллектуальных действий.

Очевидно, нельзя строить теорию обучения, в которой бы механизмы преобразования «входа» обучаемого в соответствующий «выход» не учитывались или учитывались «между прочим». Возникает проблема, как они должны учитываться, как должна строиться теория обучения, включающая в качестве своего ядра механизмы психических преобразований личностных свойств и возможностей.

Существует несколько различных подходов к решению такой проблемы. Первый состоит в том, что авторы определенной концепции учения абсолютизируют те или

иные механизмы и на основе этой абсолютизации строят соответствующую теорию обучения. Так теория планомерного формирования умственных действий представлялась ее авторами и последователями как обоснованная и единственно правильная теория программированного обучения. Аналогичные тенденции проявлялись и в теоретических разработках сторонниками механизмов ассоциаций и систем, как формирующих ум человека (Ю. А. Самарин, А. Ф. Эсаулов), последователями концепции теоретическое мышление (В. В. Давыдов), сторонниками проблемного обучения и механизмов решения проблемных ситуаций в развивающем обучении (Т. В. Кудрявцев, А. М. Матюшкин, М. И. Махмутов) и, наконец, абсолютизация алгоритмических процедур и логических действий многими сторонниками программированного обучения.

Суть второго подхода можно было бы выразить в просто познавательном отношении к механизмам умственных действий. Так поступает большинство дидактов, включающих психологические основы деятельности учащихся в теорию обучения.

Подход, которого придерживается автор, предполагает реализацию следующих теоретических положений:

- а) не может быть единственно правильной, абсолютно «работающей» концепции учения, ибо, механизмы, характеризующие преобразование «входа» в «выход», относятся к вполне определенным условиям деятельности, и их использование во всех других случаях не всегда правомерно;*
- б) решение вопроса об использовании той или иной концепции учения прежде всего связано с областью ее применимости (Л. Н. Ланда, 1975);*
- в) совершенно не исключается, что один и тот же объект может рассматриваться одновременно с позиций двух концепций (аналог — дуализм в физике: корпускула — волна);*
- г) конструируемая модель обучения должна содержать инвариантную часть, более или менее устойчивую*

*к различным условиям обучения, и динамичную подвижную часть, в которой наиболее полно учитывались бы конкретные условия обучения и соответствующие механизмы умственных и предметных действий.*

Все механизмы взаимодействия учителя и ученика, а также механизмы усвоения знаний, формирования умственных и предметных действий могут быть выражены с помощью такой иерархической структуры:

*механизмы формирования личности в деятельности; понимание процесса обучения как включения обучаемого в систему деятельности, отношений и общения (Г.Н. Александров); механизмы взаимодействий знаний, деятельности и перестраивающихся установок в умственном развитии обучаемых (С.Л. Рубинштейн); механизмы познавательной деятельности в обучении: взаимодействия видов познавательной деятельности (Г.Н. Александров); механизмы ориентировочной основы действия как механизмы взаимодействия алгоритмических и эвристических процессов; механизмы самих психических процессов, умственных и предметных действий.*

Область применимости тех или иных механизмов определяется целями обучения,

дидактическими задачами, особенностями логической структуры знаний, индивидуальными характеристиками обучаемых и возможностями совместной учебной деятельности, условиями обучения.

Схематически описанный выше подход к конструированию модели программированного обучения представлен на рис. 2.

В состав общей концептуальной модели программированного обучения входят:

*учение о взаимодействии различных видов познавательной деятельности в обучении, при ведущей роли продуктивной; соответствующие рекомендации по возможным структурам процесса обучения, в том числе и проблемному обучению; учет закономерностей и противоречий познавательной деятельности;*  
*учение об ориентировочной основе действий (П.Я. Гальперин, Н.Ф. Талызина) с учетом соотношения алгоритмических и эвристических структур познавательных процессов и несколько ином понимании ориентировочной основы действий (Г.Н. Александров);*  
*использование рассмотренных механизмов действия, а также механизмов творческих процессов, учитывающих реальные ситуации обучения. (К таким механизмам обычно относят (И.П. Калюшина) ассоциативные, взаимодействия логического и интуитивного, анализа через синтез, особого выраже-*

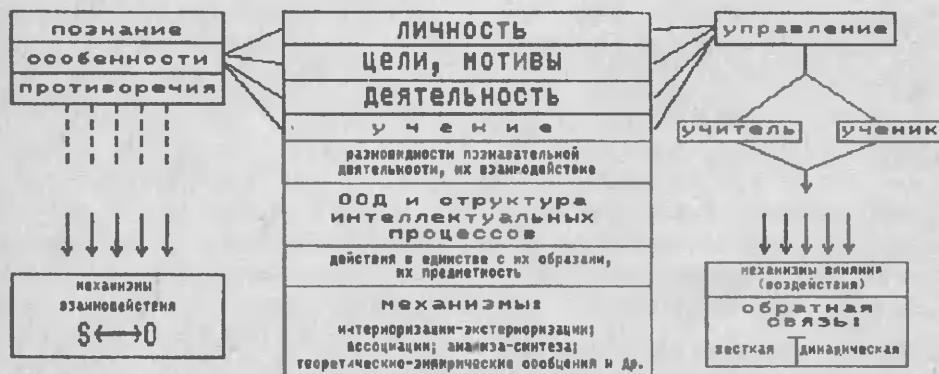


Рис. 2

*ния обратной связи в поисковой деятельности, специального обучения эвристическим приемам. Особое значение приобретают исследования в области комплексных характеристик творческой личности. В частности, речь идет об интеллектуальной активности и интеллектуальной инициативе (Д.Б. Богоявленская); учет закономерностей управления, динамично функционирующей обратной связи.*

Нами намечены области применимости различных психологических концепций учения, которые в силу ограниченных возможностей статьи, мы здесь не рассматриваем.

#### **Методический и технологические вопросы**

Как реализовать рассмотренную общую концептуальную модель программированного обучения? Из чего исходить при непосредственной разработке программных педагогических средств? Какие выделить разновидности этих средств и как их конструировать? На эти и некоторые другие вопросы, связанные с деятельностью преподавателя, использующего компьютер в учебном процессе, мы и попытаемся ответить.

Одной из основных закономерностей развития в процессе обучения является взаимодействие видов познавательной деятельности при ведущей роли продуктивной. Автор придерживается утверждения о том, что «развитие – способ существования взаимодействующих систем, связанный с перестройкой конкретной системы, с образованием качественно новых временных и пространственных структур» (Я. А. Пономарев). Следует добавить, что развитие выражается, главным образом, в становлении качественно новых содержательных образований.

Итак, познавательную деятельность обучаемого можно описать тремя основными разновидностями, которые в результате взаимодействия перестраивают-

ся, относительно низшие виды приобретают новые характерные черты, соотнобобщается деятельность разных уровней, и, в конечном счете, происходят изменения, составляющие суть развития познавательных возможностей и умственных сил обучаемого.

Самый относительно простой вид познавательной деятельности в обучении – это репродуктивная познавательная деятельность, деятельность по образцу. Здесь преимущественно срабатывают механизмы памяти. Обучаемый копирует в своих действиях образ, запечатленный в сознании. Нельзя сказать, что обучаемый не осуществляет при этом операций анализа и синтеза. Но они имеют примитивную форму, обеспечивающую выбор из памяти адекватного требованиям опрашиваемого ответа.

Самый высокий уровень деятельности выражается в творческих действиях (умениях), когда обучаемый открывает новые для себя знания, новые способы действий, когда он самостоятельно не только решает проблему, но и обладает умением видеть ее. Здесь в полной мере развернута аналитико-синтетическая деятельность, осуществляется обобщение, «далекие» переносы знаний и т. д.

Однако использование этих двух видов познавательной деятельности недостаточно как для дифференциации целей обучения, так и степени усвоения знаний.

Вводится промежуточный уровень – уровень репродуктивно-преобразовательной деятельности, который характеризует действия обучаемого, исключая непосредственное использование прошлого опыта. Это действия по распознаванию класса задач с последующим применением алгоритма или способа решения; это действия, содержащие те или иные преобразования исходных данных до извлечения из памяти известного способа решения и др. Конечно, это планируемые преобразования, но с элементами поиска. Это уровень не только

процессов памяти, но и развитого логического мышления.

Уровни репродуктивной и продуктивной познавательной деятельности соответственно «расщепляются». *Первый уровень* – на подуровни: узнавания, распознавания, воспроизведения; *второй* – на подуровни: стимульно-продуктивной, эвристической и креативной познавательной деятельности. Каждый из видов деятельности характеризуется системой умений и приемов. Они подробно описаны в ряде работ автора.

Общими количественными характеристиками уровня усвоенных знаний при непосредственном измерении могут быть: количество правильно выполненных заданий в отведенное время, количество допущенных ошибок, либо время выполнения каждого задания и время безошибочного выполнения всех заданий. Уровень развития выявляется при соотношении уровней усвоения знаний и овладения способами деятельности. Что касается креативного уровня познавательной деятельности (Д. Б. Богоявленская), то его отличает наличие интеллектуальной инициативы, – мышление не прекращается после нахождения решения поставленной задачи. Интеллектуальная активность личности проявляется, хотя она не стимулирована ни внешними факторами, ни субъективной оценкой неудовлетворенности результатов деятельности.

В ряде случаев для выявления уровня продуктивной познавательной деятельности необходимо осуществить специальное психологическое тестирование.

В соответствии с рассмотренным подходом автором разработана типология учебных занятий, выработаны рекомендации для создания программных педагогических средств.

Понятия репродуктивной и продуктивной деятельности встречаются в работах В. П. Беспалько, Б. И. Коротяева, П. И. Пидкасистого, Т. И. Шамовой и др. В отличие от этих подходов, автор описал каждый вид деятельности с помощью соответствующих умений и действий, выдвинул

идею их взаимодействия и показал соответственно пути разрешения противоречий с помощью промежуточных разновидностей и действий, обосновал возможные разновидности учебных занятий.

Так, характеризуя продуктивную познавательную деятельность, автор выделяет для ее описания следующие действия (умения. приемы):

*поиск, самостоятельное определение признаков, классификация объектов согласно этим признакам; поиск аналогий, перенос знаний из одной ситуации (условий) в другие; выявление общих закономерностей (на теоретическом и практическом материале), самостоятельное обобщение; конструирование нового объекта с заданными свойствами из известных элементов; конструирование нового объекта с заданными свойствами из элементов, часть которых неизвестна и свойства которых нужно задать; нахождение нестандартных способов и приемов решения задачи, при условии, что стандартные способы известны; поиск решения нестандартных задач (алгоритмы и способы решения которых решающему неизвестны); самостоятельное составление задач, решение которых предполагает оригинальные приемы и способы; выявление и формулирование проблемы в сложившейся ситуации (в ряде случаев это действие может быть выражено в постановке вопроса).*

Аналогично описываются репродуктивная и репродуктивно-преобразовательная деятельность. Рассматривается их взаимодействие. В зависимости от характера этого взаимодействия выделяются различные типы обучения: «традиционный», обратного движения в учебном познании от продуктивной к другим видам познавательной деятельности, специаль-

ное обучение способам деятельности (способ добывания знаний), «выход» обучаемых на уровень продуктивной деятельности через самостоятельное осмысление данного образца действий и его обобщение, выход на уровень продуктивной деятельности через овладение знаниями высокого уровня абстракции и обобщения; собственно проблемное обучение как обучение преимущественно на уровне продуктивной деятельности; при усвоении разных знаний используются одни и те же действия (логические приемы) либо, наоборот, к различным знаниям «применяются» различные действия. Здесь можно ожидать, что обучаемый, овладевая различными приемами умственной деятельности, повышает свои творческие возможности.

Исходная посылка автора состоит в том, что глобальная цель обучения должна соответствовать модели подготавливаемого специалиста. Этой цели подчиняются все частные цели, в том числе и цели реализации программных педагогических средств. Цели ставятся так, чтобы можно было установить, насколько их достижению ясно и понятно обучаемому и обучаемому. Цели, как правило, выражаются в тех или иных действиях и определяют стратегию и тактику обучения. Достижение целей устанавливается как статистически (количественно), так и качественно. Одним из важных качественных критериев достижения целей умственного развития, по нашему мнению, становится уровень сформированности умений репродуктивно-преобразовательной и особенно продуктивной познавательной деятельности обучаемых (в теории и практике критериально-ориентированных тестов – уровень мастерства).

Как известно, обычно выделяется три группы целей (задач) обучения: образовательные, развивающие, воспитательные.

В статье мы остановимся лишь на формулировке развивающих целей, полагая, что с постановкой целей других видов читатель справится самостоятельно.

Развивающие цели обучения выражаются в успешном выполнении действий,

относящихся к области логического мышления (I), поисковой деятельности (II), области, предполагающей владение некоторыми интегративными действиями – переработки и осмысления текстов разных видов (III).

**Первая группа целей.** 1.1. Выделять признаки понятий, объектов; различать существенные и несущественные признаки. 1.2. Осуществлять анализ различных объектов. 1.3. Сравнивать те или иные объекты. 1.4. Давать определения понятиям, различать их объем и содержание; различать необходимые и достаточные признаки в разных отношениях. Подводить объект под понятие. 1.5. Осуществлять деление понятия и пользоваться классификационными схемами. 1.6. Устанавливать элементы структуры суждения (посылки, заключения). Различать эквивалентные и неэквивалентные суждения. 1.7. Различать виды умозаключений, строить индуктивные, дедуктивные умозаключения, умозаключения по аналогии. 1.8. Строить отрицания к различным суждениям, содержащим термины «все», «каждый», «существует», использовать термины «хотя бы...», «по крайней мере...», «не больше...», «не меньше...», «тогда и только тогда, когда...» и др. 1.9. Понимать структуру доказательства, его элементы. Строить доказательство различными способами. 1.10. Опровергать те или иные утверждения. 1.11. Выявлять логическую структуру различных утверждений. 1.12. Понимать структуру объяснения. Строить объяснение.

**Вторая группа целей.** 2.1. Самостоятельно обобщать. 2.2. Выбирать признаки деления и строить классификационные схемы. 2.3. Находить нестандартные приемы решения задач (видеть новые свойства известных объектов). 2.4. Решать нестандартные задачи. 2.4.1. Анализировать условия задачи и ее требования. 2.4.2. Переформулировать задачу, ее элементы. 2.4.3. Строить символические, графические и другие модели задачи. 2.4.4. Выдвигать гипотезы, соотносить их друг с другом. Осуществлять антиципацию. 2.4.5. Осуществлять действия

во внутреннем плане. 2.4.6. Планировать и проводить мысленный эксперимент. 2.4.7. Разрабатывать план решения. 2.4.8. Реализовывать план решения с поэтапным контролем своей деятельности. 2.4.9. Анализировать полученные результаты. Искать и отбирать наиболее рациональные способы действия. 2.5. Самостоятельно конструировать новый объект, с новыми свойствами из известных элементов. 2.6. Самостоятельно составлять задачи, требующие нестандартных действий. 2.7. Видеть (находить) проблему, формулировать ее (в частном случае – ставить нестандартный вопрос).

**Третья группа целей.** 3.1. Выделять смысловые элементы текста при его чтении на разных уровнях (просмотровое, ознакомительное, изучающее чтение и уяснение содержания). 3.2. Увидеть замысел автора и выделить основные идеи текста. 3.3. Выделить главное в прочитанном и построить его общую тематическую структуру. 3.4. Составлять логико-структурную схему прочитанного. 3.5. Переструктурировать текст по тем или иным основаниям. 3.6. Извлекать из текста факты, описания, теории, раскрывающие и подтверждающие определенные положения, те или иные связи и отношения и др. 3.7. Передать содержание текста с той или иной установкой на воспроизведение. 3.8. Выполнять записи на основе произведенной работы с текстом (план, конспект, тезисы, аннотации, выписки из разных видов и др.).

К числу действий, характеризующих осмысливание и творческую переработку текстов, мы относим такие действия:

*выделять фрагменты текста и давать им заголовки;  
отвечать на поставленные вопросы;*

*самостоятельно ставить вопросы к тексту (разного характера и направленности);*

*предвосхищать мысли автора, конструировать возможные авторские подходы к тем или иным фактам, событиям и т.д.;*

*критически оценивать различные тексты и сообщения под определенным углом зрения.*

Следующий шаг в развитии нашей программы состоит в построении классификации программных педагогических средств. Речь идет о такой классификации, которая ориентировала бы психологов, методистов, предметников, программистов в основных задачах, решаемых ими, в их структурах, отражающих характерные черты познавательной и интеллектуальной деятельности, в особенностях реализуемых педагогических функций.

По мнению автора ни одна из существующих классификаций этим требованиям не отвечает. Он полагает, что все средства, реализуемые с помощью ПЭВМ для достижения педагогических целей, подразделяются на две основные группы: инструментальные педагогические средства (ИПС) и программные педагогические средства (ППС) (рис.3). К инструментальным педагогическим средствам отно-

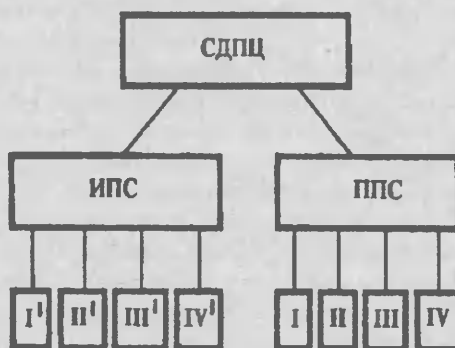


Рис. 3

сятся: справочно-информационные (I'); средства, реализующие коммуникативные функции (парное, внутри- межгрупповое взаимодействие, в том числе электронная почта, компьютерные конференции и другие) (II'); конструкторско-комбинаторные, позволяющие моделировать различные ситуации и решать конструкторские задачи (III'); комбинированные, сочетающие возможности и свойства каж-



дого из названных классов (IV'). Эта группа средств характеризует выраженные технические возможности ПЭВМ, имеющие некоторые интегративные педагогические функции, реализуемые в программных педагогических средствах. Программные педагогические средства, в свою очередь, подразделяются на основные классы: обучающие (I), контролирующие (II), для экспериментальных исследований (III). В особый класс выделяются языковые программы (IV).

В классе обучающих программ выделяются подклассы по следующим признакам:

*а) решаемым дидактическим задачам;*

*б) возможностям формализации языка общения, учебных текстов, их фрагментов в процессе обучения;*

*в) по характеру интеллектуальных процессов, происходящих при решении задач;*

*г) содержащим или не содержащим игровые элементы;*

*д) по способам реализации;*

*е) по особенностям структурирования содержания (рис.4). Назовем разновидности обучающих программ по выделенным признакам.*

1a1 – овладение новыми знаниями: понятиями, системами понятий; 1a2 – овладение новыми знаниями другого вида: закономерностями, принципами, теориями, правилами действий; 1a3 – выработка навыков (программы для тренажеров); 1a4 – обучение решению задач; 1a5 – повторение и закрепление знаний; 1a6 – обучающие программы (фрагменты) в структуре лабораторных занятий (практикумов); 1a7 – целенаправленное формирование интеллектуальных и других сложных умений; 1a8 – комбинированные программы.

1b1 – высокий уровень возможностей; 1b2 – средний уровень возможностей; 1b3 – ниже среднего уровня возможностей; 1b4 – неформализуемые или плохо формализуемые знания.

1v1 – алгоритмические, 1v2 – квазиалгоритмические, 1v3 – полуэвристические, 1v4 – эвристические.

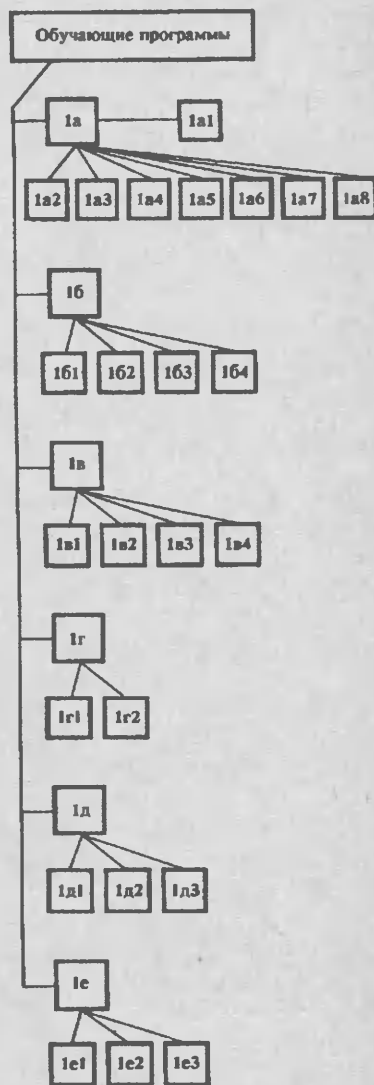


Рис. 4

1r1 – содержащие игровые элементы, 1r2 – не содержащие игровых элементов.

1d1 – линейные программы, 1d2 – разветвленные программы, 1d3 – адаптивные программы.

1e1 – с выделением модулей, 1e2 – без выделения модулей, 1e3 – комбинированные.

На рис. 5 представлена схема разновидностей контролирующих программ, а на рис. 6 – языковых.

Поясним введенные обозначения. Для контролирующих программ:

2а – по признаку целевого предназначения и дидактическим функциям. 2а1 –

программы, выполняющие функции обратной связи; 2а2 – обычного контроля разных уровней (текущего, рубежного, итогового, «входного» общего, «выходного» общего); 2а3 – тесты достижений;

2б – по степени гомогенности, 2б1 – программы, содержащие задания одинаковой трудности; 2б2 – программы, содержащие задания разной трудности;

2в – по способу структурирования ответов. 2в1 – программы с избирательными ответами (альтернативные ответы, ответы, построенные по принципу множественного выбора, ответы перекрестного выбора); 2в2 – ответы непосредственного построения и ввода.

2г – по статусу программы. 2г1 – стандартизованные программы, 2г2 – нестандартизованные программы.

Языковые программы выделены в особый класс в силу специфики речевой деятельности и соответствующих особенностей обучения и контроля знаний в этой области. По существу здесь выделены два основных подкласса: языковые тесты и обучающие программы.

Наиболее детальная и развернутая классификация дидактических языковых тестов дана в работе: Рапопорт И., Сельг Р., Соттер Н. Тесты в обучении иностранным языкам в средней школе. Таллин, 1987. Она осуществлена по двум основаниям: 1) целевым и содержательным признакам, 2) формальным признакам.

Мы следовали этому подходу, дополнив класс языковых программ обучающими:

4в1 – в области грамматического материала, 4в2 – лексического материала, 4в3 – в части формирования речевых умений.

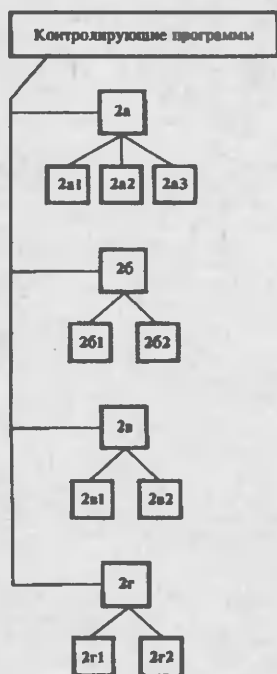


Рис. 5

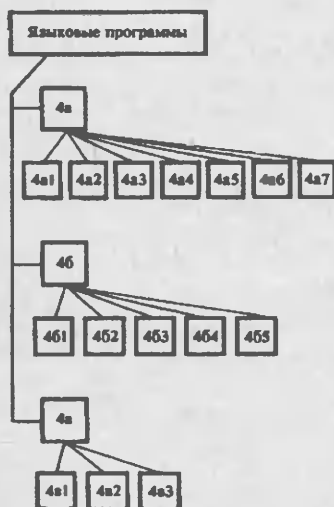


Рис. 6

*Прежде чем перейти к рассмотрению различных вариантов программных средств, хотелось бы ответить на вопрос: какую цель я ставил перед собою, строя такую достаточно сложную классификационную схему. Смысл моих действий состоял в том, чтобы раскрыть перед разработчиком более или менее полно то реальное многообразие объектов, с которыми ему придется встретиться. Это и своеобразное предупреждение против возможных упрощенных, а иногда и просто грубо искажающих педагогический смысл действий, это и определенная педагогическая ориентировка в предстоящей деятельности, способствующая созданию педагогически рациональных программных средств.*

Приведем несколько примеров по классу обучающих программ, поясняющих высказанную мысль. Ведущими признаками классификации несомненно являются решаемые дидактические задачи и возможная степень формализации знаний, принадлежащих данной предметной области. Таким образом, надо вначале установить, чего мы хотим добиться и на каком материале. Одно дело – математические правила и теоремы, другое – знание физических или химических формул, третье – грамматические правила, четвертое – это закономерности исторического развития, пятое – это характеры, отношения, чувства литературных героев. Очевидно, в первых трех случаях есть много общего, материал хорошо формализуем, чего не скажешь о четвертом и пятом случаях. Быть может и нет смысла разрабатывать компьютерные программы для изучения таких объектов. После этого надо решить, на уровне каких структур интеллектуальных процессов можно вести обучение и соответственно, какую структуру обучающей программы избрать. Варианты подобного анализа показаны: овладение новыми знаниями (рис.7), выработка навыков (рис.8), обучение решению задач (рис.9), обучающие программы с элементами игры (рис.10). В зависимости от решаемой дидактической задачи, тех интеллектуальных процессов, которые мы ожидаем в деятельности обучаемых, прежде всего и определяются общие структуры обучающих программ. Так при выработке навыков, это будут преимущественно линейные программы, при обучении решению задач, линейные программы скорее всего будут использоваться в случаях алгоритмических и квазиалгоритмических структур умственной деятельности учащихся. Разветвленные и адаптивные программы приобретают особый смысл, когда структуры умственной деятельности решающего содержат полувзвистические и взвистические процессы. Заметим, что возможное управление такими процессами в наибольшей мере способствует развитию творческого потенциала обучаемых.

Так в случае обучающих интеллектуальных игр предпочтение следует отдать разветвленным и адаптивным программам, учитывающим квазиалгоритмиче-



Рис. 7

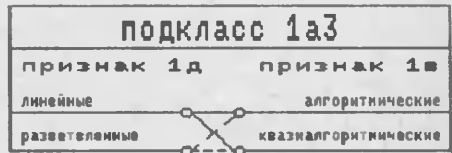


Рис. 8



Рис. 9

ские, полувзвистические и взвистические процессы в деятельности решателя задач.

Мы полагаем, что в общую систему условий адаптивного обучения нужно обязательно включить: соответствие психологических концепций учения области их применимости; возможности саморегуляции активности обучаемого, сочетаемой с внешними воздействиями учителя.

Уточнение вида программы, ее содержания и структуры происходит на основе разработанной общей концептуальной модели программированного обучения.

В соответствии с разработанной моделью обучения предлагаются варианты блок-схем обучающих программ, которые

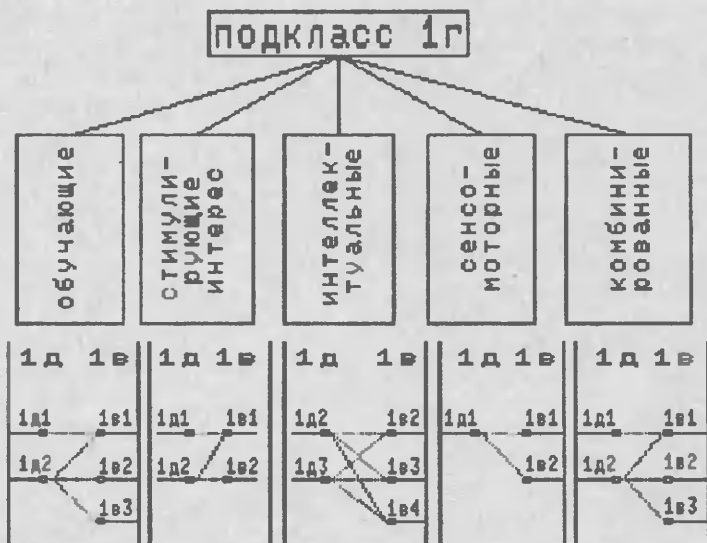


Рис. 10

автором и его учениками реализованы в практической деятельности.

#### Вариант 1.

*Общий тип.* Программа включает следующие основные блоки: «входа», проблемной ситуации, формирования ООД, применения знаний и обобщения, диагностики усвоения, «выхода». Такая программа может использоваться для решения различных дидактических задач.

#### Вариант 2.

*Усвоение новых знаний разного вида.* Он содержит блоки: «входа», постановки проблемы, объяснения, ООД, демонстрации элементов ООД, применения и закрепления знаний, диагностики усвоения, «выхода».

#### Вариант 3.

*Обучение понятиям.* Основные блоки: «вход», выход в метасистему, переход в ассоциативные системы, варьирование признаков, применение знаний и обобщение, «выход». Выход в метасистему осуществляется, как правило, на уровне репродуктивно-преобразовательной деятельности. Примеры ассоциативных систем разных рангов и уровней: система существенных признаков понятия (конъюнктивная, дизъюнктивная, смешанная); система необходимых и достаточных при-

знаков в разных соотношениях на языке сентенциональных связей; система правил вывода при различных способах доказательства и другие.

#### Вариант 4.

*Усвоение новых правил действия:* обычные блоки «входа» и «выхода», между которыми включаются блоки постановки проблемы, диалога, построения алгоритма (правила), прямого применения алгоритма, применения алгоритма в сходных, но измененных условиях; деформированных упражнений, неопределенных и обратных упражнений, обобщения.

#### Вариант 5.

*Обучение комплексу знаний (на уровне учебной темы).* Обучающие программы содержат блоки: входного контроля, с помощью которого проверяется уровень подготовки обучаемых, необходимый для усвоения новых знаний; подведения обучаемых к алгоритму или способу решения соответствующего класса задач; формирования ООД; применения знаний, в котором решаются конкретные задачи и приводятся примеры; выходного блока, где подводятся итоги деятельности обучаемых.

Обучение ведется на уровне взаимодействия всех видов познавательной дея-

тельности при ведущей роли продуктивной. При подведении обучаемых к алгоритму или способу решения задачи могут использоваться различные механизмы умственных действий.

**Вариант 6.**

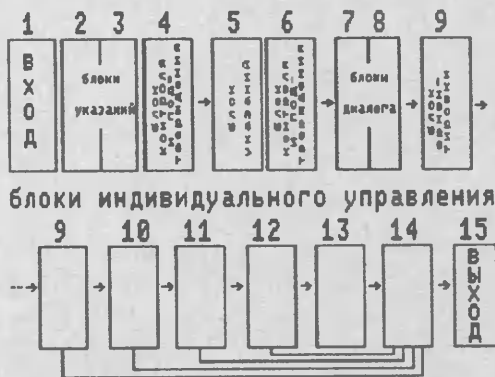


Рис. 11

**Обучение с элементами индивидуализации (рис.11).** Сущность реализуемой идеи можно пояснить так. Готовность к усвоению и применению новых знаний характеризуется определенным исходным уровнем. Решив задачу или приобретая новые знания в какой-то проблемной ситуации, учащийся как бы поднимается на новый уровень своих возможностей. Меж-

ду этими уровнями образуется некоторое смысловое пространство, преодоление которого, разумеется и соответствующих познавательных противоречий, скорее всего происходит дискретно, по некоторым уровням, называемым автором, промежуточными. Движение обучаемого от уровня к уровню происходит либо самостоятельно, либо под различными воздействиями учителя (подсказки, рекомендации, сообщение различных ориентиров для продвижения в решении, в том числе и эвристик и т. д.).

Автор построил и обосновал систему информационно-педагогического обеспечения компьютерных обучающихся и контролирующих программ. В эту систему включены: матрица-таблица с общими результатами (по всем *i*-испытуемым, *j*-задачам; решенная задача оценивается 1, нерешенная – 0), суммирование производится по каждой строке (для каждого испытуемого) и по всем столбцам (по каждому выполняемому заданию).

Затем определяется эмпирический вес задания, диагностический вес, уточняющий первоначальную двухбалльную шкалу, индекс дифференцирующей силы задания, среднее взвешенное каждого задания, полигон частот, вариационный ряд успешного решения задач, дисперсия и среднеквадратичное для этого ряда.

Содержание предложенной системы может варьироваться в зависимости от целей реализации компьютерных программ.

*Сведения, необходимые для подписки на журнал «Информатика и образование» и его приложения на 2-е полугодие 1994 г.:*

Название журнала	Индекс издания	Стоимость подписки за полугодие, руб.	Периодичность
<b>«ИНФОРМАТИКА И ОБРАЗОВАНИЕ»</b>			
для индивидуальных подписчиков	70423	18000	1 раз в 2 месяца
для предприятий и организаций	73176	36000	1 раз в 2 месяца
<b>БИБЛИОТЕКА ЖУРНАЛА "ИНФОРМАТИКА И ОБРАЗОВАНИЕ"</b>			
<b>«ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР БК-0010 - БК-0011М»</b>			
для индивидуальных подписчиков	73177	12000	1 раз в 2 месяца
для предприятий и организаций	73092	21000	1 раз в 2 месяца
<b>«ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР УКНЦ»</b>			
для индивидуальных подписчиков	73093	10000	1 раз в 3 месяца
для предприятий и организаций	73179	20000	1 раз в 3 месяца

# Предприятие «ИнфоМир» предлагает:



## КуМир-Гипертекст (КуМир-93)

**НОВОЕ ПОКОЛЕНИЕ программных и методических  
средств для IBM PC**

**КуМир-Гипертекст — это:**

- режимы демонстрации и обучения в диалоге с ЭВМ;
- практикумы с возможностью получения подсказки по задаче и проверки ответа;
- проведение автоматизированных контрольных работ;
- готовый комплект гипертекстов для каждого урока информатики;
- возможность подготовки собственных гипертекстов для уроков информатики и математики;
- полная поддержка школьного алгоритмического языка, описанного в учебнике А.Г.Кушниренко и др. «Основы информатики и вычислительной техники». — М.: Просвещение, 1990, 91, 93 и в пособии для учителя А.В.Авербух и др. «Изучение основ информатики и вычислительной техники». — М.: Просвещение, 1992;
- мгновенная и наглядная диагностика всех ошибок ученика;
- обучающая программа 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.

**Предприятие «ИнфоМир» поставляет:**

КуМир-93 — базовый комплект, включающий исполнители Робот, Чертежник и др., обучающие программы по школьному алгоритмическому языку и системе «КуМир».

**В дополнение к базовому комплекту могут быть поставлены:**

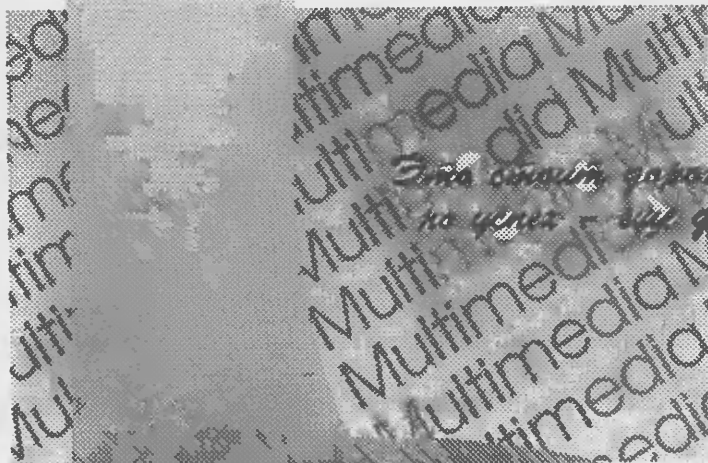
1. Комплект «Информатика-10», включающий Гипертекст «Основы алгоритмизации», комплект гипертекстов для проведения уроков в 10 классе и комплект из 6 автоматизированных контрольных работ.
2. Пакет «Функции и Графики».
3. Пакет «Планиметрия».
4. Инструментальный комплект для разработки Гипертекстов и внешних исполнителей в системе КуМир.

**ВНИМАНИЕ!** Для зарегистрированных пользователей  
КуМира-91 скидка 60% !

**ДЛЯ ПРИОБРЕТЕНИЯ ПРОГРАММНЫХ СРЕДСТВ  
НАПРАВЛЯЙТЕ ЗАЯВКИ ПО АДРЕСУ:**

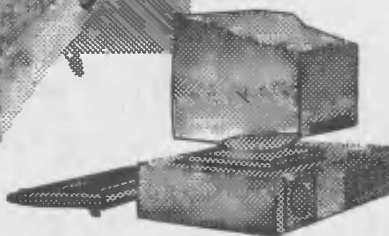
**103051, Москва, Садовая-Сухаревская, 16, комн.9. «Информатика и образование».**

**Зарегистрированные пользователи должны указывать  
регистрационный номер**



*Эта самая дорога,  
но цена — две дорожки!*

# Multivision Pro



**ВЫ ХОТИТЕ:**

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

**ТОГДА ВАМ НЕОБХОДИМ:**

комплекс **Multivision Pro**

**AIST**

**Advanced Instructional Software Trading AB**

РОССИЯ, 109060, МОСКВА, К. 60 з/п № 00, АИСТ АБ; ТЭЛ. 0657246700 - ФАКС 0657462

**MultiVision Pro**

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

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



**CANADA**

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



**U.S.A.**

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



**U.S.A.**

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





## U. S. A.

Фирма Proxima (США) представляет лучшую в мире цветную жидкокристаллическую панель Ovation, уникальную сенсорную камеру Cyclops, и лазерную указку. Ovation проецирует компьютерное и видеозображение 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 мультимедиа поможет не просто эффективно подать лекционный материал, но и оживить его. Теперь Вы не тратите время на поиски наглядных пособий, Вы делаете их сами!



## GERMANY

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



## JAPAN

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

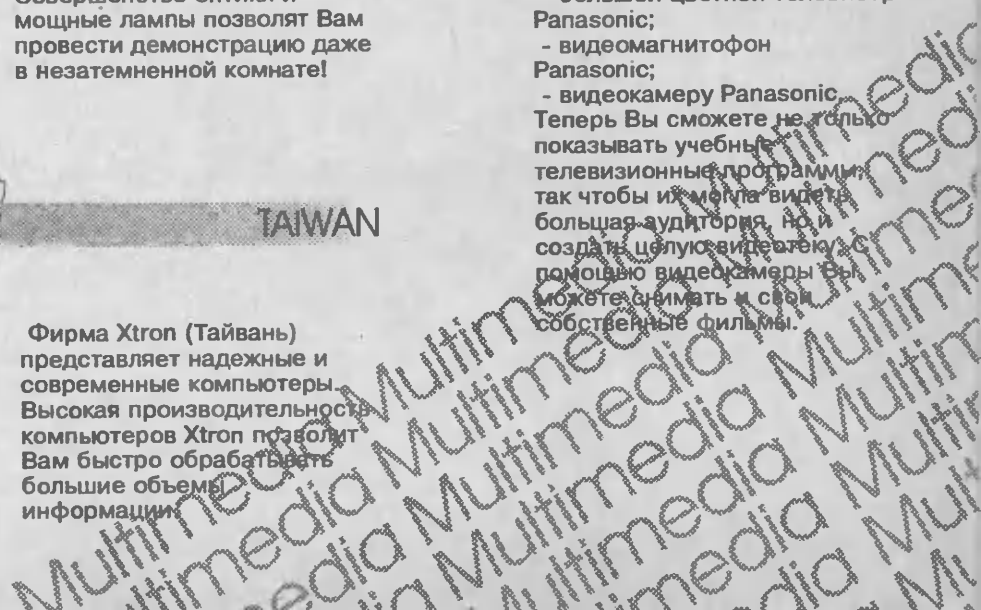
- большой цветной телевизор Panasonic;
- видеомагнитофон Panasonic;
- видеокамеру Panasonic.

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



## TAIWAN

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



## Состав комплекса MultiVision Pro



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

# MultiVision Pro

**AIST** Advanced Instructional  
Software Trading AB

РОССИЯ 103050 Москва К-50 А/Я №80 АИСТ АВ  
телефон: (095) 229-67-06; телефакс: (095) 229-76-53

Е. А. Ерохина

## ОТ ЛОГИКИ К ПРОГРАММИРОВАНИЮ (Пролог в курсе информатики)\*

### 11. Пример из геометрии: база знаний «треугольники»

Язык Пролог позволяет создавать базы знаний, при помощи которых можно решать задачи из различных предметных областей. Рассмотрим базу знаний из геометрии. В ней записаны сведения о некоторых геометрических фигурах. Требуется проверить, какие из этих фигур являются треугольниками. Назовем эти фигуры «кандидатами» (в том смысле, что это пока только кандидаты в треугольники). Тогда факты можно записать в виде:

кандидат(<имя>, <1-я сторона>,  
<2-я сторона>, <3-я сторона>);

Например, факты могут быть такими:

кандидат(ABC, 3, 4, 5);  
кандидат(DEF, 7, 15, 11);  
кандидат(MNP, 4, 3, 8);

Чтобы определить, является ли наш «кандидат» треугольником, надо проверить, удовлетворяют ли его стороны неравенству треугольника: «сумма двух любых сторон треугольника должна быть больше третьей стороны». Если описать это соотношение математически, то для всех возможных сочетаний сторон получим три неравенства:

$$a + b > c, \quad a + c > b, \quad b + c > a.$$

Чтобы не проверять все эти неравенства, можно «усовершенствовать» неравенство треугольника. Действительно, если сравнить наиболее длинную сторону (или одну из них, если треугольник равнобедренный) с суммой двух других сторон, то выполнение неравенства для остальных троек очевидно (в качестве упражнения докажите, что это так). Значит, нам потребуется правило для определения самой длинной стороны (или максимума из трех чисел). Мы можем определить максимум из двух чисел:

$$\max(a, b) = \begin{cases} a, & \text{при } a \geq b \\ b, & \text{при } a < b \end{cases}$$

Чтобы избежать написания правила «больше или равно» вместо двух правил составим три:

\* Продолжение (начало см. в ИНФО №2-93 г.).

$\max(a, b, a) < \text{БОЛЬШЕ}(a, b);$   
 $\max(a, b, b) < \text{БОЛЬШЕ}(b, a);$   
 $\max(a, a, a);$

Имея правила для определения максимума из двух чисел, можно найти максимум из трех, используя вспомогательную переменную. Способ определения поясним схемой:

$a \square \max(a, b) = d \square b \square$	$c \square$
$d \square$	$\max(d, c) = m$

$$= > m = \max(a, b, c)$$

Правило на Прологе будет выглядеть так:

$\max3(a, b, c, m) < \max(a, b, d), \max(d, c, m);$

Тогда для того, чтобы определить, является ли данный кандидат треугольником, можно написать правило, проверяющее выполнение неравенства треугольника:

$\text{треуг}(\text{имя}') < \text{кандидат}(\text{имя}', a, b, c), \max3(a, b, c, m), \text{сумма3}(a, b, c, p),$   
 $\text{ВЫЧИТАНИЕ}(p, m, d), \text{БОЛЬШЕ}(d, m);$

По правилу сумма3 вычисляется сумма трех чисел. Правило выглядит так:

$\text{сумма3}(a, b, c, d) < \text{СЛОЖЕНИЕ}(a, b, p), \text{СЛОЖЕНИЕ}(p, c, d);$

(Чтобы не выяснять, какая именно сторона самая длинная, тем более что нас интересует только ее длина и сумма длин двух остальных, мы вычисляем сумму длин всех сторон, а затем вычитаем длину самой длинной:  $m$  – самая длинная сторона,  $a + b + c = p$ ,  $p - m = d$ ). Теперь можно проверить, какие из наших кандидатов являются треугольниками. Для проверки можно задавать вопросы вида:

?треуг(ABC);

Машина будет отвечать "ДА" или "НЕТ".

Теперь можно научить машину определять вид треугольника по его сторонам. Для прямоугольного треугольника можно использовать теорему Пифагора:  $a^2 + b^2 = c^2$ . Напишем отдельное правило для вычисления квадрата числа ( $a^2 = b$ ):

$\text{квадрат}(a, b) < \text{УМНОЖИТЬ}(a, a, b);$

Правило для определения прямоугольного треугольника будет таким:

$\text{прямоуг}(\text{имя}') < \text{кандидат}(\text{имя}', a, b, c), \max3(a, b, c, d), \text{квадрат}(d, p),$   
 $\text{квадрат}(a, a2'), \text{квадрат}(b, b2'), \text{квадрат}(c, c2'),$   
 $\text{квадрат}(d, d2'), \text{сумма3}(a2', b2', c2', k), \text{ВЫЧИТАНИЕ}(k, p, l),$   
 $\text{РАВНО}(p, l);$

Для определения суммы квадратов катетов использован тот же прием, что и в правиле "треуг".

Для тупоугольного и остроугольного треугольника можно использовать теорему косинусов:  $c^2 = a^2 + b^2 - 2 \cdot a \cdot b \cdot \cos(\text{alfa})$ , где  $\text{alfa}$  – угол между сторонами  $a$  и  $b$ . Пусть  $c$  – наибольшая из сторон (или одна из наибольших). Тогда в тупоугольном треугольнике

$c^2 > a^2 + b^2$ , а в остроугольном –  $c^2 < a^2 + b^2$ . Для тупоугольного треугольника правило на Прологе будет таким:

тупоуг(имя')<-кандидат(имя',a,b,c) , треуг(имя') , max3(a,b,c,d),  
 квадрат(d,p) , квадрат(a,a2') , квадрат(b,b2') ,  
 квадрат(c,c2') , квадрат(d,d2') , сумма3(a2',b2',c2',k) ,  
 ВЫЧИТАНИЕ(k,p,l),БОЛЬШЕ(p,l);

### Задачи и упражнения:

1. Написать правило для определения остроугольного треугольника.
2. Почему в правиле для прямоугольного треугольника отсутствовала, а для тупоугольного – присутствует проверка того, что это треугольник?
3. Доказать, что для остроугольного треугольника можно также не проводить проверку «на треугольник».
4. Написать на Прологе правила для определения равнобедренного и правильного треугольника.
5. Ниже приведено неправильное решение задачи определения прямоугольного треугольника. Найти ошибку в этом решении. Символ % обозначает комментарий – строка, которая начинается с этого символа, машиной не обрабатывается.

```
% эти правила определяют стороны треугольника
стороны(x,a,b,c)<-кандидат(x,a,_,_) , кандидат(x,_,b,_) , кандидат(x,_,_,c);
сторона(x,a)<-кандидат(x,a,_,_);
сторона(x,b)<-кандидат(x,_,b,_) ;
сторона(x,c)<-кандидат(x,_,_,c);
```

```
% это правило проверки треугольника
треугольник(x)<-стороны(x,a,b,c) , СЛОЖЕНИЕ(a,b,d) , БОЛЬШЕ(d,c);
нетреугольник(x)<-кандидат(x,a,b,c) , НЕ(треугольник(x));
```

```
% эти правила считают квадраты сторон треугольника
кв_стороны(x,a,d)<- стороны(x,a,_,_) , квадрат(a,d);
кв_стороны(x,b,d)<- стороны(x,_,b,_) , квадрат(b,d);
кв_стороны(x,c,d)<- стороны(x,_,_,c) , квадрат(c,d);
прямоугольный(x)<- кв_стороны(x,a,a2') , кв_стороны(x,b,b2') ,
кв_стороны(x,c,c2') , СЛОЖЕНИЕ(a2',b2',f) ,
РАВНО(f,c2');
```

## 12. Ввод и вывод данных на Прологе. Встроенные предикаты преобразования типов

В Прологе существуют специальные функции для ввода и вывода данных. Перечислим их:

**ПС** – переводит строку при вводе и выводе данных.

**ВВОДЦЕЛ (X)** – считывает целое число и присваивает его переменной X.

**ВВОДСИМВ (X)** – вводит строку и присваивает ее переменной X.

**ВЫВОД (A,B,C,...)** – выводит на экран значения переменных A, B, C, ...

Значением переменной может быть как строка, так и число.

Строки при выводе заключаются в кавычки.

Например, правило

**работа** <-**ВЫВОД**("Введите свое имя:"), **ВВОДСИМВ**(и),  
**ВЫВОД**("Доброе утро," ,и), **ПС**, **ВЫВОД**("Я рада видеть Вас");

и вопрос

? **работа**;

даст следующий результат:

**Введите свое имя: Наташа**  
**Доброе утро, Наташа**  
**Я рада видеть Вас**

Правила такого типа удобно использовать при организации диалога в программах на Прологе. Например, рассмотрим задачу проверки билета «на счастье». Счастливым билетом считается тот, который имеет какое-то определенное сочетание цифр. Например, в Москве счастливым билетом считают тот, у которого сумма первых трех цифр равна сумме трех последних, в Киеве - если сумма всех цифр кратна трем, в Санкт-Петербурге - если сумма цифр на четных местах равна сумме цифр на нечетных. В результате решения этой задачи получается следующая программа на Прологе:

%счастливыи билет;

**деление**(a, b, c)<-**УМНОЖЕНИЕ**(c, b, 0, a);

**работа**<-**ВЫВОД**("Введите шесть цифр вашего билета!"), **ПС**,  
**ВВОДЦЕЛ**(a), **ВВОДЦЕЛ**(b), **ВВОДЦЕЛ**(c), **ВВОДЦЕЛ**(d), **ВВОДЦЕЛ**(e),  
**ВВОДЦЕЛ**(f), **ПС**, **мсб**(a, b, c, d, e, f), **лсб**(a, b, c, d, e, f),  
**ксб**(a, b, c, d, e, f);

**мсб**(a, b, c, d, e, f)<-**СЛОЖЕНИЕ**(a, b, 0), **СЛОЖЕНИЕ**(0, c, p), **СЛОЖЕНИЕ**(d, e, k),  
**СЛОЖЕНИЕ**(k, f, l), **РАВНО**(w, l), **ВЫВОД**("Билет счастливый  
по-московски!");

**лсб**(a, b, c, d, e, f)<-**СЛОЖЕНИЕ**(b, d, v), **СЛОЖЕНИЕ**(v, f, w), **СЛОЖЕНИЕ**(a, c, y),  
**СЛОЖЕНИЕ**(y, e, z), **РАВНО**(w, z), **ВЫВОД**("Билет счастливый  
по-Санкт-Петербуржски!");

**ксб**(a, b, c, d, e, f)<-**СЛОЖЕНИЕ**(a, b, b2'), **СЛОЖЕНИЕ**(b2', c, c2'),  
**СЛОЖЕНИЕ**(c2', d, d2'), **СЛОЖЕНИЕ**(d2', e, e2'), **СЛОЖЕНИЕ**(e2', f, s),  
**деление**(s, 3, z1'), **ВЫВОД**("Билет счастливый по-киевски!");

?**работа**;

Правила "мсб", "лсб" и "ксб" проверяют, счастливый ли билет по различным критериям. Правило "работа" вводит исходные данные и вызывает правила проверки.

Кроме перечисленных встроенных предикатов в программах на Прологе можно использовать следующие функции для обработки строк:

**СЦЕП**(X, Y, Z) – сливает воедино строки X и Y и записывает результат в строку Z.

**ДЛИНА**(S, L) – подсчитывает длину строки S и записывает результат в переменную L.

**КОПИЯ**(S, N, L, K) – из исходной строки S, начиная символа с номером N, выделяет L символов и записывает в строку K.

**СТРЦЕЛ**(S, N) – преобразует целое число в строку, или наоборот.

Например, рассмотрим задачу перевода натуральных чисел от 1 до 100 в слова. Сначала «научим» машину переводить единицы. Для этого введем в базу факты:

единицы(1,"один");  
 единицы(2,"два");  
 единицы(3,"три");  
 единицы(4,"четыре");  
 единицы(5,"пять");

единицы(6,"шесть");  
 единицы(7,"семь");  
 единицы(8,"восемь");  
 единицы(9,"девять");

Для перевода чисел будем использовать правила "перевод". Для единиц это правило будет выглядеть так:

**перевод(x, y) <- МЕНЬШЕ(x, 10), единицы(x, y);**

Затем запишем правила перевода чисел второго десятка: от 11 до 19. Конечно, можно написать правила, аналогичные переводу чисел до девяти. Однако, нетрудно заметить, что правила образования чисел от четырнадцати до девятнадцати одинаковы: отделяются единицы, берется соответствующее название (например, "пять"), последняя буква отбрасывается ("пят") и добавляется окончание "надцать" ("пятнадцать"). Поэтому для перевода этих чисел можно записать следующее правило:

**перевод(x, y) <- БОЛЬШЕ(x, 13), МЕНЬШЕ(x, 20), УМНОЖЕНИЕ(1, 10, z, x),  
 единицы(z, t), ДЛИНА(t, l), СЛОЖЕНИЕ(k, 1, l),  
 КОПИЯ(t, 1, k, d), СЦЕП(d, "надцать", y);**

Это правило работает так:

1. Определяется, что число лежит в интервале от 14 до 19:

**БОЛЬШЕ(x, 13) "и" МЕНЬШЕ(x, 20);**

2. Отделяются единицы и записываются в переменную z:

**УМНОЖЕНИЕ(1, 10, z, x);**

3. По правилу "единицы" получается слово, соответствующее им:

**единицы(z, t);**

4. Определяется длина этого слова и из нее вычитается 1:

**ДЛИНА(t, l), СЛОЖЕНИЕ(k, 1, l);**

5. Выделяется подстрока без последней буквы (на одну букву короче):

**КОПИЯ(t, 1, k, d);**

6. Добавляется "надцать":

**СЦЕП(d, "надцать", y);**

Для перевода остальных чисел от 11 до 19 можно просто записать факты:

**перевод(11, "одиннадцать");  
 перевод(12, "двенадцать");  
 перевод(13, "тринадцать");**

Теперь запишем правила перевода десятков: 10, 20, ... 100. Правило "перевод" в этом случае будет выглядеть так:

**перевод(x, y) <- БОЛЬШЕ(x, 9), УМНОЖЕНИЕ(z, 10, 0, x), десятки(z, y);**

То есть если число кратно 10, то берется количество десятков и переводится с помощью правила или факта "десятки".

Для перевода чисел 10, 40, 90, 100 снова можно записать следующие факты:

**десятки(1, "десять ");**

**десятки(9, "девяносто ");**

**десятки(4, "сорок ");**

**десятки(10, "сто ");**

(Обратите внимание на пробел в конце десятков: он нужен для того, чтобы при выводе числа единицы и десятки отделялись друг от друга).

Для перевода 20 и 30 нужно к количеству десятков добавить окончание "дцать". Правило будет такое:

**десятки(x, y) <- БОЛЬШЕ(x, 1), МЕНЬШЕ(x, 4),  
единицы(x, t), СЦЕП(t, "дцать ", y);**

Для целых десятков от 50 до 80 также есть общее правило превода: надо взять количество десятков и добавить к нему окончание "десять". На Прологе это правило запишется так:

**десятки(x, y) <- БОЛЬШЕ(x, 4), МЕНЬШЕ(x, 9),  
единицы(x, t), СЦЕП(t, "десять ", y);**

И наконец, запишем правило перевода всех остальных чисел. Для этого используем уже написанные правила перевода десятков и единиц:

**перевод(x, y) <- БОЛЬШЕ(x, 20), УМНОЖЕНИЕ(z, 10, a, x), МЕНЬШЕ(a, 10),  
десятки(z, p), единицы(a, q), СЦЕП(p, q, y);**

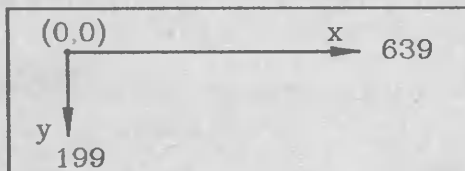
А теперь можно добавить правило "работа":

**работа <- ВЫВОД("Введите число от 1 до 100!"), ПС,  
ВВОДЦЕЛ(x), перевод(x, y), ПС, ВЫВОД("Получили ", y);**

и задав вопрос: ?работа; убедиться, что задача полностью решена.

### 13. Встроенные графические функции или рисование на Прологе

В Прологе-Д есть специальные встроенные функции, предназначенные для рисования. Расположение координатных осей и размеры экрана для IBM PC следующие:



(Примечание: размеры экрана некоторых модификаций IBM PC могут отличаться от приведенных.)



Приведем перечень встроенных графических функций и рассмотрим примеры их использования:

*1. Установка цвета фона и цвета текста.*

**ФОН(a, b);**                    a, b - числа (от 0 до 15) или переменные.  
a - цвет фона, b - текста.

Если a или b больше 16, то берется значение по модулю 16. Номера цветов приведены ниже:

<b>0</b> – черный;	<b>8</b> – темно-серый;
<b>1</b> – синий;	<b>9</b> – ярко-синий;
<b>2</b> – зеленый;	<b>10</b> – ярко-зеленый;
<b>3</b> – голубой;	<b>11</b> – ярко-голубой;
<b>4</b> – красный;	<b>12</b> – розовый;
<b>5</b> – фиолетовый;	<b>13</b> – малиновый;
<b>6</b> – коричневый;	<b>14</b> – желтый;
<b>7</b> – светло-серый;	<b>15</b> – белый.

*2. Открытие графического режима*

**ЗАПИСЬ\_В ("grp:")**

Возврат в текстовый режим:

**ЗАПИСЬ\_В ("cop:")**

*3. Установка точки.*

**ТОЧКА(x, y, c);** x, y, c – числа или переменные.

На экране ставится точка с координатами (x, y) цвета c.

*4. Отрезок прямой.*

**ЛИНИЯ(x, y, u, v, c);** x, y, u, v, c – числа или переменные.

Проводится отрезок прямой линии из точки с координатами (x, y) в точку с координатами (u, v) цвета c.

*5. Окружность.*

**ОКРУЖНОСТЬ (x, y, r, c);**

Рисуется окружность радиуса r с центром в точке (x, y) цвета c.

*6. Закраска замкнутой области.*

**ЗАКРАСКА(x, y, c);**

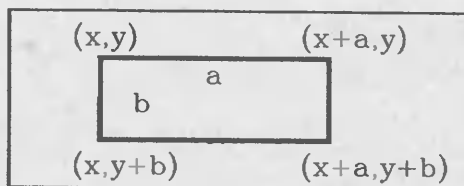
Закраска цветом c контура, внутри которого находится точка с координатами (x, y).

Графические функции, можно использовать в вопросах. Например, если задать вопрос:

**?ЗАПИСЬ\_В ("grp:"), ТОЧКА(120, 56, 15);**

то машина нарисует белую точку с координатами (120, 56). Функции можно использовать и в правилах. Напишем правило для рисования прямоугольника со сторонами a и b,

параллельными сторонам экрана. Положение такого прямоугольника может определяться по координатам любой вершины. Выберем левую верхнюю. Тогда координаты остальных вершин будут такие:



Обозначим  $x + a = d$ ,  $y + b = e$ . Правило на Прологе получится следующее:

прямоугольник( $x,y,a,b,c$ ) <- ЗАПИСЬ\_В ("gr:"), СЛОЖЕНИЕ( $x,a,d$ ),  
СЛОЖЕНИЕ( $y,b,e$ ), ЛИНИЯ( $x,y,d,y,c$ ), ЛИНИЯ( $d,y,d,e,c$ ),  
ЛИНИЯ( $d,e,x,e,c$ ), ЛИНИЯ( $x,e,x,y,c$ );

Чтобы получить на экране прямоугольник, достаточно ввести это правило в базу знаний и задать машине вопрос, например

?прямоугольник(10,20,60,90,2);

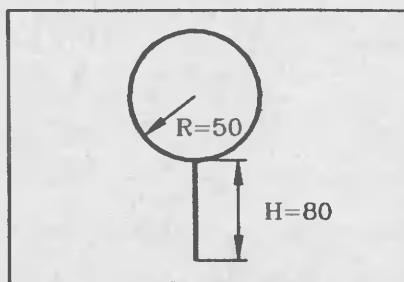
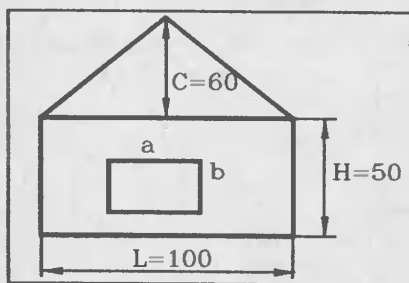
При этом машина нарисует зеленый прямоугольник с левой верхней вершиной в точке (10,20) и сторонами 60 и 90.

#### ЗАДАЧА

Написать программу на Прологе для изображения:

а) домика; б) дерева;

с заданными параметрами:



#### 14. Метод математической индукции

Метод математической индукции по сути является дедуктивным методом: при помощи этого метода общая закономерность доказывается для всех возможных (частных) случаев. Например, пусть дано утверждение: "За каждой женщиной в очереди также стоит женщина". Попытаемся ответить на вопросы: "Какой вид может иметь эта очередь?" и "Можно ли утверждать, что эта очередь состоит из одних женщин?"

Очевидно, что эта очередь может иметь один из следующих видов:

а) Состоять из одних мужчин:



б) В голове очереди может стоять несколько мужчин, а за ними – только женщины:



в) В очереди могут стоять и одни женщины:



Для того, чтобы можно было сказать, что эта очередь состоит только из женщин, нам необходимо знать, что **первой в очереди стоит женщина**.

Теперь перейдем непосредственно к методу. Метод математической индукции применяется для доказательства утверждений, зависящих от параметра (обозначим такое утверждение  $P(n)$ , где  $n$  – дискретный параметр, имеющий некоторое начальное значение).

Применение метода состоит из двух шагов: базиса и индуктивного перехода. В базисе мы проверяем, что для некоторого начального значения  $n$  (обозначим его  $n_0$ ) наше утверждение верно. Затем мы производим индуктивный переход, то есть предположив, что наше утверждение справедливо для некоторого значения  $n$  ( $n=k$ ), доказываем, что и для  $n=k+1$  наше утверждение останется верным.

Применение метода при доказательстве некоторого утверждения можно представить в виде схемы:

**1. В явном виде сформулировать УТВЕРЖДЕНИЕ, зависящее от параметра  $P(n)$ .**

**2. Б А З И С.** Проверить справедливость утверждения для начального значения параметра  $n_0$ .

**3. ИНДУКТИВНЫЙ ПЕРЕХОД.**

**ДАНО:**  $P(k)$  – верно.

**ДОКАЗАТЬ:**  $P(k+1)$  останется верным.

После этого наше утверждение считается доказанным, так как для  $n_0$  мы его доказали (базис), а для  $n_0+1$  и всех следующих значений  $n$  утверждение верно в силу индуктивного перехода.

Рассмотрим примеры применения этого метода:

*Пример 1:*

**УТВЕРЖДЕНИЕ:**  $7^{2n} - 1$  делится на 48 при натуральных  $n$ .

**Б А З И С:** Проверим, что при  $n=1$  утверждение верно:  
 $7^{2 \cdot 1} - 1 = 48$  - делится на 48.

**ИНДУКТИВНЫЙ ПЕРЕХОД:**

**ДАНО :**

$$7^{2k} - 1 \text{ делится на } 48.$$

**ДОКАЗАТЬ:**

$$7^{2(k+1)} - 1 \text{ также делится на } 48.$$

**Доказательство:**

$$\begin{aligned} 7^{2(k+1)} - 1 &= 7^{2k+2} - 1 = 49 \cdot 7^{2k} - 1 = 49 \cdot 7^{2k} - 7^{2k} + 7^{2k} - 1 = \\ &= 7^{2k} (49-1) + (7^{2k} - 1) \end{aligned}$$

Первое слагаемое кратно 48 (очевидно), и второе также кратно 48 (поскольку это уже дано). Следовательно, вся сумма также будет кратной 48.

Утверждение доказано.

*Пример 2:*

**УТВЕРЖДЕНИЕ:** Сумма каждых трех последовательных натуральных чисел делится на 3.

**Б А З И С:** Проверим, что для первой тройки натуральных чисел  $n_1=1, n_2=2, n_3=3$  утверждение верно:  
 $n_1 + n_2 + n_3 = 1 + 2 + 3 = 6$  - делится на 3.

**ИНДУКТИВНЫЙ ПЕРЕХОД:**

**ДАНО:** Для тройки натуральных чисел  $n_1 = k, n_2 = k+1, n_3 = k+2$  верно, что  $n_1 + n_2 + n_3 = 3k+3$  делится на 3.

**ДОКАЗАТЬ:** Для следующей тройки натуральных чисел  $n_1=k+1, n_2=k+2, n_3=k+3$  верно, что  $n_1 + n_2 + n_3$  делится на 3.

**Доказательство:** Вычислим сумму:

$$n_1 + n_2 + n_3 = (k+1) + (k+2) + (k+3) = 3k + 6 \text{ делится на } 3.$$

Утверждение доказано.

При использовании метода математической индукции необходимо строго следить за правильностью всех рассуждений. В противном случае возможны "доказательства" ошибочных утверждений.

Рассмотрим, например, следующее "доказательство":

**УТВЕРЖДЕНИЕ :** "В автобусе может поместиться сколь угодно много людей."

**БАЗИС:** При  $n=1$  утверждение верно: "В автобусе может поместиться один человек".

**ИНДУКТИВНЫЙ ПЕРЕХОД:**

**ДАНО:** "В автобусе может поместиться  $k$  человек".

**ДОКАЗАТЬ:** "В автобусе может поместиться  $k+1$  человек".

**"Доказательство":** Если в автобусе уже находится  $k$  человек, то еще один уж как-нибудь поместится.

Очевидно, что здесь неверно проведен индуктивный переход (автобус не резиновый, и есть предельное число людей, которые в него смогут поместиться).

Ошибку в приведенном ниже "доказательстве" предлагаем найти читателю:

**УТВЕРЖДЕНИЕ:** "Каждые  $n$  кошек - одного цвета.  
(Считаем, что одна кошка может иметь только один цвет.)

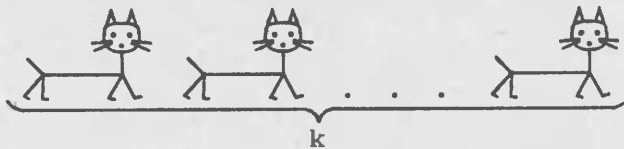
**БАЗИС:** При  $n=1$  утверждение верно, так как мы считаем, что одна кошка может иметь только один цвет.

**ИНДУКТИВНЫЙ ПЕРЕХОД:**

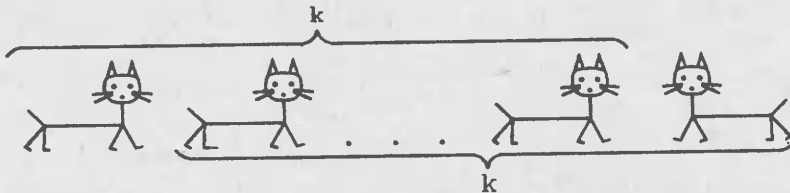
**ДАНО:** "Каждые  $k$  кошек одного цвета".

**ДОКАЗАТЬ:** "Каждые  $k+1$  кошек также будут одного цвета".

**"Доказательство":** Возьмем  $k$  кошек одного цвета:



и "добавим" к ним еще одну "новенькую":



Так как нам дано, что каждые  $k$  кошек будут одного цвета, мы можем рассмотреть  $k-1$  "первых" кошек, и "новенькую". Относительно них мы можем утверждать, что у них один цвет. Но  $k$ -я кошка была того же цвета. Следовательно, и все  $(k+1)$  кошки будут того же цвета.

**Задачи и упражнения:**

*Доказать методом математической индукции следующие утверждения:*

1.  $n^3 + 3n^2 + 5n + 3$  кратно 3 при всех натуральных  $n$ .

2. Квадрат любого числа  $k$  равен сумме  $k$  первых нечетных чисел.

3. При всех нечетных натуральных  $n^3 - n$  кратно 24.

4. При всех натуральных  $4^n + 15n - 1$  кратно 9.

5. При натуральных  $n > 5$   $2^n > n^2$ .

$$6. \frac{1}{1 * 2} + \frac{1}{2 * 3} + \frac{1}{3 * 4} + \dots + \frac{1}{n * (n + 1)} = \frac{n}{n + 1} \quad \text{для натуральных } n.$$

$$7. \frac{1}{6} + \frac{1}{12} + \frac{1}{20} + \dots + \frac{1}{n^2 + 3n + 2} = \frac{n}{2n + 4} \quad \text{для натуральных } n.$$

$$8. 2 + 6 + 12 + \dots + n(n-1) = \frac{n * (n + 1) * (n + 2) * (n + 3)}{3}$$

для натуральных  $n$ .

### 15. Рекуррентные соотношения

В школьном учебнике математики дается следующее определение понятия  $a$  в степени  $n$ : " $n$ -й степенью числа  $a$  называется произведение этого числа на себя  $n$  раз:

$$a^n = \underbrace{a * a * \dots * a}_n \quad \text{п штук.}$$

Это определение (и все подобные ему) следует признать недостаточно строгим: из него не следует, чему будет равно  $a^1$ , а тем более  $a^0$ .

Для того, чтобы дать строгое определение понятия (натуральной) степени числа, следует вспомнить, что числа  $a^0, a^1, a^2 \dots$  образуют последовательность. (Напомним, что последовательность - числовая функция, область определения которой - числа расширенного натурального или натурального ряда ( $N_0$  или  $N$ ):  $0, 1, 2, 3 \dots$ ).

Выведем рекуррентное соотношение, при помощи которого можно получить все натуральные степени каждого числа. По определению

$$\begin{cases} a^0 = 1 \\ a^n = a^{n-1} * a, \text{ при } n > 0, n \text{ из } N_0 \end{cases}$$

**РЕКУРРЕНТНОЕ СООТНОШЕНИЕ** - соотношение, выражающее последующий член последовательности через один или несколько предыдущих.

Таким образом, мы можем дать следующее определение натуральной степени числа:

Структура этого определения тесным образом связана с методом математической индукции: определение  $a^0$  соответствует базису, а определение  $a^{n+1}$  через  $a^n$  аналогично индуктивному переходу.

**УПРАЖНЕНИЕ:** Сформулируйте рекуррентное определение факториала  $n! = 1 * 2 * 3 \dots n$ .

(Продолжение следует)

А.Г. Юдина

## НАЧАЛА АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ В 9 КЛАССЕ\* (2 урока в неделю)

### ЗАНЯТИЯ 25-26

#### Повторение

#### Команды и процедуры с параметрами и без

Графические команды, команды ввода-вывода, расчётные и диалоговые программы.

#### Задачи для повторения:

- 1) пирамиды из квадратов, прямоугольников, треугольников (ёлочка);
- 2) снеговик;
- 3) перо переменного размера;
- 4) шахматная доска переменного размера;
- 5) "цветы" с различным количеством и формой лепестков;
- 6) "венки" из таких цветов;
- 7) звуковая дорожка;
- 8) задачи из занятий 23 и 24.

### РАЗВЕТВЛЯЮЩИЕСЯ АЛГОРИТЫ. УСЛОВНЫЕ ВЕТВЛЕНИЯ

#### ЗАНЯТИЕ 27

#### Команды условного ветвления IF и IFELSE

Примеры последовательных и разветвляющихся алгоритмов, запись их в виде блок-схем. Полное и неполное ветвление. Структура команд ветвления:

неполное:

```
IF      [   ]
```

если условие то

полное:

```
IFELSE [   ] [   ]
```

если <условие> то иначе

#### Задание

Написать программу, которая:

- а) спрашивает, какие у Вас вес и рост (две отдельные переменные) и сообщает, находится ли Ваш вес в пределах нормы;
- б) спрашивает у покупателя, в какой вагон нужен билет - плацкартный, купейный или общий, и называет соответствующую цену.

### РЕКУРСИЯ

#### ЗАНЯТИЕ 28

Рекурсия (процедура, обращающаяся к самой себе)

Примеры рекурсии «из жизни»: конфета «А ну-ка, отними!» (на фантике нарисована девочка, держащая конфету, на которой нарисована девочка, держащая конфету... и т.д.), «У попа была собака...», зеркала, расположенные друг против друга.

Вопрос: каков результат работы следующей программы:

```
TO A
REPEAT 4 [FD 50 RT 90]
END
```

Ответ известен. А такой программы:

```
TO A
REPEAT 4 [FD 50 RT 90] A
END?
```

\*Окончание (начало см. в ИНФО №4-93 г.).

Объяснение и демонстрация "бесконечного" движения черепашки, которое можно прервать, нажав одновременно клавиши CTRL+S. А если менять параметр в REPEAT: 3, 2, 1 - ничего не меняется. Значит, для достижения того же результата достаточно

```
TO A
FD 50 RT 90 A
END
```

### Задание

Написать программу, которая до прерывания:

а) двигает черепашку по треугольной, пятиугольной, шестиугольной, круглой траекториям;

б) печатает стихотворение про попа и собаку.

Следующий шаг - ввести переменный параметр, чтобы черепашка могла бегать по квадрату любого размера:

```
TO A :X
FD :X RT 90 A :X
END
```

А что получится, если

```
TO A :X
FD :X RT 90 A :X + 3
END
```

Знакомая уже квадратная спираль, только "бесконечная".

### Задание

Используя рекурсию и прерывание:

а) получить на экране спирали различной формы, увеличивающиеся и уменьшающиеся;

б) проиграть "звуковую дорожку";

в) вывести на экран натуральный ряд.

## ЗАНЯТИЕ 29

### Простейший графический редактор

«Рисовалка» - программа, позволяющая двигать черепашку нажатием на различные клавиши.

### Задание

Написать программу, которая ожидает ввода символа (нажатия на какую-либо клавишу).

Если будет нажата клавиша "Q" - поворачивает черепашку против часовой стрелки на 90 градусов; "W" - поворачивает черепашку по часовой стрелке на 90 градусов; "E" - продвигает черепашку на 20 шагов вперёд; Если будет нажата любая другая клавиша, ничего не делает и заканчивает работу.

```
(TO рисовалка
MAKE "K READCHAR
IF :K = "Q [LT 90]
IF :K = "W [RT 90]
IF :K = "E [FD 20]
END)
```

Программа малоинтересная - выполняет всего одно какое-то действие (в лучшем случае). А если добавить в конце команду A (обращение к самой себе), можно будет рисовать, пока не надоест.

### Задание

Написать "рисовалку", расширив набор управляющих клавиш, и нарисовать любую картинку.

(Самостоятельное определение не слишком большого набора действий, достаточно удобного для рисования: необходимы повороты направо и налево на малый угол и на 90 градусов (или сразу рисование дуг различной кривизны), очистка экрана, заливка цветом, возможность поднять-опустить перо и т. д. ).

## ЗАНЯТИЕ 30

Как написать рекурсивную программу, чтобы она вовремя остановилась? Использовать условие.

### Разбор задачи

Написать процедуру, печатающую числа, кратные трём, начиная с заданого и до 1000.

Рекурсивная "бесконечная" процедура:



```

ТО ряд :X
PR :X ряд :X + 3
END

```

В процедуру необходимо вставить проверку условия - не достигла ли переменная :X числа 1000. Какая команда осуществляет проверку условий? Команда условного выполнения IF (и IFELSE).

```

ТО ряд :X
IF :X < 1000 [PR: X ряд: X + 3]
END

```

Теперь как только переменная достигнет 1000, действия в скобках выполняться не будут и программа завершится.

(Другой вариант:

```

ТО ряд :X
IF :X < 1000 [STOP]
PR :X ряд:X + 3
END)

```

### Задание

Написать рекурсивную программу, которая:

а) выводит натуральные числа, начиная с заданного числа и с заданным шагом (2 переменных параметра), до 0.

б) рисует спираль - треугольную, круглую («улитка») - не «вылезаящую» за назначенную границу. (В улитке трудно подобрать начальный шаг и смещение - они должны быть малыми, но не слишком - иначе «не хватает памяти»).

Для выполнения этого задания понадобятся две новые команды: команды датчики Y и X возвращают текущие координаты черепашки; т.е. при их выполнении образуется число, которое необходимо передать какой-либо команде (какой? PR, MAKE - так же, как с командами ввода). Перед работой со спиралью попробовать новые команды X и Y.

в) переписать «рисовалку» так, чтобы из неё можно было выйти нажав, например, клавишу "K (команда STOP).

## ЗАНЯТИЕ 31

### Управляемая рекурсия (продолжение)

Эта программа рисует увеличивающуюся спираль, которая останавливается, перейдя границу, заданную линией  $Y = 70$ :

```

ТО спираль1 :X
FD :X RT 121
IF Y < 70 [спираль1 :X + 2]
END

```

(Спираль выбрана треугольная "винтовая" - для красоты). Условие остановки программы - расстояние черепашки до оси абсцисс. А если в программе значение переменной не увеличивать, а уменьшать?

```

ТО спираль2 :X
FD :X RT 121
спираль 2 :X - 2
END

```

При выполнении этой процедуры (например, спираль2 60) спираль сначала «сворачивается» - переменный параметр уменьшается от 60 до 0. А затем - опять «разворачивается» - только в другую сторону ( $FD - 20 = BK 20$ ). Происходит это довольно быстро. Чтобы увидеть момент, когда значение :X достигает нуля и, продолжая уменьшаться, становится отрицательным, можно «растянуть» процесс, добавив паузу:

```

ТО спираль2 :X
FD :X RT 121 WAIT 5
спираль 2 :X - 2
END

```

(Или использовать команду SLOWTURTLE). Как же остановить черепашку в центре спирали? Программа должна выполняться, пока переменная :X больше нуля:

```

ТО спираль2 :X
FD :X RT 121
IF :X < 0 [спираль2 :X - 2]
END

```

Процедуры спираль1 :X и спираль2 :X (в

окончательном виде) очень похожи, но критерии остановки программ совершенно разные: в первом случае это ордината черепашки, а во втором - текущее значение переменного параметра.

### Задание

1) «бесконечная» уменьшающаяся спираль;

2) уменьшающиеся только до центра спирали треугольная и круглая («обратная улитка» - см. примечание к «прямой» улитке);

3) спираль с двумя переменными параметрами - шаг и изменение шага;

4) обратная улитка с «музыкой».

### ДВА ТИПА ЦИКЛОВ: "N РАЗ" И "ПОКА"

(с использованием команды REPEAT и рекурсивные)

## ЗАНЯТИЕ 32

### Тренажёр «Таблица умножения»

#### Задание

а) написать программу, которая даёт Вам задание перемножить два произвольных натуральных числа до 15, получает ответ и сообщает - правильно или нет.

#### Сценарий программы:

1) компьютер должен загадать два случайных числа. Необходима новая команда - RANDOM ?. Это команда даёт случайное целое число в диапазоне от 0 до указанного параметра.

2) программа должна задать вопрос (нужны команды вывода);

3) затем - получить ответ (а это - команды ввода);

4) проанализировать - правильный был ответ или нет - и выдать соответствующее сообщение (команды условного ветвления и вывода).

Программу можно написать в виде процедуры с двумя переменными параметрами - сомножителями:

```

ТО пример :K :L
(PR [Сколько будет] :K [*] :L [?])
MAKE "Z FIRST READLIST
IFELSE :Z - :K * :L
[PR [Молодец!]] [PR [К сожалению, не так...]]
END

```

А вызывать эту процедуру так:

пример RANDOM 15 RANDOM 15

б) усложнить - ввести счёт правильных и неправильных ответов, и написать программу-тренажёр.

```

ТО пример :K :L
(PR [Сколько будет] : K [*] : L [?])
MAKE "Z FIRST READLIST
IFELSE :Z - :K * :L
[PR [Молодец!]] MAKE "счёт_прав :счёт_прав+1]
[PR [К сожалению, не так...]]
MAKE "счёт_неправ :счёт_неправ+1]
END

```

Возможны два варианта тренажёра - 10 вопросов и отметка (цикл "N раз" - через REPEAT); и рекурсивный - пока не остановишь (как?).

```

ТО тренажёр_1
MAKE "счёт_прав 0 MAKE "счёт_неправ 0
REPEAT 10 [пример RANDOM 15 RANDOM 15]
оценка
END

```

```

ТО оценка
СТ
(PR [Ваша оценка -])
ROUND :счёт_прав * 5 / (:счёт_прав + :счёт_неправ)
END

```

```

ТО тренажёр_2
MAKE "счёт_прав 0 MAKE "счёт_неправ 0
цикл_оценки
END

```

```

ТО цикл
СТ
пример RANDOM 15 RANDOM 15
(PR [СЧЁТ -] :счёт_прав [:] :счёт_неправ)
PR [Если хотите закончить, нажмите S!
Для продолжения - любую клавишу.]
MAKE "клавиша READCHAR
IF :клавиша = "C [STOP]
цикл
END

```

### ЗАНЯТИЕ 33

#### Построение графиков функций

Все рисунки до сих пор получались на экране в результате выполнения команд FD, BK, RT и LT, которые перемещают и поворачивают черепашку относительно её предыдущего положения. Это команды относительного перемещения, результат их работы зависит от предыдущих команд.

А есть ещё и группа команд абсолютного перемещения черепашки. В неё входят

SETX число - перемещает черепашку по горизонтали до точки с заданной абсциссой;

SETY число - перемещает черепашку по вертикали до точки с заданной ординатой.

Каждая точка экрана имеет координаты, начало отсчёта - в центре. Именно команды абсолютного перемещения естественно использовать для построения графиков функций. Во всех заданиях по этой теме ученики должны использовать только их вместо FD, RT и LT. (Полным набор команд абсолютного перемещения делает команда SETPOS число1 число2, но в этом разделе можно обойтись без неё).

#### Задание

а) написать процедуру, рисующую координатные оси;

б) написать процедуру, рисующую произвольный прямоугольник во второй четверти;

в) написать процедуру, размечающую координатные оси с некоторым шагом; последняя программа - четыре цикла с переменным параметром: разметка от центра вправо, влево, вверх и вниз. Например:

```
MAKE "X 10
```

начальная  
установка  
переменного  
параметра

```
REPEAT 10 '
[SETX :X SETY 2 SETY 0
MAKE "X :X + 10]
```

тело цикла  
изменение  
переменного  
параметра  
в цикле

(При написании программы, состоящей из четырёх однотипных кусков, надо, конечно, использовать копирование - F1, F3, F4).

г) ввести в предыдущую процедуру переменный параметр - шаг разметки, и связать с ним число повторений каждого цикла.

### ЗАНЯТИЕ 34

Во всех предыдущих графических программах рисунок возникал в результате перемещения черепашки (с опущенным пером). Другой способ - получить на экране отпечатки черепашки - команда STAMP. Вид черепашки и её отпечатков меняет команда SETSH число (установить форму номер ...), а выбрать форму можно, дав команду SHAPES.

#### Задание

а) написать программу «звёздное небо» - на синем фоне в беспорядке разбросаны жёлтые точки.

Эта программа циклическая. Начальная установка - установить цвет неба (поменять цвет черепашки SETC ? и залить - FILL), установить форму - точку и цвет черепашки - жёлтый.

Цикл - повторить много раз следующие действия: поднять перо, переместить черепашку в случайное место, опустить перо, оставить след - точку.

(Перемещение в случайное место - два действия:

```
SETX RANDOM ? SETY RANDOM ?).
```

```
TO небо
```

```
RG SETC 5 FILL SETSH 10 SETC 14
```

нач.  
уста-  
новки

```
REPEAT 100
```

```
[PU SETX RANDOM 300
```

```
SETY RANDOM 300 PD STAMP]
```

тело  
цикла

```
END
```

(Можно добавить «космическую музыку»: после появления каждой звёздочки - одну ноту случайной высоты.);

б) написать программу «Лес» - на вы-

бранном фоне в случайных местах растут разноцветные деревья;

в) написать программу «Пейзаж» - экран поделён пополам, выше линии горизонта небо и звёзды, ниже - лес.

Механически соединить две предыдущие программы мешает то, что звёзды появляются в лесу, а деревья вырастают на небе. Необходимо предотвратить это - сделать команду STAMP условной:

черепашка-дерево может оставить след только на фоне «земли», а черепашка-точка - только на фоне неба:

```
TO звёзды
SETC 14
SETSH 10
REPEAT 100
[PU SETX RANDOM 300 SETY RANDOM 300 PD
IF COLORUNDER - 5 [STAMP] ]
END
```

```
TO лес
SETSH 23
REPEAT 100
[PU SETX RANDOM 300 SETY RANDOM 300 PD
SETC RANDOM 16
IF COLORUNDER - 6 [STAMP]]
END
```

```
TO пейзаж
RG SETC 5 SETX 400 "линия горизонта"
PU SETY 5 PD FILL PU
SETY -5 PD SETC 6 FILL
звёзды лес
END
```

### ЗАНЯТИЕ 35

Как строятся графики функций в тетради? - по точкам, которые затем соединяются. Если точки расположены достаточно часто, их можно не соединять - получится точечный график. Для каждой точки произвольно устанавливается значение абсциссы и вычисляется значение ординаты. По оси X точки удобно расположить равномерно, на небольшом расстоянии друг от друга, т. е. сделать абсциссу переменным параметром цикла. Программа должна быть похожа на программу «звёз-

дное небо», только там изображение состоит из точек, хаотически разбросанных по экрану и цикл простой, без параметра.

#### Сценарий программы:

а) начальные установки формы, цвета черепашки, начального значения переменного параметра - абсциссы первой точки графика;

б) в цикле повторяются следующие действия: вычислить по формуле значение ординаты, поднять перо, переместить черепашку в точку с текущими координатами, опустить перо, оставить след и изменить значения параметра цикла.

#### Задание

а) построить график функции  $y = 0,1 x^2$

```
(TO график
RG SETSH 10 SETC 14 "нач. установки
MAKE "X -30
REPEAT 60
[MAKE "Y :X * :X * 0.1
точка "тело цикла
MAKE "X :X + 1]
END
```

```
TO точка
PU SETX :X SETY :Y PD STAMP
END)
```

б) написать программу, рисующую координатные оси и три разноцветных точечных графика функций:

$$y_1 = 0,1 x^2$$

$$y_2 = 50 \sin 5x$$

$$y_3 = x \sin 5x$$

(Четыре отдельные процедуры: оси и графики трёх функций вызываются пятой, общей процедурой).

### ЗАНЯТИЕ 36

На предыдущем занятии три процедуры для трёх разных графиков были очень похожи. Менялась формула вычисления, а также приходилось подбирать начальное значение абсциссы и число точек, чтобы график «не вылезал за экран», и шаг -

чтобы он не был слишком «редким». Как автоматизировать этот подбор и сделать универсальную процедуру для разных графиков?

Во-первых, выделить вычисление ординаты в отдельную процедуру - «копилку» функций:

```

ТО функция :N :X
IF :N = 1 [MAKE "Y 0.1 * :X * :X]
IF :N = 2 [MAKE "Y 50 * SIN 5 * :X ]
IF :N = 3 [MAKE "Y :X * SIN 5 * :X ]
.....
END

```

«Копилку» можно сколько угодно расширять. Теперь номер функции будет переменным параметром основной процедуры. Также переменными параметрами будут шаг и цвет точек. График процедура пускай пытается построить на весь экран - от левого края до правого, но оставлять след черепашка имеет право, только если ордината - в пределах экрана (так же, как в программе "Пейзаж").

### Задание

а) написать процедуру с тремя переменными параметрами (номер функции, шаг и цвет) по сценарию:

Начальные установки:

установить форму и цвет черепашки, начальное значение абсциссы (левый край экрана).

Цикл (повторяется столько раз, сколько точек с заданным шагом уместятся в ширину экрана):

1) вычислить значение ординаты;

2) если значение ординаты не выходит за пределы экрана, поднять перо; переместить черепашку в точку с установленными координатами; опустить перо; оставить след;

3) изменить значение абсциссы - параметра цикла - на шаг.

(ТО график :N :шаг :цвет

SETC :цвет SETSH 10

MAKE "X -140

REPEAT 280 / :шаг

[функция :N :X

IF AND :Y > -85 :Y < 85 [точечка]

MAKE "X :X + :шаг]

END)

б) добавить в "копилку" функции

$$y_4 = 50 \sin x + \cos 15x$$

$$y_5 = 50 \sin 3x + \cos 25x$$

и построить их графики.

## ЗАНЯТИЕ 37

### Изменение масштаба графика

Если попробовать построить графики функций  $y=x^2$  или  $y=\ln x$ , ничего хорошего не получится - слишком мелкое изображение в естественном масштабе 1 единица = 1 шаг черепашки. Как изменять масштаб, т.е. растягивать или сжимать при необходимости графики по осям?

Пусть мы хотим «растянуть» график по осям в M раз. В первую очередь необходимо изменить процедуру, ставящую точку:

ТО точечка

```

PU SETX :X * :M SETY :Y * :M PD STAMP
END

```

Во-вторых, необходимо в M раз уменьшить начальное значение абсциссы, а также верхнюю и нижнюю границы ординат.

Теперь можно строить графики с нужным увеличением. Но вот беда - чем сильнее увеличение, тем реже точки графика. Чтобы плотность точек не изменялась, нужно пропорционально уменьшить шаг. И вот модифицированная процедура, строящая графики:

ТО график :N :шаг :цвет :M

SETC :цвет SETSH 10

MAKE "X - 140 / :M

REPEAT 280 / :шаг

[функция :N :X

IF AND :Y > -85 / :M :Y < 85 / :M

[точечка]

MAKE "X :X + :шаг / :M]

END

### Задание

а) ввести в универсальную процедуру четвертый переменный параметр - во сколько раз увеличить изображение;

б) предварить построение графика по-

строением размеченных осей (в одинаковом масштабе, разумеется);

в) подобрать масштаб для ф-ций и решить графически уравнения:

$$x^4 - 3x^2 - x + 0.5 = 0$$

$$2x^3 + x^2 - 2x + 1 = 0$$

**ЗАНЯТИЕ 38**

Любая программа, пусть самая сложная, состоит из фрагментов трёх типов: последовательных, разветвляющихся и циклов.

Циклы, в свою очередь, могут быть типа «N раз» - если можно рассчитать заранее количество повторов (такие циклы использовались, например, в программах построения графиков); если же невозможно установить количество повторов - цикл "Пока" - необходимо использовать управляемую рекурсию.

На рисунке изображена блок-схема цикла. Ветвления полные - используются команды IFELSE.

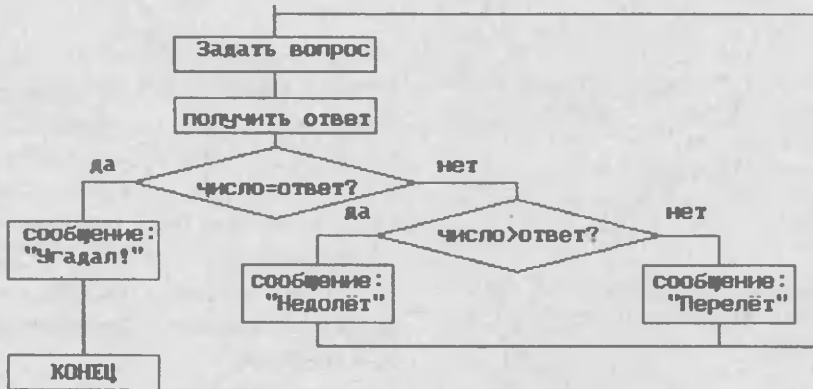
**Задание**

а) написать программу «Угадай-ка», которая:

- 1) загадывает случайное число до 1000;
- 2) задаёт вопрос ("Угадай число");
- 3) получает ответ;
- 4) если число угадано, сообщает об этом и заканчивает работу; в противном случае продолжает анализ (выдаёт сообщения "Недолёт" или "Перелёт") и возвращается к пункту 2. (п.1 - начальная установка; п.2-4 - тело цикла).

Так как невозможно предсказать количество повторов, цикл организуется с помощью рекурсии. Программа будет состоять из двух процедур: процедура начальной установки и рекурсивный цикл.

Ветвления полные - используются команды



```
(TO угадайка
MAKE "число RANDOM 1000
СТ попытка
END
```

```
TO попытка
PR [Угадай число!]
MAKE "ответ FIRST READLIST
IFELSE :число = :ответ
[PR [Угадал!]]
[IFELSE :число :ответ
[PR [Недолёт!]] [PR [Перелёт!]] попытка]
END)
```

б) ввести в программу счётчик попыток.

```
(TO угадайка
MAKE "число RANDOM 1000
MAKE "счётчик 0
СТ попытка
END

TO попытка
MAKE "счётчик :счётчик + 1
PR [Угадай число!]
MAKE "ответ FIRST READLIST
IFELSE :число > = :ответ
[(PR [Угадал! Число попыток - ] :счётчик)]
[IFELSE :число :ответ
[PR [Недолёт!]] [PR [Перелёт!]] попытка]
END
```

В заключении хочу пожелать успехов тем, кто начнет работать с замечательной средой обучения LogoWriter, а также сказать два слова об ошибках. На первых порах основная причина ошибок — несоблюдение следующих правил: знаки ариф-

метических операций и сравнения должны отделяться с двух сторон пробелами, а при указании имени переменной ("счетчик) или значенич переменной (:счетчик) после кавычек и двоеточия пробела быть не должно.

## Министерство образования Республики Беларусь

### АССОЦИАЦИЯ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

Проведение международных соревнований школьников по информатике ICI в Беларуси уже стало традицией. Эти соревнования проводятся Министерством образования Республики Беларусь и Ассоциацией ИНТО при участии корпорации IBM и международного союза ORT.

Очередные международные соревнования школьников по информатике ICI-94 пройдут с 16 по 22 мая 1994 года. К участию в ICI-94 приглашаются команды государств и республик, центров информатики, учебных заведений.

В состав команды могут быть включены до 4 участников, руководитель команды и его заместитель.

К участию допускаются школьники 93/94 учебного года, родившиеся после 1 сентября 1976 года. Увеличение числа участников от одной команды просим согласовать с оргкомитетом. Наблюдатели, родственники, преподаватели могут сопровождать команду.

Все расходы по участию в ICI-94 несут команды. Заявки на участие принимаются до 1 апреля 1994 года. Для получения более подробной информации обращайтесь в Оргкомитет.

### ОРГКОМИТЕТ

г.Минск

Тел.: (0172) 29-62-98, 29-64-01, 5-70-67

Факс: 3(0172) 1-90-50, 25-74-95

г.Москва

Тел.: (095) 320-53-89

Факс: (095) 324-30-55

И. Н. Антипов, О. А. Боковнев, М. Е. Степанов

## О ПРЕПОДАВАНИИ ИНФОРМАТИКИ В МЛАДШИХ КЛАССАХ

Важной проблемой, связанной с преподаванием информатики в общеобразовательной средней школе, является определение возраста учащихся, с которого целесообразно начинать изучение этого предмета. В статье предлагается один из возможных путей преподавания информатики в младших классах, во многом восходящий к идеям С. Пейперта [1].

Концентрируя внимание на интеллектуальном развитии учащихся, Пейперт вводит понятие среды обучения, в рамках которой можно приобщиться к таким фундаментальным достижениям как математика. Обучение при этом предстает самостоятельным творческим поиском, осуществляющимся в неформальном диалоге ученика с преподавателем. Компьютер и язык программирования Лого являются при этом универсальными средствами организации предметной области, в которой учащийся может ставить перед собой проблемы и разрешать их. Мы также будем использовать понятие среды обучения, интерпретируя его, однако, несколько иначе, чем Пейперт.

Любая программа, предполагающая диалог, в какой бы форме он не проводился, заставляет пользователя принять определенные правила работы с ней. При отказе от этих правил пользователь теряет контроль над компьютером. Таким образом, программа предстает перед нами как предметная область, подчиненная своим внутренним законам. Компьютерная предметность, поддерживаемая любой программой, создается с помощью подвижных изображений на экране, включая и различные виды общепринятых знаков,

таких как буквы и цифры, а также с помощью звуковых эффектов. Удачные программы вызывают у пользователя ощущения взаимодействия с совокупностью реально существующих предметов, подчиненных определенным законам. По этой причине мы будем в дальнейшем считать, что любая обучающая программа представляет собой реализацию той или иной формы предметности.

По виду управления программы можно разделить на две группы. К первой группе относятся программы с интерактивным интерфейсом, который можно рассматривать как реализацию непосредственного воздействия на компьютерную предметность. Если для диалога используется клавиатура, то нажатие на определенную клавишу или группу клавиш приводит к немедленным результатам, хотя они могут и не соответствовать желаниям пользователя.

Ко второй группе относятся программы, способные принимать от пользователя и исполнять отдельные команды и их совокупности, то есть в диалоге применяется особый язык, владение которым является гарантией успеха. Использование языка как бы отделяет пользователя от компьютерной предметности и заставляет его полностью спланировать свои действия еще до их начала. Формирование плана и его фиксация на специфическом языке в совокупности предстают как программирование.

Программирование, таким образом, имеет две стороны: узко специальную, связанную с изучением конкретного языка, и общезначимую, связанную с освоением предметной области, ее внутренних законов и методов планирования в рамках



этой предметности. Предметность должна быть освоена учащимся как идеальная форма и, по словам Э.В. Ильенкова, форма вещи должна преобразоваться в форму деятельности [2]. Выбор компьютерной предметности определяются при этом целями обучения. Ее идеальная форма может соответствовать математическим, физическим и другим видам знания. Основной упор при обучении должен быть сделан именно на превращение формы компьютерной предметности в нужную форму знаний.

Компьютерная предметность, обладая несомненной реальностью, тем не менее имеет определенную специфику, проистекающую из самого способа существования этой предметности в виде экранных изображений и из особенностей операторской деятельности пользователя. Даже при интерактивном режиме работ между учащимся и компьютерным предметом качества посредников появляются такие устройства как клавиатура, мышь, джойстик, световое перо и т.д.

Необычное соединение реального и иллюзорного в компьютерных программах приводит к тому, что у детей, вырастающих в тесном общении с компьютерами, меняются фундаментальные духовно-культурные структуры, понятия и представления, касающиеся жизни и смерти, одушевленности людей, животных и компьютеров [3]. А.И. Ракитов указывает, что у «компьютерного поколения» возможно смещение в восприятии реального и компьютерного миров и даже полное вытеснение первого последним. Особенно возрастает опасность при снижении возраста, с которого начинается знакомство с компьютером.

Возникающие при этом проблемы нуждаются в серьезном изучении, но уже сейчас можно утверждать, что среда обучения в младших классах должна органи-

зовываться таким образом, чтобы использование компьютерной предметности уравнивалось наличием ее материального эквивалента. Небезынтересно в этом отношении свидетельство Тошио Савады о том, что в компьютеризированной Японии широко используются японские счеты (соробан), и, возможно, именно этим объясняется рост количества способных математиков среди учащихся японских школ [4].

При выработке принципов формирования среды обучения, ориентированной на использование компьютерной предметности, особое внимание следует обратить на исследования отечественных ученых, посвященные роли предметной деятельности в процессе обучения. В статье «Деятельность и знание» Э. В. Ильенков подчеркивает: «Действительное мышление формируется в реальной жизни именно там – и только там, – где работа языка неразрывно соединена с работой руки – органа непосредственно-предметной деятельности... Только тут предмет и проявляет себя как вещь в себе, заставляя считаться с собой больше, чем словами и «визуализирующими» эти слова «схематизмами» [2]. Используя термины феноменологии «эйдос» и «логос» в трактовке А.Ф. Лосева: эйдос «есть смысловой лик вещи, созерцательно и умственно осязательно данная его фигура, логос же есть метод смыслового оформления вещи, задание мыслить вещь... /цит. по 5/, можно пояснить мысль Э. В. Ильенкова следующим образом. Цель обучения состоит в формировании логоса, т.е. умения мыслить вещь, но без наличия эйдоса (смыслового образа этой вещи) этого невозможно добиться. Эйдос же формируется в процессе предметной деятельности.

В исследованиях, специально направленных на изучение роли предметного действия в обучении, часто используется

понятие материализации. Его смысл поясняет Н. Ф. Талызина: «Как овладеть этими невидимыми действиями, которые человек совершает не руками. Не внешним образом, а про себя, в голове? На помощь нам приходят действия внешние, материальные. Они дают возможность невидимые внутренние действия сделать видимыми, показать их содержание учащимся, сделать их понятными» [6]. Как отмечает Н. Г. Салмина: «организация материализации должна исходить из принципа построения специфического действия, адекватного формируемому знанию» [7].

Таким образом, среда обучения, ориентированная на формирование определенного вида знаний, должна складываться из компьютерной предметности и материальной модели этой предметности. Назначение двух видов предметности состоит в следующем.

*1. Материальная модель позволяет учащемуся самостоятельно осуществлять достаточно простые предметные действия, и, тем самым, дает возможность начать освоение соответствующих мыслительных операций.*

*2. Работа в интерактивном режиме с обучающей программой позволяет перенести навыки с материальной модели на компьютерную предметность и связать компьютерный мир с миром реальным.*

*3. Использование языка программирования для планирования в сфере компьютерной предметности позволяет строить и исполнять значительно более сложные действия, чем это возможно на материальной модели. По этой причине введение языка открывает перед учащимся новый уровень возможностей.*

*4. Использование языка программирования применительно к материальной модели позволяет организовать содержательный диалог между учащимися, при этом каждый из них может ощутить*

*себя не только в роли программиста, но и в роли исполнителя.*

*5. Использование языка программирования приводит к тому, что фаза планирования, требующая предварительного осознания своих последующих действий становится необходимой, в то время как в материальной модели возможно не вполне осознанное манипулирование составными элементами модели. С другой стороны, наличие материальной модели позволяет более четко разделить процесс программирования на планирование в собственном смысле этого слова и на процесс фиксации плана на специальном языке.*

Для пояснения смысла этих положений используем пример, характер которого в значительной степени определяется ограниченным объемом статьи. Рассмотрим простейший вариант курса информатики для учащихся младших классов. Язык программирования позволяет строить изображения на плоскости, разбитой на клетки, и является крайне упрощенным аналогом графики языка Лого. Система команд демонстрируется с помощью рис. 1. Язык позволяет изображать на плоскости семейства геометрических фигур. Обучающая программа представляет собой графический редактор, допускающий интерактивный и командный режимы работы.

Материальной моделью компьютерной плоскости является разграфленный на клетки бумажный лист и набор карточек сходный с детской настольной игрой. Простейшие фигуры учащиеся могут выкладывать с помощью квадратных карточек, моделирующих соответствующие команды языка программирования. Перевод порядка построения плоской фигуры в линейную запись на языке программирования становится особенно наглядным, если разложить соответствующие карточки по прямой линии, объединив одинаковые последовательные карточки в стопки (рис. 2). Выложив фигуру и получив соответствующую

Система команд

1. Сдвиги на одну клетку / Н,П,В,Л/.
2. Б – уход с клетки без следа.
3. С – уход с клетки со следом.

Сдвиги – карточки, Б и С – фишки, след помечается безмянными фишками.

Порядок рисования одной фигуры		Команды сдвига: Н – вниз П – вправо В – вверх Л – влево  В интерактивном режиме им соответствуют нажатия на клавиши-стрелки
Фиксированная исходная клетка		

Пример программы

Программа рисует пять фигур, описанных в процедуре. Рисование очередной фигуры начинается с того места, где закончена предыдущая.

5 Ф

Ф: Б Н П 2В П С П В Л В – описание процедуры

– семейство из пяти фигур

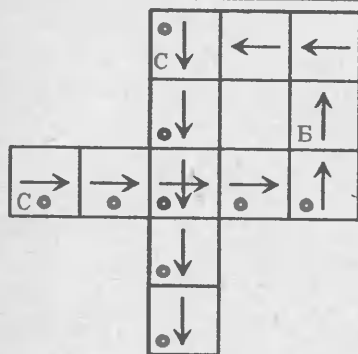
Рис. 1

щую команду для ее построения, ребенок может: с одной стороны предложить команду товарищу для воспроизведения этой фигуры, а с другой – построить ее на экране компьютера в интерактивном режиме. Наконец, эту команду можно исполнить и в командном режиме.

При освоении процедур следует пользоваться трафаретами фигур, либо заранее заготовленными, либо вырезаемыми учащимся самостоятельно. Для построения семейства геометрических фигур тра-

фареты распределяют по плоскости, а затем выявляют описание процедуры, налагая карточки-команды (рис. 3). При материальном моделировании семейства достаточно использовать два-три трафарета, на экране же можно построить многофигурное семейство, используя возможности языка программирования.

При построении структурно сложных сред обучения особое внимание следует обратить на создание материальных моделей, поскольку они являются наиболее



Порядок построения плоской фигуры

Карточки, выложенные в линию



Команда: С 4П В Б В 2Л С 5Н

Рис. 2

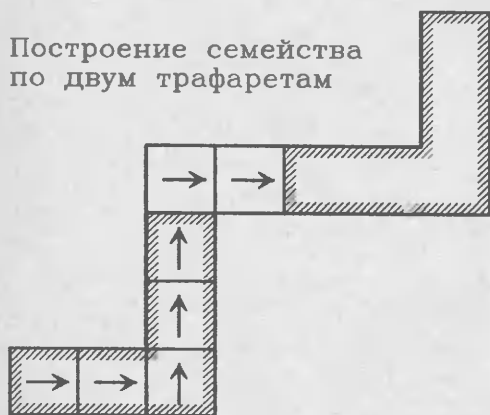
косным элементом среды обучения и в то же время играют очень важную роль в общей структуре. Облегчить выбор эффективных материальных моделей может использование идей, успешно апробированных при со-

здании конструкторов, игрушек, настольных игр, головоломок и т. д.

## ЛИТЕРАТУРА

1. *Пейперт С.* Переворот в сознании: дети, компьютеры и плодотворные идеи. М., 1989.
2. *Ильенков Э. В.* Философия и культура. М., 1991.
3. *Ракитов А. И.* Философия компьютерной революции. М., 1991.
4. *Тошио Савада.* Соробан – японские счеты. Курьер ЮНЕСКО, декабрь, 1986.
5. *Сильвестров В. В.* Понятийная основа соотношения "логоса" и "зйдоса" в европейской философской традиции. В кн. *Mathsis.* Из истории античной науки и философии. М., 1991.
6. *Талызина Н. Ф.* Формирование познавательной деятельности младших школьников. М., 1988.
7. *Салмина Н. Г.* Виды и функции материализации в обучении. М., 1981.

Построение семейства по двум трафаретам



Программа:  
10Ф

Ф: С 2П 3В Б 2П

Рис. 3

С. Е. Пронина

## ИНФОРМАТИКА В НАЧАЛЬНОЙ ШКОЛЕ

В настоящее время информатика, как самостоятельный предмет, включена в программу 10-11 классов. Однако, по мнению специалистов, изучение информатики должно начинаться раньше — уже в младших классах общеобразовательной школы.

Ниже приводится экспериментальный авторский курс основ информатики в начальной школе.

### Цели эксперимента:

- привить школьникам культуру общения с ЭВМ;
- воспитать у учащихся нравственно-ответственное отношение к компьютерам и информационным системам, с которыми им придется работать;
- научить учащихся пользоваться готовым программным обеспечением;
- сформировать у учащихся первоначальные понятия о компьютерах, алгоритмах, информатике.

Половина учебного времени отводится на теоретическую подготовку, другая — на практическую работу за компьютером. Класс разбивается на подгруппы по 10-15 чел. Длительность одного занятия определена в соответствии с санитарно-гигиеническими нормами: 20 мин для учащихся 1-3 классов (0,5 час в неделю).

### 1 класс

1. Правила поведения и техника безопасности при работе с компьютерами. Экскурсия.
2. Знакомство с клавишами управления курсором. (Игра *KLAD.SAV*.)
3. Работа с программой, контролирующей навыки устного счета в пределах 10. Знакомство с клавишами дополнительной цифровой клавиатуры и клавишей «пробел».
4. Знакомство с основной группой клавиш. Практическая работа с русифицированным клавиатурным тренажером.
5. Знакомство с текстовым редактором.
6. Диалог с компьютером. Компьютер сочиняет сказки. (Работа с программой *BVHAL.SAV* (Роботландия)). Знакомство с клавишей *AP2*.
7. Игра *SPION.SAV*. Развитие реакции в общении с компьютером.
8. Знакомство с исполнителем Квадратик (Роботландия).

Таким образом учащиеся 1-го класса знакомятся с компьютером, учатся общаться с ЭВМ в режиме диалога, работают с несложными программами по математике и русскому языку.

Основное внимание уделяется формированию культуры общения с компьютером.

### 2 класс

1. Знакомство с меню. Выбор режимов работы. Работа с контролирующей программой по математике, использующей меню для выбора режимов работы.
2. Знакомство с курсором и клавишами управления курсором (работа с программой *BVHAL.SAV* (Роботландия)).
3. Рисование, компьютерная графика. Знакомство с графическими редакторами.
4. Викторина "Угадай сказку по смыслу", работа с загадками.
5. Нахождение алгоритма задачи перевоза волка, козы и капусты через реку. (Работа с программой *Перевозчик* (Роботландия)).
6. Ханойские башни. Нахождение самых коротких алгоритмов для перекладывания 3 и 4 колец. Работа с программами *Ханойские башни* (Роботландия).
7. Решение алгоритмического этюда Коных (Роботландия).
8. Работа с диктантами и упражнения по русскому языку.
9. Игры, развивающие реакцию, логическое мышление и ориентацию (*R3.SAV*, *TETRIS.SAV*, *KLAD.SAV*).

Учащиеся 2-го класса знакомятся с понятиями: «меню», «курсор», программными средствами (графические редакторы), развивают логическое и алгоритмическое мышление. В процессе обучения происходит пропедевтика понятий «алгоритм» и «информация».

### 3 класс

1. Переливашка. Составление алгоритмов переливания жидкостей в сосудах (Роботландия).
2. Логическая игра *DOKER.SAV* (*SOMIR.SAV*, Мудрый крот (Роботландия)).
3. Робот — устройство, работающее по программе. Составление программ для роботов. Программирование в среде *SOMIR*.
4. Исполнитель — автомат. Составление алгоритмов получения с помощью автомата наперед заданного числа.

5. Упражнения по русскому языку на компьютере.
6. Работа с контролирующими программами по математике.
7. Робот ЗИП. Составление алгоритмов работы робота.
8. Робот-планетомобиль. Составление алгоритмов работы робота. Демонстрация работы робота.

Учащиеся 3-го класса усваивают понятие «алгоритма», развивают логическое мышление, знакомятся с роботами, работают с программами по русскому языку и математике. Большое внимание уделяется понятиям «робот», «программирование робота».

### Техническое и программное обеспечение курса

Курс ориентирован на машинный вариант обучения информатике.

В качестве технических средств обучения используются: микропроцессорный робот-планетомобиль, школьные микрокалькуляторы, КУВТ УКНЦ.

В качестве моделей исполнителей алгоритмов используются микропроцессорный робот-планетомобиль и различные исполнители, реализованные в программных средах Роботландия и SOMIR.

Компьютеры используются как технические средства обучения и как объект изучения. Микрокалькуляторы применяются для приобретения навыков работы с простейшими видами вычислительной техники.

В качестве программного обеспечения используются:

- пакет прикладных программ Роботландия;
- элементарная среда программирования SOMIR;
- графические редакторы RBPNT.SAV, AWGRED.COD, GRAFT.COD;
- обучающие и контролирующие программы по различным предметам, алгоритмические, логические, динамические учебные игры.

\*\*\*\*

- \* Вы производите и продаете компьютеры, периферию, программные продукты?
- \* Вы осуществляете гарантийное и постгарантийное обслуживание техники?
- \* Вы открыли курсы «компьютерной грамотности»?

Все эти сведения просто необходимы читателям журнала «Информатика и образование»!

К вашим услугам цветные слайды на обложке, информационные статьи и интервью, рекламные полосы.

## РЕКЛАМА В НАШЕМ ЖУРНАЛЕ ПОМОЖЕТ ВАШЕМУ БИЗНЕСУ!

По вопросам размещения рекламы  
обращаться к Васильевой Наталье Алексеевне.  
Телефон/факс: 208-67-37.

# НОВЫЕ ПРОЕКТЫ

И. В. Роберт

## ВИРТУАЛЬНАЯ РЕАЛЬНОСТЬ

Виртуальная реальность — это новая технология неконтактного информационного взаимодействия, реализующая с помощью комплексных мультимедиа—операционных сред иллюзию непосредственного вхождения и присутствия в реальном времени в стереоскопически представленном «экранном мире».

Технология неконтактного информационного взаимодействия, реализуемая системой «Виртуальная реальность», позволяет компьютеру отобразить непосредственно в цифровой форме импульсы от «информационной перчатки» («интерфейс-перчатка») и «информационного костюма». Рука пользователя, одетая в «информационную перчатку», может быть спроецирована в виртуальной форме в трехмерной компьютерно-генерированной среде. Манипулируя «информационной перчаткой», пользователь может взаимодействовать с виртуальным миром, передвигая и управляя объектами, либо использовать набор жестов в качестве команд. При наличии «информационного костюма», «информационной перчатки» и «информационных очков» со встроенными стереоскопическими экранами (очки-телемониторы) пользователь может, образно выражаясь, «шагнуть» прямо в виртуальный мир.

Уже в настоящее время возможности системы «Виртуальная реальность» используются при тренинге спортсменов, профессиональной подготовке специалистов в области астронавтики, архитектуры, медицинской диагностики, организации развлечений и досуга, а также в областях, использующих научную визуализацию. Например, если возможности трехмерной компьютерной графики позволяют осуществлять прогноз результатов хирургической операции, представляя трехмерное изображение на экране компьютера, то использование системы «Виртуальная реальность» позволяет создать иллюзию реально проводимой хирургической операции.

## Основные компоненты системы «Виртуальная реальность»

Базовыми компонентами типичной системы «Виртуальная реальность» (1), (2), (8), (9), (11), (12) являются:

- перечни или списки с перечислением и описанием объектов, формирующих виртуальный мир, в подсистеме создания и управления объектами виртуального мира;
- подсистема, распознающая и оценивающая состояние объектов перечней и непрерывно создающая картину «местонахождения» пользователя относительно объектов виртуального мира;
- головной установочный дисплей (очки-телемониторы), в котором непрерывно представляются изменяющиеся картины «событий» виртуального мира;
- устройство с ручным управлением, реализуемое в виде «информационной перчатки», или «спейс-болл», определяющее направление «перемещения» пользователя относительно объектов виртуального мира;
- устройство создания и передачи звука.

Контакт пользователя с системой «Виртуальная реальность» может осуществляться голосом или с помощью специального устройства — джойстинга, обеспечивающего эффект обратной силовой связи, а также с помощью очков-телемониторов.

Отдельно остановимся на последних устройствах, так как обеспечение связи посредством голоса ничего принципиально нового в себе не несет и используется сравнительно давно. Следует лишь отметить, что в системах «Виртуальная реальность» реакция на звуковой сигнал, полученный от пользователя, производится в соответствии с составленной программой.

Устройство ДЖОЙСТРИНГ создает эффект обратной силовой связи, ИНТЕРФЕЙС=ПЕРЧАТКА обеспечивает общение жестами, преобразуя каждое движение пальцев руки в электрические сигналы, воспринимаемые и расшифровываемые компьютером. Например, можно, манипулируя перед экраном пальцами в информационной перчатке, перемещать объекты на экране. Более того, можно «войти» в виртуальный мир экрана, отодвигая, перемещая, трогая предметы, отображенные на экране.

Так, например, можно ощутить сферичность шара, иллюзию хватания предмета, изображенного на экране. Эти эффекты достигаются следующим: между слоями ткани интерфейс-перчатки проложены тонкие световодные нити, обвивающие каждый палец, проходящие вдоль кисти руки и подсоединенные к специальной плате, смонтированной в перчатку в области запястья. С помощью специальных устройств происходит преобразование световых сигналов в электрические. Так, например, если согнуть палец, сгибается световод, энергия импульса изменяется, что немедленно улавливается фотозлементом, который вырабатывает электрический сигнал, воспринимаемый компьютером.

Кроме того, в интерфейс-перчатку смонтированы на уровне подушечек пальцев руки устройства тактильной обратной связи. В нужный момент, определяемый системой, эти устройства надавливают на кончики пальцев, создавая эффект касания предмета виртуального мира (например, поверхности шара, изображенного на экране). Иллюзию хватания предмета обеспечивает устройство, передающее усилия пользователя по «тросикам», приводящим в движение сервомоторы. Компьютер в соответствии с программой производит управление сервомоторами, осуществляющими обратную силовую связь с пользователем. Производимые пользователем с помощью джойстинга манипуляции улавливаются компьютером, который считывает информацию и подает сигналы углового положения, генерируя обратную связь при помощи сервомоторов. Это позволяет, например, «ощутить тяжесть» предмета, изображенного на экране — «взятого в руку», или «натолкнуться» на препятствие, изображенное на экране. В интерфейс-перчатку смонтированы также датчики изгиба пальца, отведения пальца в сторону, абсолютного положения и ориентации. Они обеспечивают синхронность функционирования неконтактного взаимодействия руки пользователя с системой.

ОЧКИ-ТЕЛЕМОНИТОРЫ представляют собой пару очков, внутри которых находятся два миниатюрных монитора, каждый строго напротив каждого глаза. Очки-телемониторы обеспечивают стереоскопическое видение экранного представления виртуального мира за счет того, что каждый глаз пользователя видит предмет на экране под разными углами. Современные системы «Виртуальная реальность» обычно обеспечивают создание различных картинок для каждого глаза с определенным смещением. При этом решается задача построения ранее описанной картинки дважды за короткое время с интервалом в несколько дюймов. Очки-телемониторы, снабженные специальными датчиками, информируют компьютер о поворотах головы человека, сидящего перед экраном, в результате чего пользователь имеет возможность «окинуть взглядом» всю стереоскопически представленную картинку виртуального мира.

Для более совершенного, многогранного и многофункционального неконтактного взаимодействия пользователя с виртуальным миром используется специальный ИНФОРМАЦИОННЫЙ КОСТЮМ.

Как видно из вышеизложенного, применение системы «Виртуальная реальность» в известном смысле решает проблему отдаления интерфейса между человеком и компьютером.

Следует добавить, что техническая реализация системы «Виртуальная реальность» имеет различные варианты. Мы схематически представили описание одной из возможных.

### Виды взаимодействия в системе «Виртуальная реальность»

В настоящее время можно выделить три подхода к осуществлению информационного взаимодействия пользователя с объектами виртуального мира, создаваемого системой «Виртуальная реальность» (1); (2); (9); (11); (12). Первый подход реализует идею погружения в виртуальный мир. При этом, образно выражаясь, пользователь, облачившись в скафандр, снабжающий информацией, «входит» в дискретную цифровую Вселенную. Манипулируя информационной перчаткой, он непосредственно взаимодействует с компьютером, перемещая, трогая объекты виртуального мира, представленные на экране, «движется» или «летает» внутри него, естественно, с синхронным звуковым сопровождением. Второй подход — это оконное представление трехмерного пространства виртуального мира на экране компьютера.



При этом в качестве средства управления используются устройства типа «спейс-болл», или «летающая мышка», обладающие значительной степенью свободы. Третий подход — это реализация взаимодействия с объектами виртуального мира «третьим лицом», представленным движущимся изображением на экране компьютера и отождествляемым с пользователем. При этом действиями «третьего лица» управляет пользователь, находя свое собственное изображение на экране.

Все три вышеперечисленных подхода реализуют основную идею информационного взаимодействия, обеспечиваемого системой «Виртуальная реальность». Эта идея заключается в обеспечении, во-первых, непосредственного участия пользователя в событиях, протекающих в виртуальном мире, но происходящих в реальном времени и, во-вторых, максимального отдаления интерфейсов между пользователем и компьютером.

### Психолого-педагогические возможности

Реализация вышеописанных возможностей позволяет создавать принципиально новый уровень информационно-предметной среды за счет «погружения» в трехмерную, стереоскопически представленную виртуальную реальность; обеспечивающую: моделирование ОЩУЩЕНИЙ непосредственного контакта пользователя с объектами виртуальной реальности (видеть, слышать, осязать рукой); бесконтактное УПРАВЛЕНИЕ объектами или процессами виртуальной реальности; ИМИТАЦИЮ РЕАЛЬНОСТИ — эффект непосредственного участия в процессах, происходящих на экране, влияния на их развитие и функционирование; ИНТЕРАКТИВНОЕ ВЗАИМОДЕЙСТВИЕ с объектами или процессами, находящими свое отображение на экране, реализация которых в реальности невозможна.

Возможности системы «Виртуальная реальность» позволяют через реализацию и внедрение специальных методик встраивания технологий обучения в предметноориентированные учебные среды осуществлять ПЕДАГОГИЧЕСКОЕ ВОЗДЕЙСТВИЕ лонгирующего характера, которое:

- развивает наглядно-образное, наглядно-действенное, интуитивное, творческое, теоретическое мышление;
- формирует эстетические вкусы.

Использование системы «Виртуальная реальность» открывает новые методы и возможности в процессе формирования:

- умений и навыков в сфере проектирования предметного мира;
- умений и навыков в области художественной деятельности;
- абстрактных образов и понятий при моделировании изучаемых объектов и явлений как окружающей действительности, так и тех, которые в реальности не воспроизводимы.

Реализация возможностей системы «Виртуальная реальность» ограничена лишь программными средствами, разработанными для функционирования «виртуальных миров», а также возможностями hardware, реализующих эти среды.

Используя возможности системы «Виртуальная реальность», можно визуализировать процесс трансформации стереоскопически представленных геометрических фигур; обеспечивать визуализацию взаимодействия между стереоскопически-представленными объектами, используя инструмент моделирования ситуаций; обеспечивать информационный обмен, обратную связь между пользователем и объектами виртуального мира. Кроме того, можно обеспечить:

- создание объектов «виртуальной реальности» (экранное изображение стереоскопически-представленных трехмерных геометрических фигур);
- оперирование ими;
- наблюдение различных форм стереоскопически-представленного динамического изображения стереометрического объекта;
- построение стереоскопически-представленных фигур;
- управление отображением на экране модели стереометрического объекта, ее преобразованиями.

СВОЙСТВА описываемой в качестве «виртуальной реальности» позволят обеспечить: динамическое представление на экране (перемещение, вращение) стереоскопически-представленного изображения трехмерных геометрических фигур; динамическую развертку и «свертывание» на экране стереоскопического изображения трехмерной фигуры; «вхождение внутрь» стерео-изображения фигуры с возможностью изучения и наблюдения с различного ракурса изображений ее элементов; «вхождение внутрь» с возможностью изменения «внутреннего интерьера» изображения, «перемещения» в нем.

Возможности системы «Виртуальная реальность» обладают мощным стимулирующим ПЕДАГОГИЧЕСКИМ ВОЗДЕЙСТВИЕМ лонгирующего ха-

рактера, которое формирует «видение глубины изображения»; развивает склонность к анализу, синтезу, абстрагированию, обобщению; инициирует развитие операционального, наглядно-образного, теоретического мышления обучаемого; открывает новые методические возможности в процессе формирования абстрактных образов и понятий.

Говоря о перспективах использования системы «Виртуальная реальность», можно прогностировать ее применение в образовании — при изучении стереометрии, черчения; при решении конструктивно-графических, художественных и других задач, для решения которых необходимо развитие умения создавать мысленную пространственную конструкцию некоторого объекта; в процессе профессиональной подготовки специалистов в условиях, максимально приближенных к реальной действительности; при организации досуга и развивающих игр.

### Литература:

- [1]. Материалы семинара Virtual Reality'91, The Conference Forum, London (Виртуальная реальность'91, Конференция и Форум. Первая ежегодная конференция и выставка) г. Лондон, Великобритания, 5 — 7 июня 1991 г.
- [2]. Материалы 3-й ежегодной международной конференции «Виртуальная реальность'93: воздействие и применения», Олимпия, конференц-центр, г. Лондон, Англия, 6 — 7 апреля 1993 г. — Лондон.: «Virtual Reality International 93: Impact & Applications», 1993.
- [3]. Материалы семинара MultiMedia'91 The first European MultiMedia event, Conference &

Exhibition, Olimpia 2 London (МультиМедиа — Первое Европейское событие, Конференция и Выставка, Олимпия 2 Лондон), 25 — 27 июня 1991 г.

[4]. Материалы семинара MultiMedia information, Churchill College Cambridge, UK (МультиМедиа информация, Черчилль Колледж, г. Кембридж), 15 — 17 июля 1991 г.

[5]. Материалы семинара The Edinburgh Conference interactive learning — the MultiMedia wave, Heriot—Watt University (ИЛ'91 Эдинбургская конференция Интерактивное обучение — волна МультиМедиа, Херион—Уат Университет, г. Эдинбург), 13 — 15 сентября 1991 г.

[6]. Международная конференция «Применение новых компьютерных технологий в образовании» (тезисы докладов). г. Троицк. 12—13 августа 1991 г. М.: Наука, 1991.

[7]. Обучающая программно-методическая система «Многогранники» (Методические рекомендации, инструкция пользователя, 2 диска (3.5"), MSX). Каз. ПК ПС, 1990.

[8]. Bennett Davis «Grand illusions». Discover. The world of science rebuilding the brain. June 1990, 41. USA.

[9]. Proceedings of the 2ND annual conference on Virtual Reality Internacional: Impacts & Applications. Held in London, in april 1992. — London.: British Library Cataloguing in Publication Data, 1992.

[10]. Robert I. Psychological and pedagogical issues in using Virtual Reality systems. «Virtual Reality Internacional'92». Proceedings of the 2ND annual conference on Virtual Reality Internacional: Impacts & Applications. Held in London, in April 1992. — London.: British Library Cataloguing in Publication Data, 1992.

[11]. The Virtual Reality Newsletter. Volume 1, issue 1. February 1992. — London.: Cydata Limited, 1992.

[12]. The Virtual Reality Newsletter. Volume 1, issue 2. March 1992. — London.: Cydata Limited, 1992.

**Фонд Новых Технологий в Образовании «Байтик», г.Троицк,  
Департамент по образованию Московской области,  
Computer Using Educators, Inc, USA  
при поддержке RUI a.g. Apple Computer IMC**

в период с 30 июня по 3 июля 1994 г. проводят традиционную международную конференцию «Использование новых технологий в образовании». Конференция посвящена вопросам практического использования новых технологий в школьном образовании.

**Телефон:** (095) 334-03-67  
**Факс:** (095) 334-57-76  
**Email:** trolzk@anet.sovusa.com

# ПЕДАГОГИЧЕСКИЙ ОПЫТ

В.В. Булитко, А.Л. Ермилов

## НАБЛЮДЕНИЕ ЧИСЛОВЫХ ГАЛАКТИК

Все началось с того, как однажды автор вывел на экран своего компьютера пару тысяч простых чисел по такой системе: числу  $p$  ставится в соответствие точка в полярной системе координат с радиусом  $\rho$  и углом  $\varphi$  по MOD  $2\pi$ .

При этом на экране появилось изображение, подобное этому (рис. 1):

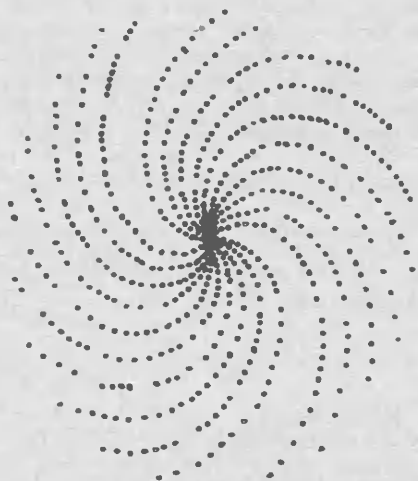


Рис. 1

Неожиданность этой «спиралевидной галактики», странные пропуски в ветвях заинтересовали автора и послужили темой небольшого исследования.

Вскоре выяснилось следующее. Если отобразить по вышеизложенной системе все числа из  $\mathbb{R}^+$ , то получим просто раскручивающуюся спираль (рис. 2).

«Удаляя» из этой спирали образы не натуральных чисел, мы получаем много пропусков (рис. 3).

(Все три изображения выполнены в разных масштабах.)

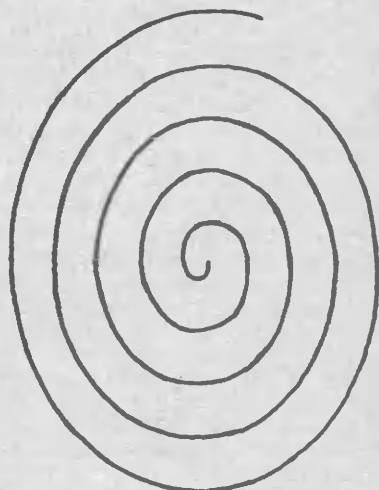


Рис. 2

Вот тут и начинается самое интересное: образы оставшихся (т.е. натуральных) чисел группируются в четко выраженные «ветви». Например, для чисел из диапазона  $[300, 5000]$  таких ветвей — 44, что можно заметить на последнем изображении. Удалось установить, что эти ветви — образы множеств:

$$k_i = \{44n + i \mid n \in \mathbb{Z}^+\}, \\ 0 \leq i \leq 43$$

Понятно, что если отображать лишь простые числа, то сразу исчезает половина ветвей — образов множеств  $k_0, k_2, \dots, k_{42}$  (исключение составляет число  $2 \in k_1$ , но оно настолько мало, что «теряется» где-то в центре изображения). Остается 22 ветви, но ведь на первом изображении их 20! Две исчезнувшие ветви — образы классов  $k_{11}$  и  $k_{33}$  (легко заме-



Рис. 3

тить, что всякое простое число, кроме 11, не входит ни в один из них).

После такого разбора может показаться, что галактика из простых чисел и сама довольно проста, однако это не так.

Трудности начинаются, если задаться вопросом о «магии» числа 44: почему именно столько ветвей в галактике – образе отрезка  $N[300, 5000]$ ?

Рассмотрим образы чисел  $k$  и  $k + d$  на плоскости (рис. 4):

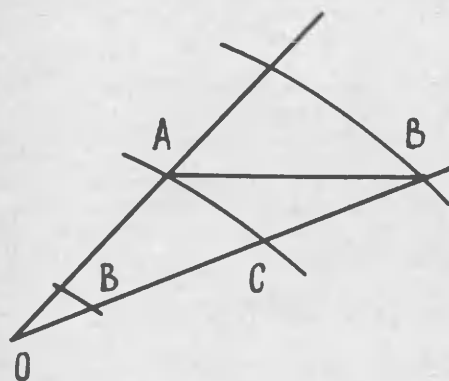


Рис. 4

Здесь  $A$  – образ  $k$ ,  $B$  – образ  $k + d$ ;  $O$  – начало координат.

Как видно из рисунка,  $OC = OA = k$ ,  $OB = k + d \rightarrow CB = d$ . Возникает вопрос: когда  $A$  и  $B$  попадут в одну ветвь? Дело заключается в том, что на плоскости (эк-

ране) вообще нет никаких ветвей – лишь множество точек. Ветвь выделяет зрительная система человека и можно лишь пытаться понять, на основании каких признаков это делается. Эта задача имеет серьезный смысл: ведь как известно (см., например, [1]), в последнее время во многих математических исследованиях большую роль играет интерактивная компьютерная графика (ИКГ). Коротко суть ИКГ сводится к генерации математических гипотез за счет просматривания человеком специального вида изображений, генерируемых ИКГ-системой. Для ИКГ-системы очень важен характер этих картинок с точки зрения зрительного восприятия (т.е., в нашем приближении, глаза человека) так как он (характер) влияет на возникновение и продуктивность ИКГ-гипотез.

Авторами была выдвинута рабочая гипотеза, что глаз выделяет ветви, базируясь на минимизации расстояния между образами точек на плоскости. Отвлечемся от простых чисел и будем рассматривать галактику, состоящую из всех чисел  $N$ . Кроме того, условимся через  $\alpha(k, d)$  обозначать расстояние на плоскости между образами точек  $k$  и  $k+d$ . Допустим, что для всех натуральных чисел из  $[a, b]$  лучшей прибавкой является число  $h$ , т.е.

$$\forall k \in [a, b] \quad \min_{n \in N} \alpha(k, n) = \alpha(k, h).$$

Тогда все натуральные числа из  $[a, b]$  распадаются на  $h$  классов

$$K^h = \{h \cdot n + i \mid n \in N\} \quad (i \in \overline{0, \dots, h-1}).$$

Эти классы обладают тем свойством, что для всякого числа из такого класса ближайшие к нему (по расстоянию на плоскости) числа – это соседние представители этого класса. Формально: для образа числа  $h \cdot n + i$  ближайшими на плоскости являются образы чисел  $h(n-1) + i$  и  $h(n+1) + i$ .

Мысль авторов заключалась в том, что глаз и выделяет эти  $K^h$ . Изучение лучших прибавок дало следующие результаты:

1) Лучшие прибавки являются числителями подходящих дробей к  $2\pi$  (6, 19, 25, 44, 333, 710, 103993, 312689, 1980127, ...).

2) Для всяких двух чисел  $h_1, h_2$  ( $h_1 < h_2$ ) можно эффективно найти сегмент  $[a, b]$  такой, что:

$$\alpha(x, h_1) < \alpha(x, h_2) \Leftrightarrow x \in [a, b]$$

Это означает, что не существует общей лучшей прибавки — всегда найдется число большее данного и лучше его (в смысле расстояния на плоскости) для больших чисел.

3) Для всякого числа  $n$  можно эффективно найти лучшую прибавку: достаточно перебрать все прибавки из отрезка  $[1, \text{int}(\alpha(n,1)) + 1]$ . (Или, что быстрее, перебрать числители подходящих к  $2\pi$  дробей, попадающие в этот диапазон.)

4) Для чисел от 1 до 1818868611 авторами найдены лучшие прибавки (машинным путем):

Отрезок	Лучшая прибавка для всех чисел из него
[1, 6]	1
[7, 75]	6
[76, 230]	19
[231, 263]	25
[264, 21531]	44
[21532, 70921]	333
[70922, 1818868611]	710

Все было бы замечательно, если бы количество ветвей, выделяемых глазом на разных участках галактики, совпадало с

количеством классов  $K_n^h$ , т.е. совпадало с числом  $h$ . Однако, оказалось, что на некоторых отрезках  $N$  это действительно так, а на других глаз выделяет сразу множество ветвей, а не только классы  $K_n^h$ .

Авторам хотелось бы упомянуть, что если «свернуть» всю окружность до сектора, то проявится другое семейство ветвей (рис. 5):

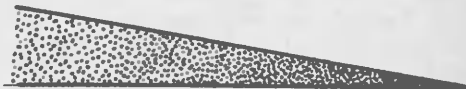


Рис. 5

Возможно, что это свидетельствует об «уравновешивании» углового и радиального масштабов в восприятии.

Таким образом, мы видим, как решение математической задачи позволяет узнать больше о зрительном восприятии человека и будет полезно при разработке ИКГ-систем.

Литература

1. Зенкин А.А. Когнитивная компьютерная графика. — М.: «Наука», 1991.

ПРИЛОЖЕНИЕ

Пример простейшего варианта программы для вывода числовой галактики. Функция prime дает true тогда и только тогда, когда ее аргумент — простое число.

```
(Turbo-Pascal 5.0, CGA)
program galax;
uses graph;

const p1=1
      p2=3000
      mash=100.0/p2
      { прогоняются числа из [ p1, p2 ] }

var counter, gm, gd: integer;
function prime (n: integer): boolean;
begin
.....
end;

begin
gd:=CGA; gm:=CGAC1; initgraph(gd,gm,'');
for counter:=p1 to p2 do if prime(counter) then
  putpixel(159+round(mash*counter*cos(counter)),
          99+round(mash*counter*sin(counter)),3);

write(#7); readln; closegraph
end.
```

А.И. Сафьянников

## «PRIVATE TUTORS»

Ни для кого не секрет, что в последнее время интерес к изучению иностранных языков все возрастает. Знание языка открывает перспективы для профессиональной карьеры во многих областях, требуется для общения на работе и на отдыхе, в России и за границей. Появилась масса методик быстрого освоения языка: «English», «English for businessmen», «Talking American» и др. Как правило, эти курсы ориентированы на взрослых. Хотя «мы все учились понемногу...», в том числе и языку, школьного и вузовского уровня явно недостаточно для нормального существования в мире с открытыми границами. Не так давно к традиционным школьным методам изучения иностранного языка добавились лингафонные кабинеты и компьютерные классы, призванные повышать эффективность обучения.

Однако известно, что нет ничего лучше для изучения языка, чем пребывание в языковой среде. Наши богатые дореволюционные соотечественники посылали с этой целью своих отпрысков в Европу, брали французских гувернеров и гувернанток, да и сами свободно разговаривали с детьми на французском или немецком. То есть занимались тем, что мы называем сейчас индивидуализацией обучения. Вновь появились в нашем обществе люди, готовые и способные отправить своих горячо любимых чад в Оксфорд или Сорбонну. Но «узок круг этих «революционеров», страшно далеки они от народа».

Однако есть возможность и в обычной школе, в детском саду или в собственном доме создать языковую среду, игровую среду, куда ребенок погружается сам и с удовольствием. Эта среда — разговаривающие интерактивные игрушки фирмы «Texas Instruments», охватывающие возраст от 1 года до 14-15 лет. Сейчас каталог развивающих игрушек АО «ЭЛТИ-КУДИЦ», являющегося официальным дилером «Texas Instruments» в странах

СНГ, насчитывает 27 наименований и постоянно пополняется новыми изделиями. Большая часть этих игрушек предназначена для изучения иностранного языка. Познакомимся с некоторыми из них.

### *«Клоун Кларенс-Говорящее кольцо»*

Игрушка представляет собой голову клоуна, составленную из пяти насаженных на штырь пластмассовых колец, на которых наклеены части смеющегося лица клоуна. На затылке изображен грустный клоун. Снимая кольца, ребенок слышит сообщения типа: «Ты снял шляпу», «А это у меня брови», «А это глаза» и т.д. Но самое интересное начинается, когда клоуна собирают. Клоун комментирует все действия ребенка, естественно, по-английски: «Куда ты ставишь нос, здесь должен быть рот», «Поверни кольцо, улыбающемуся рту должны соответствовать веселые глаза» и т.д. Собирая лицо клоуна как пирамидку, можно получить около 1000 различных лиц.

### *«Тедди-Дотронься и скажи»*

Медвежонок Тедди держит в своих лапах небольшую квадратную доску, куда можно вставлять карточки с картинками на разные темы: цвета, размеры, животные, формы, противоположности, части лица, части тела, одежда и т.д. Нажимая на какое-то изображение на карточке, ребенок слышит его название. После нескольких повторений темы медвежонок задает вопросы. Например, «Где овечка?», «Кто здесь самый маленький?», «Где желтый цвет?» и т.д. Если ребенок не расслышал вопрос, он всегда может попросить медвежонок повторить его нажатием на клавишу «Repeat». После правильного ответа ребенок поощряется веселой мелодией и словесным выражением восторга со стороны Тедди. Вместе с модулями расширения изделие содержит

42 различных игровых сюжета. Можно представить, сколько радости и пользы принесет эта игрушка детям.

### «Парта с книжкой-собеседницей»

Эта игрушка устроена по тому же принципу, что и медвежонок Тедди, только вместо карточек в парту вставляются книжки. Переходя вместе с героями сказок от страницы к странице, участвуя вместе с ними в приключениях, ученик отвечает на гораздо более сложные вопросы, чем задает медвежонок Тедди. Например: «Покажи, где находится обезьянка?», «Кто держит в руках сопочек?», «Куда прыгает собака?». Из этого понятно, что уловив какое-то одно слово, на вопрос не ответить. В книжках содержатся задания по изучению алфавита, чисел, построению предложения и др. Эта игрушка подводит ребенка к чтению через три этапа: предварительное

чтение, распознавание букв и функциональное чтение. В комплекте содержится 7 книг с 75 игровыми сюжетами и 1500 вопросами. Модуль расширения включает в себя дополнительную микросхему и еще 5 книг.

На эти игрушки дается 1-2 года гарантии, работают они от пальчиковых батареек, помещены в удароустойчивые корпуса, имеют прекрасный дизайн.

Решение коллегии Министерства образования Российской Федерации № 20/1 от 10.11.93 «О Программе информатизации образования в РФ на 1994-1995 гг.» предусматривает выделение средств учебно-методическому центру «ЭЛТИ-КУДИЦ» на разработку методических рекомендаций по использованию средств малой электроники в системе дополнительного образования. И тогда развивающие говорящие игрушки «Texas Instruments» придут в образование не как красивая экзотика, а как одно из средств обучения.

## «ЭЛТИ-КУДИЦ»

*предлагает новейшие образовательные технологии:*

1. Учебные компьютерные классы на базе АТ 286/386 для школ и детских садов. В комплект входит программное и методическое обеспечение, обучение специалистов, гарантийное обслуживание. Стоимость от 4000 USD.
2. Региональные телекоммуникационные образовательные сети. Стоимость абонентского пункта от 150 USD.
3. Оборудование фирмы «Texas Instruments» (США):
  - калькуляторные классы с учебными пособиями. Стоимость от 350 USD;
  - графические калькуляторы нового поколения с учебными пособиями, позволяющие существенно повысить эффективность занятий по математике и физике. Стоимость от 130 USD;
  - уникальные развивающие игрушки и языковые тренажеры (английский, немецкий, французский, итальянский). Стоимость от 30 USD.
4. Материалы Монтессори фирмы «Nienhuis» (Голландия) в комплексе с сертифицированным обучением и методической поддержкой. Стоимость от 4000 USD.

Возможна оплата в рублях по курсу ММВБ

**Тел.: (095) 392-78-18, 392-62-95**

**Факс: (095) 392-81-27**

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

**ЭЛТИ-КУДИЦ**

# АЭРОН

Статическое электричество - враг  
Ваш и Вашего компьютера!

Накапливаясь на одежде и поверхности тела человека, статическое электричество негативно влияет на здоровье: возникают неприятные ощущения, раздражительность, снижаются защитные функции кожного покрова.

При взаимодействии человека, несущего заряд, и компьютера, происходят микрозаряды высокого напряжения и повреждение узлов ПК.

Предлагаем прибор для снятия статического электричества, очистки воздуха от микробов и неприятных запахов.

Купите "АЭРОН"

Ваши компьютеры будут работать дольше  
Вы будете меньше болеть, утомляться  
и чувствовать себя более комфортно

# АЭРОН

630093, г.Новосибирск,  
Красный проспект, 11.

АО "АЭРОН" (для писем)

Тел.: (383-2) 22-44-23,  
23-15-01



# ЗАДАЧИ

Н.М.Бадин, С.Г.Волченков

Ярославский государственный университет

## ИЗУЧЕНИЕ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ

Парк существующих вычислительных машин очень разнообразен: от микрокалькуляторов до супер-ЭВМ. Соответственно различаются и методы программирования для них. Курс «Основы информатики и вычислительной техники», преподаваемый в 10-11 классах, основывается на так называемом «фон Неймановском» представлении о ЭВМ. Структура «фон Неймана» предполагает наличие одного процессора, который в каждый момент времени последовательно исполняет команды одной программы. В стороне остается весьма широкий и завоевывающий все больше прикладных областей класс параллельных вычислительных систем (ПВС) [1,2]. Каждая такая система может содержать от нескольких единиц до сотен и даже тысяч процессоров. При этом достигается гигантская общая производительность: до триллиона операций с плавающей запятой (Гигафлопс) в секунду. Но для ПВС необходим принципиально другой принцип программирования, - нужно уметь составлять алгоритмы решения одной задачи одновременно на нескольких процессорах. Человек мыслит аналогично однопроцессорной машине, а в данном случае, и в этом принципиальная новизна, ему одновременно требуется работать над несколькими проблемами, переключаясь с одной на другую (т.е. в режиме «разделения по времени мозговых атак»).

Чтобы дать школьникам хотя бы некоторое представление об этой области информатики, приведем в качестве примера решение одной из задач, требующих разработки параллельных алгоритмов. Эта задача была предложена на ярославской областной олимпиаде по программированию в феврале 1993 года.

Задача. *Имеется шестнадцать чисел  $a_1, a_2, \dots, a_{16}$  и  $n$  синхронно работающих процессоров. Каждый процессор за один такт*

*может сделать одно сложение. Требуется найти все частичные суммы:*

$$S_1 = a_1; S_2 = a_1 + a_2; S_3 = a_1 + a_2 + a_3; \dots$$

$$S_{16} = a_1 + a_2 + \dots + a_{16}.$$

Считается, что результат, полученный любым процессором, может использоваться всеми процессорами на всех последующих тактах.

Составить алгоритм, вычисляющий данные суммы за минимальное количество тактов, если

- а)  $n=2$ ;
- б)  $n=4$ ;
- в)  $n=8$ ;
- г)  $n=12$ .

Очевидно, что одному процессору ( $n=1$ ) потребуется 15 тактов для вычисления всех необходимых сумм. Поскольку каждая последующая сумма может быть вычислена только после предыдущей ( $S_k = S_{k-1} + a_k, k = 2 \dots 16$ ), то при  $n > 1$  потребуется не  $15/n$  тактов, а больше. Так при  $n = 2$  уже на первом такте один из процессоров должен или простаивать, или, что лучше, вычислять какую-нибудь вспомогательную сумму.

Не приступая к нахождению решения для конкретного  $n$ , можно определить, каково теоретически минимальное число тактов, требующееся для решения поставленной задачи. Дольше всего придется вычислять сумму  $S_{16}$ , поэтому и сосредоточим на ней свое внимание.

Назовем рангом суммы количество исходных чисел, ее составляющих. Так, например, ранг  $S_k$  равен  $k$ , ранг суммы  $a_4 + a_5 + a_6$  равен трем. На первом такте, сколько бы ни было процессоров, нельзя получить сумму ранга больше двух. Наибольший ранг, полученный на втором такте, равен 4 т.е. (2+2), на третьем - 8, а на четвертом - 16. Таким образом, независимо от

числа процессоров сумма  $S_{16}$  может быть вычислена не меньше, чем за четыре такта.

Итак, мы нашли нижнюю временную границу для разрабатываемого алгоритма. Возникает вопрос: «Достижима ли данная граница? И если да, то сколько потребуется процессоров?». Оказывается, что для  $n=8$  можно построить алгоритм, находящий все суммы ровно за четыре такта! Схематично алгоритм изображен на рис.1. На горизонтали отмечены

На первом такте только один процессор вычисляет искомую сумму  $S_2$ , а остальные выполняют подготовительную работу, вычисляя вспомогательные суммы, используемые на последующих тактах. На втором такте к первому процессору присоединяется второй, на третьем - третий и четвертый, а на четвертом - все остальные. Ни один из процессоров не простаивает ни одного такта времени и не выполняет бесполезной работы.

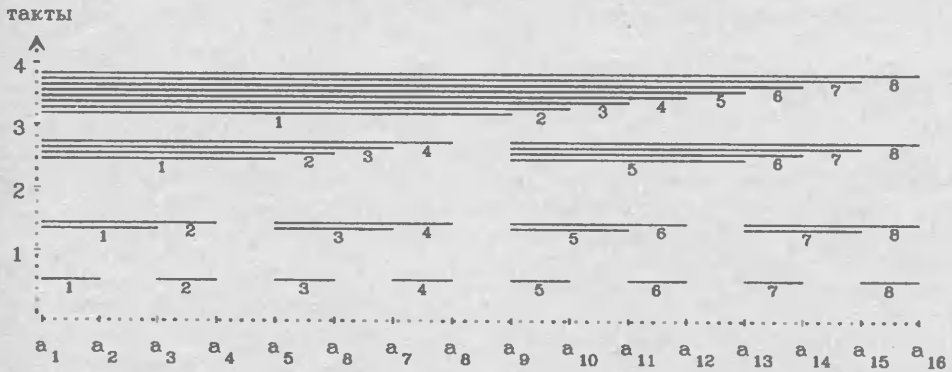


Рис. 1

шестнадцать точек, соответствующих исходным числам, а по вертикали откладывается последовательность тактов. Каждый отрезок соответствует сумме чисел, лежащих под ним, включая концы отрезка. Проекция его на ось ординат - номер такта, на котором получена данная сумма. Рядом с отрезком стоит номер вычислившего эту сумму процессора. Отрезки, включающие  $a_1$ , соответствуют искомым суммам  $S_k, k = 2...16$ .

Алгоритм, работающий для этой задачи минимально возможное время, можно построить для восьми процессоров, и увеличение количества процессоров выигрыша уже не даст. То есть, для  $n=12$  при решении задачи некоторые процессоры не будут полностью загружены.

Для  $n=4$  (рис.2) на первых трех тактах второй, третий и четвертый процессоры вычисляют промежуточные суммы, а на следую-

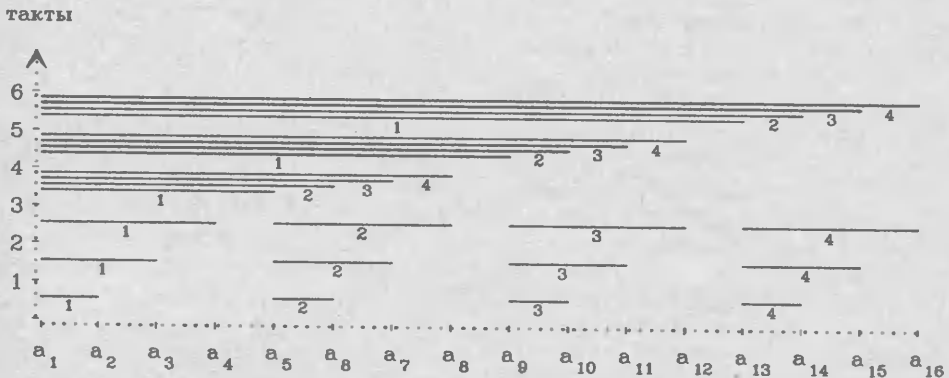


Рис. 2

щих трех тактах - уже итоговые. В целом получаем 6 тактов.

Наконец, для двух процессоров можно построить алгоритм, работающий 10 тактов. Причем подход, аналогичный предыдущим случаям, такого результата не дает. Другими словами, второй процессор первые семь тактов вычисляет все необходимые промежуточные суммы ( $a_9 + a_{10}$ ,  $a_9 + a_{10} + a_{11}$ , ...), а в остальное время оба процессора вместе вычисляют итоговые суммы. В этом случае мы получаем решение в 11 тактов. Если же второй процессор будет на одном такте готовить промежуточную сумму, а на другом вычислять итоговую, то общее время удастся сократить до 10 тактов (таб.1).

Таблица 1

N такта	1-й процессор	2-й процессор
1	$S_2 = a_1 + a_2$	$s = a_3 + a_4$
2	$S_3 = S_2 + a_3$	$S_4 = S_2 + s$
3	$S_5 = S_4 + a_5$	$s = a_6 + a_7$
4	$S_6 = S_5 + a_6$	$S_7 = S_5 + s$
...	...	...
10	$S_{15} = S_{14} + a_{15}$	$S_{16} = S_{14} + s$

Итак простая на вид задача предоставляет широкое поле для размышлений. В отличие от выведенной выше общей временной оценки алгоритма можно попытаться получить такую оценку снизу в зависимости от числа используемых процессоров. Попробуйте рассчитать число непроизводительных тактов работы (т.е. тактов, на которых вычисляются вспомогательные суммы), приходящиеся в среднем на один процессор. Так для  $n=2$  оно равно  $(2 \cdot 10 - 15) / 2$ , т.е. 2.5 такта / процессор. Оказывается, это число чуть больше двух для всех оптимальных алгоритмов! Это можно использовать для оценки минимального количества тактов при других  $n$ .

Данную задачу можно использовать как в теоретическом плане, так и в практическом.

На упоминавшейся выше олимпиаде школьников был предложен простой интерпретатор, написанный на Бейсике, позволяющий для каждого процессора задавать расписанные по тактам программы, состоящие из команд суммирования. При этом моделировалась общая для всех процессоров память из 100 регистров. Имелся набор сервисных команд: запуск на выполнение, просмотр содержимого памяти и т.п.. Тем самым участники олимпиады ставились в равные условия, работая в простой среде интерпретатора и сосредотачиваясь непосредственно на разработке параллельного алгоритма. Но школьникам можно предложить самим написать такой интерпретатор или программу моделирования ПВС. На этом пути возникают новые возможности изучения параллельных систем.

Аналогичная задача рассматривалась на олимпиаде по программированию в 1990 году. Предлагалось написать программу вычисления одного значения многочлена шестой степени для четырехпроцессорной системы. Каждый процессор имел свою локальную память из четырех ячеек, но мог обмениваться данными с другими процессорами. [3]. Составление алгоритма для ПВС во многом схоже с планированием работ, поэтому такого рода задания тоже могут быть сформулированы на языке параллельных процессов.

## Литература.

1. Вальковский В.А., Малышкин В.Э.. Элементы современного программирования и супер-ЭВМ. — Новосибирск: Наука, 1990.-140с..
2. Программирование на параллельных вычислительных системах; Пер. с англ. / Р.Бэбб, Дж.Мак-Гроу, Т.Аксельрод и др.; под ред. Р.Бэбба II. - М.Мир, 1991.-376с..
3. Волченков С.Г., Дашниц Н.Л., Потехин Н.В. Ярославские олимпиады по программированию: Методические указания. / Ярослав. обл. ин-т усоверш. учителей. - Ярославль, 1990.-31с.

## **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 — 15 000 РУБЛЕЙ !!!**

Научно-техническое предприятие «НОВАЯ АЛЬТЕРНАТИВА» предлагает Вам приобрести 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.

**МЫ ОБЪЯВЛЯЕМ** о своем выходе из предприятия АЛЬТЕРНАТИВА при НИИ АПН СССР.

**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: 656057, г.Барнаул, а/я 2513. НОВАЯ АЛЬТЕРНАТИВА.**

**Директор Гриценко А.Н.**

# ТОЧКА ЗРЕНИЯ

Хотя предмет «Основы информатики и вычислительной техники» изучается в школе уже восемь лет, это, по-прежнему, самый динамичный во всех отношениях курс. Нам хотелось бы знать ваше мнение об учебниках по информатике, о программном обеспечении, о программно-методических комплексах, с которыми вы работаете. Публикации на эти темы помогут разработчикам скорректировать свои идеи, а вам — обменяться через журнал мнениями друг с другом. Мы ждем ваших писем в рубрику «Точка зрения».

А.Г. Щеголев

## ДЛЯ ЧЕГО НАМ УЧЕБНИК?

*(Размышления над учебником по информатике. Но и не только!)*

О том, что преподавание учебного предмета (в особенности такого как информатика) нуждается в достойных учебниках, сомнений нет ни у кого. Однако, если поинтересоваться мнением учителей и учеников, зачем вообще нужен учебник, ответы обычно даются самые разные.

Одни считают, что учебник — это связанное изложение изучаемого материала. Другие называют его книгой для самостоятельного чтения. Третьи считают справочным материалом.

Развитие каждой учебной дисциплины настоятельно требует разобраться в том, какую же именно роль играет учебник при освоении предмета. И, как это часто бывает, при этом приходится решать вопросы, выходящие за рамки самого предмета.

Итак, постараемся выяснить, какова роль такой столь своеобразной книги, как школьный учебник, в учебной деятельности. Для кого предназначена эта книга, как ею пользоваться, нужна ли она только для подготовки к занятиям или, не в меньшей степени, для работы в классе?

Попытаемся представить себе разговор, в котором участвуют Автор (А), занятый поиском новых подходов к созданию учебников, Педагог (П) — учитель информатики и Ученик (У) средней школы.

**А.** Итак, какой же учебник всем нам необходим?

**П.** Что же тут непонятного? В учебнике должен быть кратко изложен весь материал, который нужен ученикам.

**А.** (Ученику). Ну, а ваша точка зрения?

**У.** Для меня главное — чтобы было где прочесть, если пропустил занятие. Ну, и на перемене подготовиться перед уроком.

**А.** А как лучше разбивать материал на параграфы?

**П.** Хороший учебник должен быть написан так, чтобы каждый параграф в точности соответствовал очередному уроку.

**У.** Вот-вот. Тогда и проблем не будет никаких.

**П.** По крайней мере, и ученику, и учителю удобнее структурировать материал именно таким образом.

**А.** К сожалению, не могу с Вами согласиться. Разве в Вашей практике не было случаев, когда в одном классе дело продвигается быстро, а в другом гораздо медленнее? Кроме того, часть уроков в одном из классов может пропасть из-за праздников, и т.п. Добавьте сюда же ненадежность компьютеров, когда заранее неизвестно, сколько из них «соизволит» заработать сегодня. Значит, учитель (и только учитель) вправе перепланировать занятия в каждом классе (вплоть до перестановки отдельных тем и изменения количества времени, отводимого на каждую тему). Разве можно при всем этом всерьез говорить о каком-то едином поурочном планировании?!

**П.** Да, пожалуй, вы правы. Но как же быть с учебником?

**А.** С вашего позволения, еще один вопрос. Как вы полагаете, может ли учитель требовать

от ученика больших познаний, чем отражено в учебнике?

П. Да, конечно. По отдельным темам я провожу занятия в форме докладов и дискуссий. При подготовке ученики, естественно, привлекают дополнительный материал из журналов, проспектов и т. д. Правда, я слышал, что у некоторых моих коллег возникали конфликты с учениками из-за того, что учитель спрашивает с них «лишнее».

У. (перебивая). А в самом деле, почему я должен еще где-то искать материал для уроков? Ведь в учебнике и так должно быть все необходимое. Значит, учитель не имеет права требовать знания чего-то дополнительного даже на «пятерку»!

А. (Педагогу). Скажите, а вы никогда не задумывались над тем, как выглядит в глазах учеников привлечение дополнительных (не из учебника) сведений? Согласитесь, в позиции школьников есть определенная логика: если в учебнике действительно есть «все», зачем же учитель стремится донести еще что-то? Усилия учителя вывести занятия «за рамки учебника» воспринимаются подчас совсем по-гоголевски (о, бессмертный «Ревизор»!): «Не приведи бог служить по ученой части! Всего боишься: всякий мешается, всякому хочется показать, что он тоже умный человек.»

П. Так что же, вы призываете вовсе не выходить «за рамки» учебника?!

А. Отнюдь. Как можно назвать учителя, работающего «строго по учебнику»? Видимо, «озвучивателем» этой книги (и, увы, цербером, контролирующим, как усвоен материал книги). Роль явно незавидная.

П. У меня еще свежи воспоминания о работе по экспериментальной программе, когда учебники еще не были написаны. И вы знаете, в этом были свои плюсы! Во-первых, ученики научились конспектировать происходящее на занятиях. В жизни этот навык, безусловно, пригодится. Во-вторых, мы выборочно использовали разные учебники. Но использовали фрагментарно, а главное, критически. При этом работа с учебником могла проходить и дома, и прямо на уроках. Задание ученикам, естественно, не сводилось к «прочсть и пересказать». Ведь у каждого из этих учебников своя логика изложения, которой мы не следовали. Поэтому задание заключалось в том, чтобы рассмотреть конкретный фрагмент учебника (параграф) под определенным углом зрения. При этом школьники имели право в чем-то и не согласиться с автором учебника! Конечно, такой ответ не должен сводиться к критиканству:

каждая точка зрения обязательно должна быть аргументирована.

А. Тогда, может быть, лучше обойтись вообще без учебника?

У. Нет, учебник все-таки нужен. Я совершенно не представляю себе, как наверстать упущенное, если, например, заболеешь.

П. Да и учителю, сказать по совести, спокойнее, когда учебник под рукой. Если бы можно было совместить преимущества работы без учебника, о которых я говорил, и в то же время учебник бы все-таки был.

А. Итак, налицо противоречие, снять которое в принципе не так уж сложно. Ведь в работе без стабильного учебника самым ценным было как раз столкновение логики учебника и логики самой учебной деятельности, руководимой учителем. Именно противоречие, конфликт между различными аспектами, различными последовательностями подачи материала создают проблемные ситуации, заставляют учеников думать, активизируют творческое мышление школьников. Очевидная для учеников неединственность в подходах — вот что главное!

П. Но как же этого добиться? Ведь обычно в учебниках...

А. Речь действительно идет о создании учебников нового поколения. Главное, что должно быть обеспечено — несовпадение «сюжетных линий» работы под руководством учителя (в классе и дома) и изложения материала в учебнике. Так бывает, когда вы берете с полки книгу и находите в ней ответы на все интересующие вас вопросы. Однако то, что написано в книге, может быть упорядочено совсем по другому, непривычному для вас принципу.

У. Так что же, учебник должен быть как справочник, что ли?

А. На этот вопрос придется ответить так: и да, и нет. Да — в том смысле, что в учебнике в четкой, систематизированной и наглядной (временами даже подчеркнуто «суховатой») форме должны быть представлены все основные результаты учебной деятельности. Нет — потому что учебник — это развернутое, достаточно подробное изложение со своим сюжетом, выразительными средствами, стилем и т. д. В этом смысле учебник представляет собой нечто, сочетающее в себе четкую определенность справочника и сюжетную связность книги для чтения.

П. Неужели все так просто: несовпадение подходов — и все?

А. Нет, конечно. При построении учебника нужно учитывать вариативность учебной деятельности: какими путями в принципе может пойти

«сюжетная линия» исследования, происходящего на занятиях. Заметьте, что учитель обретает роль не «озвучивателя» учебника (пусть даже самого лучшего), а лидера, задающего направление дальнейшего исследования.

У. Так значит, по учебнику нельзя будет понять, какой параграф будет изучаться завтра?

А. Конечно.

П. Но позвольте, если я вас правильно понял, авторы такого учебника предполагают, что сюжетная линия занятий пойдет не «по учебнику».

А. Именно так.

П. Однако будем реалистами. Я знаю, что часть моих коллег предпочтет наиболее простой с их точки зрения способ: работать с учебником «как обычно», то есть параграф за параграфом.

А. Того, кто доведет такой «эксперимент» до конца, неизбежно ждет неудача (естественно, если он не догадается об этом заранее). К примеру, в учебнике три раздела: в первом сосредоточены сугубо «теоретические» (безмашинные) вопросы, во втором – компьютерная практика, а в третьем – опять «теория». Согласитесь, что только человек с минимальным педагогическим опытом способен рассчитывать на успех, занимаясь в первом полугодии вообще без компьютеров, в третьей четверти обрушивая на учеников «только машинные» занятия, а в последней четверти внезапно переходя вновь на «теорию».

П. Да, конечно, возникнет потребность в чередовании отдельных тем.

А. Именно это и нужно для преподавания предмета «по-новому». Ведь преподаватель может органично выстроить подобные «чередования» вовсе не единственным образом (хотя и далеко не всяким)! Отсюда становится понятным, что пресловутая вариативность – это не популистский жест по отношению к учителям (дескать, можете делать все, что хотите), а реальная возможность выстроить свой, индивидуальный вариант организации занятий для достижения целей предмета.

П. Но ведь это очень сложная работа. Хорошо бы от чего-нибудь «оттолкнуться»...

А. Разумеется, долг авторов учебника помочь учителю. Лучшее подспорье для этого – вариант поурочного или тематического планирования учебной деятельности. Сопоставляя предлагаемое автором планирование с текстом учебника, учитель осознает, какие именно «столкновения» наиболее плодотворны с точки зрения достижения целей учебного предмета. Имея в руках «отправную точку» для самостоя-

тельных построений, соответствующих как особенностям класса, так и расположенности самого учителя, можно создать «тот самый» единственно нужный в данных условиях вариант.

У. Так все-таки, могут или нет меня спросить о том, чего нет в учебнике?

А. Обратите внимание, что при подобном подходе этот вопрос теряет смысл. Ведь когда учитель не следует «сюжету» учебника, часть материала учебника может вообще не быть использована, а с другой стороны, могут быть затронуты вопросы, в учебник не вошедшие. А если так, то и апеллировать к учебнику, как к некоему «катехизису» просто не приходится: согласитесь, не тот случай.

У. Но я так и не понял: в учебнике будет все, что нужно, или не все?

А. Разумеется, в учебнике будет «почти все». Однако расположено и упорядочено в учебнике оно будет по другому, непривычному с точки зрения занятий принципу.

П. Скажите, а почему Вы так часто используете слова «исследование», «сюжет»?

А. Дело в том, что при таком подходе учебная деятельность приобретает, во-первых, некоторые качества научного исследования, и, во-вторых, определенные черты театрального действия. Учитель ярко, живо, артистично вместе с учениками проводит научное исследование. Учебник же при этом – в лучшем случае путеводитель, приглашающий посетить те или иные достопримечательности, но никак не «маршрутный лист».

П. Так значит, учитель ничем не скован при выборе материала для работы по предмету?

А. По крайней мере учебником – безусловно, нет.

П. Но ведь «оттолкнувшись» от предложенного авторами, можно «выстроить» свой вариант, который не будет иметь ничего общего с тем, что имел в виду автор.

А. Конечно, такая опасность есть. Поэтому всегда должны быть четко оговорены цели, которые должны быть достигнуты в процессе изучения предмета. Они-то и накладывают ограничения на отбор материала и последовательность его подачи. Поэтому, помимо учебника и тематического планирования, авторами учебного предмета должны быть также предложены «Инварианты успешного изучения предмета». Это очень компактный список того, что должно быть обязательно достигнуто при изучении преподаваемого курса. Иначе говоря, нужно четко представлять себе, что же должно оставаться неизменным в каждом из вариантов организации занятий.

П. А если учитель, тем не менее, не соблюдает предложенные авторами «инварианты»?..

А. В таком случае этот учитель преподает учебный предмет по другой программе. Может быть, получше, может, похудшей, но — по другой.

П. А есть ли уже учебники, создаваемые по такому принципу?

А. Таковы, например, учебники по предмету «Информатика и моделирование процессов в природе и обществе» для 6–9 классов средней школы. С его программой можно познакомиться в двух первых номерах ИНФО за 1993 год («Информатика и диалектика»). Сама

программа предмета как раз и представляет собой тот самый «отправной» вариант тематического планирования, от которого можно отталкиваться при построении «текущего» варианта. В журнале также приведены «инварианты», позволяющие установить соответствие работы учителя целям учебного предмета в данном классе.

В конце 1992 года увидел свет первый из учебников по этому предмету — «Информатика-6». Готовятся к выходу и учебники для остальных классов. И все они строятся по тем принципам, о которых мы сегодня говорили.

\*\*\*\*

*Как видим, частный, казалось бы, вопрос о предназначении учебника с неизбежностью выводит нас на размышления об учебной деятельности вообще. В очередной раз подтверждается мысль о том, что для профессионального преподавания учебного предмета необходимо «выходить за рамки» этого предмета и интересоваться тем, от чего зависит протекание учебной (да и вообще познавательной) деятельности.*

**ПОЛЬЗОВАТЕЛЯМ КОМПЬЮТЕРОВ ПК8020, КУВТ «КОРВЕТ» И СОВМЕСТИМЫХ («ЭЛЕКС», «КВАНТ-8», «КОНТУР», «ВЕКТОР», «НЕЙВА», «ФОРМАНТА», «ХОББИ», «СУРА», «ВЕСТА», «ОРБИТА» И ДР.)**

### **ШИРОКИЙ ВЫБОР ПРОГРАММ ДЛЯ ВАШЕГО КОМПЬЮТЕРА:**

системные; инструментальные (языки, средства и системы программирования); прикладные (редакторы, графические редакторы, электронные таблицы, системы управления базами данных, программы печати, архиваторы и др.); образовательные и обучающие (по информатике, иностранным языкам, ПДД и др.); электронные справочники, словари; игровые (более 20 программ).

#### **В ТОМ ЧИСЛЕ:**

- автоматизированная система составления расписания занятий для школ;
- система учета материальных ценностей на складе;
- программа для быстрого запоминания иностранных слов;
- квазирезидентная программа-переводчик с английского;
- системная программа Proffer Commander (аналог Norton Commander);
- многошрифтовой редактор ChiWorker (аналог ChiWriter);
- программы переноса текстовых файлов между «Корветом» и IBM-совместимыми машинами.

#### **А ТАКЖЕ:**

Лабораторный практикум по физике (механика) — 9 лабораторных работ. В комплект поставки входят: набор оборудования, программное обеспечение и методическое пособие. Экспериментальная установка соединяется с компьютером, что позволяет фиксировать перемещение тела при помощи датчиков с последующей обработкой результатов эксперимента на ЭВМ.

**Заявки в произвольной форме направляйте по адресу:**

**123298, Россия, Москва, ул. Народного Ополчения, 38, кор. 2, фирма "КОСТИП".**

**Тел./факс (095) 943-63-48**

**ВАМ БУДЕТ БЕСПЛАТНО ВЫСЛАН КРАТКИЙ КАТАЛОГ ПРОГРАММ  
ДИСКЕТЫ ДЛЯ ЗАПИСИ ПРИОБРЕТАЕМЫХ ПРОГРАММ — БЕСПЛАТНО!  
ЗАКЛЮЧАЕМ С АВТОРАМИ ПРОГРАММ ДОГОВОР НА РАСПРОСТРАНЕНИЕ**



# ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

Б.М.Васильев, А.П.Частиков

## МИКРОПРОЦЕССОРЫ (история, применение, технология)

В ряду гениальных изобретений человечества программируемый логический блок по имени «микрпроцессор» занимает видное место. «Отлитая» в кремнии идея управляемого программируемого логического вычислительного блока была представлена общественности 15 ноября 1971 г.

Импульс был дан в Японии (исходил от японского инженера Масатоси Сима из исследовательского отдела фирмы «Visicom»), но историю микропроцессоров (МП) открыли американские фирмы. Началась эта история с того времени, когда фирма Intel выпустила МП серии 4004 – «интегральное микропрограммируемое вычислительное устройство», представляющее собой однокристалльный центральный процессор, имеющий в своем составе 4-разрядный параллельный сумматор, шестнадцать 4-разрядных регистров, накапливающий сумматор и стек. МП 4004 был реализован как r-канальный МОП-прибор на 2300 транзисторах и мог выполнять 45 различных команд. Последующие поколения МП, представляющие собой 8-, 16- и 32-разрядные приборы, появились соответственно в 1972, 1974 и 1981 гг. И, наконец, пользователям представлен 64-разрядный КМОП, БиКОМП- или ЭСЛ-прибор, содержащий несколько миллионов транзисторов.

По всем показателям темпы микропроцессорной революции не имеют себе равных – они даже превосходят интенсивность развития обычных средств вычислительной техники. Время, прошедшее между изобретением МП и внедрением 32-разрядной архитектуры, составляет 10 лет, т.е. почти вдвое меньше, чем для ЭВМ. Всего лишь за одно десятилетие было ликвидировано значительное отставание микропроцессорной техники от обычных ЭВМ. Более того, микропроцессоры начали выходить на передовые позиции.

Сказанное справедливо в отношении не только производительности, но и совершенства архитектуры.

При оценке параметров микропроцессора и выборе микропроцессорной серии наибольшую роль играет разрядность прибора, которая задает объем обрабатываемых данных. Чем больше разрядность, тем выше производительность и шире возможности адресации. В ранних приборах разрядность регистров, шин управления, а также информационных шин почти всегда была одинаковой. Сейчас такая структура встречается редко. Например, микропроцессор Motorola 6800 имеет 32-разрядную внутреннюю архитектуру, 16-разрядную шину данных и 24-разрядную адресную шину (адресует до 16 Мбайт оперативной памяти). Для удобства такую архитектуру называют 32/16/24. В настоящее время стремятся к большей разрядности, например делают полную 32-разрядную архитектуру (32/32/32).

Если считать, что выпуск предыдущих микропроцессоров должен прекращаться при появлении кристаллов с более высокой разрядностью, то 4-разрядные производились бы всего 1 год, 8- и 16-разрядные – 5 лет.

Разрядность	Модель	Количество транзисторов, шт	Год выпуска	Торговая марка
4	4004	2200	1971	Intel
8	8008	2300	1972	—”—
8	8080	4800	1973	—”—
16	8086	8200	1978	—”—
8	Z80	8400	1976	Zilog
8	8048	12400	1977	Intel
16	8086	29000	1978	—”—
8	6800	5600	1974	Motorola
16	68000	75000	1980	—”—
16	80286	130000	1982	Intel
32	NR-9000	450000	1982	Hewlett-Packard

## Неформальная история

Так назвал первую главу своей замечательной статьи «Эволюция микропроцессоров» Марк Герец, сотрудник фирмы «Viasijp Corporation», фрагменты которой включены в этот обзор (см.: Byte, 1985, №9). Там, где текст этого реферата идет от первого лица, имеется ввиду М. Герец.

В 1959 г. инженеры из фирмы «Texas Instruments» придумали, как расположить более одного транзистора на одной подложке и соединить эти транзисторы без проводов. Таким образом родилась интегральная схема (ИС). В настоящее время такие тоненькие плоские кусочки кремния могут содержать миллионы транзисторов, их называют попросту чипами.

В 1969 г. фирма, просуществовавшая к тому времени всего год, называемая «Intel», объявила о создании микросхемы, содержащей 1 Кбит памяти типа RAM (RAM - оперативное запоминающее устройство). Тогда еще не существовало других микрокомпьютерных чипов, к которым можно было бы привязать эту микросхему памяти, но было много других приложений для такой микросхемы, тем более что эта память была самой емкой из всех тех, что производились до этого времени.

Примерно в то же время - летом 1969 г. - к фирме «Intel» обратилась японская компания «Busicom», занимающаяся выпуском калькуляторов, с заказом на производство набора специализированных микросхем. Микросхемы были разработаны инженерами фирмы «Busicom» для нового семейства калькуляторов этой японской фирмы. Калькуляторы должны были содержать несколько микросхем, каждая из которых состояла из 3000-5000 транзисторов.

В группу инженеров из фирмы «Busicom», которая получила представительство в фирме «Intel», был введен от фирмы «Intel» один из ее разработчиков Марсиан (Тед) Хофф. Хофф познакомился с проектом, предложенным фирмой «Busicom», и нашел, что проект слишком сложен для того, чтобы оказаться рентабельным. До этого он работал с вычислительной машиной PDP-8 фирмы Digital Equipment Corporation, которая имеет очень ограниченный набор команд. Хофф рассудил, что сложность калькулятора может быть сильно уменьшена, если использовать в нем небольшой универсальный процессор. Такой подход, использующий программное обеспечение вместо электронной логики для вычислений, должен резко повысить потребность калькулятора в памяти, но в тот момент фирма «Intel» как раз и занималась

производством устройств памяти. Хофф также понял, что такой процессор может быть применен и в других приложениях, и он продал свою идею руководству фирмы «Intel».

Разработчики фирмы «Busicom» все еще продолжали работать по своему первоначальному проекту, когда Хофф и его группа начали работать над альтернативным проектом. Хотя инженеры из фирмы «Busicom» до предела упростили свой проект, все равно каждая микросхема должна была содержать более 2000 транзисторов, и для работающего калькулятора требовалось 12 таких микросхем. Группа Хоффа считала, что для их процессора потребуется 1900 транзисторов.

Универсальный процессор Хоффа выиграл соревнование с проектом фирмы «Busicom», и фирма «Intel» получила от «Busicom» контракт на производство микросхемы универсального процессора, который позже получил название 4004.

## 4-разрядный микропроцессор

Непосредственное производство микросхем оказалось довольно трудным делом, пока в начале 1970 г. с фирмой «Intel» не начал сотрудничать Федерико Фаджин (который позже основал фирму «Zilog»). Федерико Фаджин довел микросхему от стадии концепции до кремниевого кристалла всего за девять месяцев. Сначала фирма «Intel» продавала процессор 4004 исключительно фирме «Busicom», но летом 1971 г. получила право продавать набор микросхем другим производителям.

В ноябре 1971 г. фирма «Intel» рекламировала свой 4004 как 4-разрядный процессор, выполняющий 60 000 операций в секунду. К февралю 1972 г. фирма «Intel» продала этих процессорных наборов на сумму 85 000 долларов.

Первоначально микропроцессоры выполняли только основные функции, а генераторы синхроимпульсов, запоминающие устройства программ и данных, блоки сопряжения строились на отдельных интегральных схемах. Первые микропроцессоры имели ограниченный набор команд и программировались непосредственно в машинных кодах. Со временем благодаря совершенствованию архитектуры вычислительных систем, появлению языков высокого уровня, росту производительности и гибкости микропроцессорной техники область применения микропроцессоров неизмеримо расширилась. Их используют в промышленности (испытания, управление, контрольно-измерительная аппаратура), в торговле и бытовых приборах. Микропроцессоры ввели в нашу жизнь машинные игры и персональные ЭВМ.

Благодаря включению в 4-разрядные микропроцессоры дополнительных средств интерес к ним не ослабевает. Уже реализуются такие их дополнительные средства:

*встроенные ПЗУ и ЗУПВ большой емкости;  
расширенные наборы команд;  
гибкие протоколы ввода/вывода;  
встроенные преобразователи;  
вспомогательные процессоры;  
драйверы дисплеев.*

Здесь ПЗУ - запоминающее устройство (тип ROM) с постоянной хранением информации, не допускающее ее изменение в процессе работы машины.

ЗУПВ - запоминающее устройство с произвольной выборкой или с произвольным (прямым) доступом к данным.

Исключительная дешевизна и популярность 4-разрядных микропроцессоров была обусловлена: во-первых, отработанной для них р-МОП технологией; во-вторых, достигнутой высокой экономической эффективностью их производства из-за стабильного высокого спроса на эти приборы. Другие виды технологий обеспечивали выпуск микропроцессоров по более высокой цене.

## Рождение 8-разрядного микропроцессора

В то самое время, когда разрабатывался процессор 4004, фирма CTC («Computer Technology Corporation», позже «Datapoint») предложила фирмам - «Intel» и «Texas Instruments» - разработать БИСы для ее нового интеллектуального терминала. Обе фирмы предложили 8-разрядный процессор общего назначения. Можно проследить стереотип развития: 4-разрядный процессор для калькуляторов, поскольку калькуляторы работают в двоично-десятичном коде, и 8-разрядный для терминалов, поскольку терминалы имеют дело с символами.

Фирма CTC не выбрала ни один из этих вариантов; она построила свой терминал на стандартных логических микросхемах. Но тем не менее «Intel», и TI («Texas Instruments») продолжили работу над своими проектами. «Texas Instruments» в конце концов получила патент на свою микросхему, а фирма «Intel» дала рождение процессору 8008.

Процессор 8008, представленный в апреле 1972 г., был первым появившимся на рынке 8-разрядным микропроцессором. Он требовал по меньшей мере 20 микросхем поддержки, зато мог выполнять 45 команд со скоростью 300 000 команд в секунду и мог адресовать 16 Кбайт памяти. В то время это был огромный объем

памяти, и процессор 8008 был серьезным шагом вперед по сравнению с процессором 4004.

Документация по микропроцессорам 4004 и 8008 была сплошной загадкой для лиц, не имеющих специальной подготовки по вычислительной технике. Документация подразумевала, что вы знаете все, что было в этой области до того, как вы начали читать эту документацию. Таким недостатком обладает большая часть технических описаний и по сей день.

Первоначальная цель, которую преследовала фирма «Intel», разрабатывая микропроцессоры 4004 и 8008, была замена «произвольной логики» «множеством малых (МИС) и средних (СИС) интегральных схем, соединенных друг с другом проводами». Очень немногие думали тогда, что эти микросхемы могут пригодиться для универсальных вычислительных машин. Но некоторые фантазеры были заинтригованы самой возможностью иметь вычислительную машину, способную делать хоть что-нибудь. Правда, и раньше существовали наборы, из которых можно было собрать вычислительную машину, но они более подходили для демонстрации принципов вычислительной техники, чем для выполнения вычислительных задач. Доступность процессора 8008 радикально изменила ситуацию.

В 1973 г. фирма «Scelbi Computer Consulting» объявила о создании первой универсальной микро-ЭВМ на основе микропроцессора 8008. За этим последовала машина RGS-008 фирмы «RGS Electronics». Затем в июле 1974 г. журнал «Радиоэлектроника» познакомил читателей с микро-ЭВМ Mark-8, разработанной Джонатаном Титусом.

До этого момента все статьи и рекламные объявления о микро-ЭВМ ограничивались публикациями в радиолобительских изданиях. Mark-8 был первым микрокомпьютером, о котором появилось сообщение в журнале по общим проблемам электроники. Эти первые микрокомпьютеры были все еще более демонстративными, чем полезными, однако микрокомпьютерная революция началась.

Стали очевидны и недостатки процессора 8008. Шинуданных называли «узким горлом», которое задерживало весь процесс обработки. Стало также ясно, что для многих целей применения адресное пространство в 16 Кбайт недостаточно, и его следует расширять.

## Всесильный 8080

В апреле 1974 г. фирма «Intel» заставила нас навсегда изменить сложившееся представление

о вычислительных машинах. Фирма объявила о создании микропроцессора 8080 – значительно более совершенного, чем процессор 8008. Микропроцессор 8080 требовал всего шесть микросхем поддержки, выполнял 75 команд, обладал вдесятеро большей пропускной способностью по сравнению с 8008 и адресовал 64 Кбайта памяти. (Все посчитали, что предел достигнут. Ни одна программа, думало большинство в то время, не может быть такой большой!)

Проект микропроцессора 8080 был предложен Фаджином, но группу разработки возглавлял Масатоси Сима, молодой инженер, которого фирма «Intel» переманила из фирмы «Busicom». Учтя неудобства и ограничения процессоров 4004 и 8008, разработчики нового микропроцессора постарались настолько усовершенствовать свой проект, что новая микросхема стала действительно полезным вычислительным элементом. «Intel 8080» – первый микропроцессор, целью создания которого не была простая замена логики. Этот микропроцессор больше походил на вычислительную машину, чем все те, что были до него, и с точки зрения аппаратуры был гораздо проще в применении.

В январском номере журнала «Популярная электроника» за 1975 г. на первых страницах была помещена первая из серии статей, посвященных построению вычислительной машины «Альтаир 8800», так называемой мини-машины, построенной на основе микропроцессора 8080. Вычислительная машина «Альтаир» была разработана фирмой MITS («Micro Instrumentation and Telemetry Systems»), которая была основана Эдом Робертсом для поддержки и обеспечения его экспериментов в электронике. Весь набор для постройки машины «Альтаир», включая микропроцессор 8080, материнскую плату, блок питания, лицевую панель с множеством лампочек и 256 байтами (не 256 Кбайтами) памяти продавался за 395 долларов (в то время курс доллара в СССР соответствовал 80–90 коп).

Сначала все подумали, что это опечатка. Микросхема, содержащая процессор 8080, представленная всего за девять месяцев до того, одна продавалась за 360 долларов. Однако цена была настоящая – фирма MITS вступила с фирмой «Intel» в специальное соглашение и продала в первый день больше компьютеров, чем надеялась продать за всю жизнь этой разработки.

Вычислительная машина «Альтаир» сыграла значительную роль в том успехе, который сопутствовал микропроцессору 8080, во многом благодаря тому, что теперь программисты

имели основания и хорошую отговорку, чтобы писать программное обеспечение для микропроцессорного чипа. Кроме того, разработанная для машины «Альтаир» архитектура открытой шины (улучшенная версия которой стала шиной S-100/IEEE 696) позволила всем начать разрабатывать периферийные устройства для этого компьютера.

Одним из таких периферийных устройств был контроллер диска, разработанный фирмой «Digital Microsystems», который означал начало использования операционной системы для микропроцессора 8080 и названной CP/M (Управляющая программа для микро-ЭВМ). Операционная система CP/M, детище инструктора Высшей школы военно-морских сил Гарри Килдалла, продавалась за 70 долларов и сыграла основную роль в успехе микропроцессора 8080 и его архитектуры. В результате значительная часть используемого программного обеспечения для микро-ЭВМ либо работала на основе набора команд процессора 8080, либо была прямым развитием какой-нибудь системы, которая применяла этот набор команд.

В микропроцессоре 8080 впервые применена п-МОП технология с обогащением, реализована десятичная и ДКД-арифметика, использовалась 16-разрядная адресная шина.

## Семейство «Motorola»

В ответ на успех микропроцессора 8080 фирма «Motorola» начала работать над процессором 6800, проект которого был разработан Чаком Педдом. «Motorola» была первой фирмой, которая представила серию микросхем для периферийных устройств, спроектированных специально для работы с данным микропроцессором. Периферийные микросхемы включали устройства для параллельного (микросхема 6820) и последовательного (микросхема 6850) ввода-вывода и обеспечивали включение функций ввода-вывода в систему чрезвычайно простым для разработчиков систем образом.

Разработчики были ориентированы на замену логики микропроцессорами, и требовались миникомпьютеры и кросс-ассемблеры для создания программного обеспечения, а это стоило недорого. (Кросс-ассемблер – это транслятор с языка ассемблера, выполняющийся на ЭВМ одного типа и порождающий программу в командах ЭВМ другого типа.) К тому времени еще никто не написал ничего, что бы разяснило, как применять новые микропроцессоры, всем тем, кто ничего не смыслил в их внутренностях или не имел опыта работы с вычислительными машинами.

«MOS Technology» опубликовала рекламное объявление, согласно которому фирма собиралась представить на выставке WESKON в сентябре 1974 г. и там же начать продажу 20-долларового микропроцессора. Микропроцессор фирмы «Technology», получивший наименование 6501, был полностью совместим по выводам с микросхемой 6800 фирмы «Motorola» — вы могли выпаять из платы микропроцессор 6800, вставить на его место 6501, и он начнет работать. Правда, программное обеспечение все равно пришлось бы поменять, поскольку существовали определенные различия в архитектуре микропроцессоров и в их наборах команд. Фирма «MOS Technology» также планировала выпуск серии микропроцессора со встроеной схемой тактирования, наличие которой на отдельной микросхеме требовали все ранее выпускавшиеся микропроцессоры. Эта версия микропроцессора должна была получить наименование 6502 и стоить 25 долларов.

Все производители микропроцессоров пришли в волнение. В то время процессоры 8080 фирмы «Intel» и 6800 фирмы «Motorola» продавались по розничной цене 179 долларов за штуку. М. Герц вспоминает: «Прекрасно помню, как я стоял в вестибюле (т. е., собственно, в гостинице) фирмы «E-Mu Systems» со Скоттом Уздем и Дейвом Рос-сумом, которые как раз незадолго до этого начали встраивать микропроцессор 8080 в свои синтезаторы. Мы разговаривали с коммерческим представителем фирмы «Intel», который считал рекламное объявление просто мистификацией или трюком. Он сказал, что руководство фирмы «Intel» просто не может производить микропроцессоры с такой ценой, и либо рекламное объявление было эффектным трюком, либо речь шла об оптовой цене — за миллионы штук. Я сказал, что не может быть, чтобы микропроцессорные чипы шли по тому же пути, что и микросхемы для научных калькуляторов — сначала сотни долларов, а теперь всего несколько долларов. Коммерческий представитель фирмы «Intel» сказал, что микропроцессоры никогда не будут дешевле 100 долларов.

Точка зрения коммерческого представителя фирмы «Intel» была почти всеобщей в Кремниевой Долине, но «MOS Technology» была расположена на Восточном побережье. Я позвонил туда, и они подтвердили, что это действительная цена и это цена за одну-единственную штуку, и что микросхемы можно купить в их апартаментах в соседнем отеле.

Я пошел туда в тот же вечер и увидел, что апартаменты фирмы заполнены народом. Мик-

росхемы были насыпаны в два огромных круглых аквариума.

Фирма «MOS Technology» тогда же в отеле вечером продемонстрировала еще одну новинку — первые многофункциональные периферийные микросхемы. Фирма поместила память типа RAM, память типа ROM, таймер и ввод-вывод в одну микросхему. Одна версия называлась TIM (монитор терминального интерфейса) и содержала в одной микросхеме полный монитор для общения с серийным терминалом. Вторая схема называлась KIM (монитор клавишного ввода) и была встроена в микрокомпьютерную плату, которая имела клавиатуру, центральный процессор, дисплей, память типа ROM, память типа RAM, параллельный ввод-вывод и стоила 245 долларов. Это уже была полная система. Ни один поставщик микропроцессоров не предлагал ничего подобного раньше. Это было впечатляюще!

В тот вечер я возвращался домой, унося в качестве добычи мой 25-долларовый микропроцессор и два руководства. В тот момент я не отдавал себе отчета в том, что мне придется потратить еще 300 долларов, чтобы создать работоспособную систему, не считая используемого в качестве терминала устройства «ASR-33 Телетайп».

В тот первый день ярмарки WESCON фирмы «Intel» и «Motorola» снизили цену на свои микросхемы до 69 долларов 95 центов. Революция была на полном ходу, электронная промышленность уже не могла повернуть назад.

Микропроцессору 6501 была суждена короткая жизнь. Фирма «Motorola» возбудила против фирмы «MOS Technology» дело, обвинив последнюю в том, что Чак Педдл украл технологию у фирмы «Motorola» и что микропроцессор 6501 создан с нарушением авторских прав, поскольку был взаимозаменяемым со своим аналогом «Motorola 6800». Фирма «MOS Technology» согласилась прекратить выпуск процессора 6501.

Однако многие вычислительные машины, построенные на основе микропроцессора 6502, стали теперь легендой: собственная разработка фирмы «MOS Technology» KIM-1, машины «Apple I» и II, различные модели «Атари», а также среди прочих машины «Коммодор PET» и VIC-20.

## Появление Z80

Где-то в конце 1975 г. или в начале 1976 г. Федерико Фаджин покинул фирму «Intel» и основал новую фирму под названием «Zilog». В 1976 г. фирма «Zilog» объявила о создании микропроцессора Z80, значительно улучшенного варианта микропроцессора 8080, который включал весь на-

бор команд последнего и соответственно позволяя работать матобеспечению, написанному для микропроцессора 8080. Фирма также объявила, что имеются микропроцессоры, которые работают на тактовой частоте 4 МГц, т. е. вдвое большей, чем процессор 8080. Кроме того, микропроцессор Z80 выполнял множество дополнительных команд — общее количество которых составляло 176. Это было слишком хорошо, чтобы казаться правдой, и микропроцессор Z80 был принят с тем же скептицизмом, что и 6502 незадолго до этого. Однако микропроцессор Z80 был реальностью, и действительно имелись партии процессоров, которые работали на частоте 4 МГц.

Несколько плат для набора Z80 с шиной S-100 появились на рынке тут же после разработки микропроцессора, и всем сразу понадобились эти платы. Тем не менее, несмотря на то, что микропроцессор Z80 был значительно мощнее процессора 8080 в смысле набора команд, очень немногие отважились писать программы с учетом преимуществ дополнительных команд Z80. Причина здесь простая: большинство машин, работавших в то время, были созданы на основе микропроцессора 8080, и если вы стали бы писать программы, которые смогут работать только на микропроцессоре Z80, то ваш рынок оказался бы слишком маленьким. Это обстоятельство преследовало микропроцессор Z80 до конца его безоговорочной популярности.

Тем не менее разработчики не стали изменять микропроцессор 8080 в новых системах. Микросхема, содержащая процессор Z80, была несравнимо более легкой в использовании, для нее требовалось только одно напряжение питания, и при этом не нужны микросхемы поддержки. Этот микропроцессор работал гораздо быстрее, даже без использования дополнительных команд.

Процессор Z80 быстро приобрел завидную популярность. До конца 80-х годов он продавался в розничной торговле по цене менее 1 доллара за штуку. Уместно напомнить, что в 1975 г. представитель фирмы «Intel» в ответ на слухи о появлении процессора 6502 стоимостью в 25 долларов утверждал, что никогда микропроцессоры не будут стоить меньше 100 долларов за штуку, т. е. скепсис в отношении цен на микропроцессоры носил в то время массовый характер.

Вместе с микропроцессором Z80 пришел еще один новый принцип микрокомпьютерных схем: встроенная схема регенерации динамической памяти RAM. Динамические устройства памяти типа RAM всегда стоили примерно вчетверо дешевле своих статических конкурентов в расчете на один бит, и это делало их очень

привлекательными для использования. Однако из-за того что приходится постоянно регенерировать такую память (иначе она забывает свои данные), ее использование было чрезвычайно сложным делом. Наличие в микропроцессоре Z80 схемы регенерации динамической памяти было очень большим шагом вперед и позволяло строить системы, которые были гораздо дешевле предыдущих. Такие микро-ЭВМ, как TRS-80 фирмы «Radio Shack», разработанная Стивом Ленингером, использовали именно это преимущество микропроцессора.

Одновременно оптимизировать все характеристики микропроцессора нельзя, и стали изготавливать целые микропроцессорные семейства с единой архитектурой. Каждое семейство имеет свою собственную комбинацию ПЗУ, ЗУПВ и дополнительные функциональные возможности.

Новейшие 8-разрядные процессоры представляют собой упрощенные варианты широко распространенных сегодня 16- и 32-разрядных приборов.

## 16-разрядные микропроцессоры

Тем временем фирма «Intel» признала, что ее микропроцессор 8080 нуждался в модернизации. В 1976 г. фирма объявила о создании процессора 8085. Этот микропроцессор выполнял все команды 8080-го и еще несколько дополнительных команд. Однако «Zilog» застал «Intel» врасплох. Те преимущества, которые имел процессор 8085 перед 8080, не шли ни в какое сравнение с мощностью микропроцессора Z80. С чисто аппаратной точки зрения процессор 8080 был намного приятнее, чем Z80, но последний имел неоспоримое преимущество — он был быстрее. Микропроцессор 8085 работал на частоте 3 МГц, в то время как фирма «Zilog» уже продавала свои микропроцессоры, которые работали на тактовой частоте 4 МГц. Фирме «Intel» нужно было предложить в ответ что-то совсем новое — таким ответом оказался 16-разрядный микропроцессор.

Попытки создания 16-разрядных микропроцессоров делались и раньше. Фирма «National Semiconductor» начала работу по созданию микропроцессорного набора IMP-16 еще в 1972 г. Соответствующий компьютер был действительно построен, но так и не появился на рынке. Создавалось впечатление, что рынок не готов к появлению 16-разрядной микро-ЭВМ и, по всей видимости это так и было.

В литературе некоторые авторы упоминают о создании в 1974 г. фирмой «National Semiconductor»

первого 16-разрядного микропроцессора PACE. А в 1976 г. разработан еще один представитель этого поколения — TMS 9900 фирмы «Texas Instruments». Появились несколько версий исходных микропроцессоров (к ним относятся также МП 8086, 80000 и MC68000), обладающих повышенной производительностью. Трудно дать подробный анализ всех известных сегодня 16-разрядных микропроцессоров, и поэтому далее будет рассказано только о наиболее популярных.

Служи о существовании партии 16-разрядных микропроцессоров ходили годами, но сами микропроцессоры не появлялись до декабря 1983 г., когда фирма «Western Desing Center» (не путать с «Western Digital») объявила о создании 16-разрядной версии микропроцессора 6502.

Итак, возвращаясь к прерванному повествованию, отметим, что фирма «Intel» нуждалась в чем-то, что могло бы конкурировать с микропроцессором Z80. Фирма пришла к выводу, что большинство предыдущих 16-разрядных разработок окончилось неудачей из-за того, что не существовало программного обеспечения, которое могло бы быть легко перенесено на эти микропроцессоры — все это были первые микропроцессоры в жизни выпускавших их фирм.

«Intel», однако, имел серьезное преимущество: растущую базу программного обеспечения для микропроцессора 8080. Фирма решила, что ее 16-разрядный микропроцессор будет прямым развитием процессора 8080. К сожалению, разработчики не сохранили полную совместимость с программами для 8080, но по крайней мере каждый регистр микропроцессора 8080 имел соответствующий регистр в процессоре 8086, что делало возможным создание трансляторов программ 8080 — 8086 и соответственно давало программистам привычную для них отправную точку. Это сыграло большую роль в успехе микропроцессора нового поколения — 8086, который был представлен миру в 1978 г.

Морс, один из главных разработчиков Intel 8086, такими словами определил его назначение: «Да будет удовлетворять потребностям рынка микропроцессор нового поколения, и пусть он будет иметь 1 Мбайт памяти, команды побайтовой обработки строк, во время выполнения которых допускаются прерывания, и полный набор команд десятичной арифметики».

Эти возможности, отвечающие прежде всего потребностям рынка, были реализованы. Кроме того, фирма при его создании позаботилась о том, чтобы была сохранена преемственность в отношении программ, созданных для Intel 8080, выпускавшиеся сериями в несколько сот тысяч ежегодно.

А затем кого-то в фирме «Intel» осенила догадка. Почему бы не сделать аппаратную часть также легко переносимой, как и программное обеспечение? Так родился микропроцессор 8088. Микропроцессор 8088 — это 8086 внутри, но снаружи 8-разрядная шина данных. Когда процессору требуется выбрать 16-разрядное данное, он сначала получает один байт, а затем второй. Программист не должен заботиться об этом, все происходит автоматически на аппаратном уровне. Таким образом, было очень легко расширить имеющиеся 8-разрядные проекты до 16-разрядных.

Можно подумать, что из-за такой процедуры, обеспечивающей совместимость, процессор будет работать вдвое менее эффективно, чем он может. Но разработчики из фирмы «Intel» учли это при проектировании 8086. Этот микропроцессор внутри состоит из двух различных, но связанных процессоров. Один из этих процессоров имеет исполнительное устройство, которое, собственно, и выполняет обработку данных или выполняет команды. Другой процессор получил название «устройство шинного интерфейса» (BIU). Устройство шинного интерфейса выполняет все действия по связям с внешним миром, а также на нем лежит ответственность за генерирование адресов, выдачу данных в систему и получение данных из системы. Внутри BIU имеется очередь. В то время когда исполнительное устройство занято перемалыванием данных, устройство шинного интерфейса получает с шины следующую команду и ставит ее в очередь. Устройство шинного интерфейса микропроцессора 8086 способно идти впереди исполнительного устройства на 6 байт, храня эти байты в своей очереди. Благодаря наличию очереди быстродействие процессора 8088 лишь на 20% (в среднем) ниже, чем у процессора 8086. (См. статью «Основные сведения о микропроцессорах 8086 и 8088 фирмы «Intel», журнал «Байт», июль 1983, стр. 147) Микропроцессоры 8086/8088 были первыми, использовавшими механизм очереди.

Фирма «Intel» ввела еще одну новинку в свое семейство микропроцессоров 8086 — наличие сопроцессоров. Суть этого принципа состоит в подвешивании непосредственно на шину главного процессора другого процессора, расширяющего функции главного. Наиболее примечательным из таких сопроцессоров был процессор 8087, предназначенный исключительно для выполнения арифметических операций, он выполнял их чрезвычайно быстро.

Улучшенный вариант МП 8086 — это МП 80286, выпущенный в 1982 г. Это полностью совместимые приборы, как говорят, совмести-

мы «снизу вверх». Такая совместимость препятствует внесению крупных изменений в архитектуру нового прибора и, соответственно, ограничивает реализацию широкого набора новых функций и типов данных. В МП 80286 при 5-кратном увеличении числа транзисторов был достигнут 6-кратный рост производительности. На одном кристалле площадью 43,9 мм<sup>2</sup> размещены 130 тыс. транзисторов, что позволило добавить еще схемы управления памятью и 4-уровневую систему ее защиты. МП 80286 обладал виртуальной памятью, значительно расширенной виртуальной системой команд.

В 1982 г. фирма «Intel» опять подтолкнула вперед полупроводниковую промышленность, объявив о создании процессора iAPX 286. Архитектура этого микропроцессора была значительной модернизацией процессора 8086, включавшей встроенное управление виртуальной памятью и многие другие особенности, разработанные специально для использования в многозадачных, многопользовательских приложениях. Новый микропроцессор имел режим, в котором он работал как процессор 8086, и фирме удалось добиться существенного увеличения быстродействия в этом режиме. Встроенное управление памятью – это было то, чего ни один микропроцессор не имел в то время. Главное преимущество встроенного контроля памяти состоит в том, что такое управление работает намного быстрее, чем в процессорах, требующих внешнего управления памятью. Использование микропроцессора 80286 фирмой IBM в ее персональной ЭВМ IBM PC AT обеспечило этому микропроцессору успех по крайней мере на следующие несколько лет.

**Микропроцессор MC68000.** В 1977 г. разработчики из фирмы «Motorola» занимались созданием нового процессора для рынка 16-разрядных микропроцессоров, обещая при этом сделать его внутри 32-разрядным. Они также поставили своей целью исключить все специализированные команды и позволить процессору выполнять все операции, на всех регистрах, над всеми типами данных и со всеми способами адресации. Такое свойство называется ортогональностью. Программистам нравится ортогональность, поскольку она означает, что им не нужно запоминать всякие исключения в наборе команд. Результатом усилий разработчиков фирмы «Motorola» было появление в 1979 г. микропроцессора MC 68000.

Микропроцессор MC 68000 – это как раз один из таких процессоров, которые одни любят, а другие терпеть не могут. Равнодушных

здесь очень мало. По сравнению с семейством 8086/8088 этот микропроцессор требовал огромных программистских усилий, прежде чем он смог бы делать хоть что-то. Однако становилась популярной операционная система UNIX, а процессор 68000 выглядел как неплохо приспособленная для UNIX машина. Было представлено несколько машин на базе микропроцессора 68000 с операционной системой UNIX, но, пожалуй, ни одна из них не имела успеха.

Фирма «Motorola» также объявила о своем микропроцессоре 68008, 8-разрядной версии процессора 68000, сходной по концепции с процессором 8088. Однако процессор 68000 не имел настоящего механизма очереди, а это означало, что процессор 68008 работал вдвое медленнее, чем 68000. Конечно, процессор 8088 выглядел на этом фоне предпочтительнее.

По своей внутренней архитектуре процессор 68000 был схемой с микропрограммным управлением, т.е. его внутренние функциональные элементы были универсальными. В такой архитектуре память типа ROM (которая хранит микропрограмму) управляет работой микросхемы. Реакция процессора на каждую команду управляется микропрограммой памяти. Если команда выполняется неправильно, вы можете исправить ее прямо в памяти микропрограммы, и до известных пределов можно даже модифицировать набор команд. (Дополнительный процессор фирмы IBM для ее системы XT/370 есть на самом деле микропроцессор 68000 со специализированной микропрограммой.)

Теперь можно определенно сказать, что микропроцессоры 68000 уже заняли нишу спроса и применения. Например кремниевый микропроцессор 68030 с полным набором команд (CISC-процессор) и 1,5-микронной технологией уже применяется в качестве компонента встроенных систем. Многие его особенности по-прежнему удачно согласуются с ОС Unix и аппаратной средой, содержащей устройства памяти большой емкости. Фирма «Motorola» перешла на его производство в 1,0-микронной технологии с дешевым пластмассовым корпусом. Готовое изделие – VMC68EC030 впервые продемонстрировано в октябре 1990 г.

Вплоть до описываемого времени все процессоры были обычными устройствами с произвольной логикой, которые имели внутри центрального процессора специальные схемы для выполнения каждой отдельной операции. Оба подхода обладают как преимуществами, так и недостатками – микропрограммное управление



дает процессору определенную гибкость за счет потери в быстродействии, в то время как произвольная логика дает быстродействие за счет потери гибкости. Кроме того, исправлять ошибки в системах с произвольной логикой чрезвычайно тяжело — особенно когда разработчик покидает фирму.

**Микропроцессор Z8000.** В фирме «Zilog», например, Масатоси Сима начал работу над 16-разрядным процессором, использующим произвольную логику. Принцип произвольной логики отлично работал в микропроцессорах 8080 и Z80; однако то, что при проектировании 16-разрядного микропроцессора Z8000 Сима пошел по тому же пути, сыграло с фирмой «Zilog» злую шутку. После того как первый кристалл с микропроцессором Z8000 был сделан, но до того, как его группа закончила вылавливание всех ошибок в схеме, Сима покинул «Zilog», чтобы вернуться в «Intel». Фирма «Zilog» так и не смогла выудить все ошибки из схемы процессора Z8000. Кроме того, фирма «Zilog» подала в свое время хороший пример, обеспечив для микропроцессора Z80 полную совместимость со всем программным обеспечением, разработанным для предыдущего поколения 8080. К сожалению, на этот раз фирма не последовала своему же собственному примеру и создала процессор Z8000 совершенно непохожим на процессор Z80. Фирма «Zilog» сделала попытку вмести́ть целый мини-компьютер на один чип, но к сожалению, работу эту сделала не очень хорошо. Различие в системах команд с микропроцессором Z80, невыловленные до конца ошибки в схеме и недостаточно мощные команды из-за принципа произвольной логики, положенного в основу разработки, — все это сыграло свою роль в том, что микропроцессор Z8000 так и не получил успеха (во всяком случае, в 80-х годах).

**Микропроцессор 16032.** В 1981 г. фирма «National Semiconductor» сделала вторую попытку выпустить на рынок 16-разрядных микропроцессоров свой процессор, на этот раз 16032. Микропроцессор 16032 должен был стать 32-разрядным микропроцессором (по внутренней шине данных) с 16-разрядной внешней шиной. Так как фирма «Motorola» так и не сделала своего обещанного арифметического сопроцессора, а фирма «Intel» так и не смогла преодолеть 5-мегагерцевый барьер своего арифметического сопроцессора 8087, все были поражены, когда фирма «National Semiconductor» объявила о создании своего арифметического сопроцессора, который работал на частоте 10 МГц, обеспечивая вдвое большую производительность, чем процессор 8087 фирмы «Intel».

К сожалению, фирма «National» стала первой компанией из производителей микропроцессоров, которая начала продавать абсолютно готовые периферийные микросхемы до того, как стал доступным сам микропроцессор. МП 16032 долгое время имел ошибки в схеме, однако программистам нравится система команд этого микропроцессора, потому что она напоминает им систему команд машин VAX (семейство высокопроизводительных супер-мини-компьютеров фирмы «Digital Equipment»). Поскольку машины семейства VAX оказались столь удачными в качестве носителей операционной системы UNIX, микропроцессор 16032 тоже должен был бы иметь успех, как основа для компьютера с операционной системой UNIX. И сейчас это обстоятельство придает микропроцессору 16032 жизненную силу.

## Микропроцессоры 386 и 486

В 1985 г. наиболее известным был микропроцессор INTEL 80286, имеющий тактовую частоту 6 и 8 МГц. Дальнейшее развитие его — это 80386SX, ранее известный как P9. Микропроцессор 80386SX можно рассматривать как быструю микросхему 80286, которая может запускать 32-битные программы 80386 с очень малой скоростью. Если вы продолжаете работать с ранними версиями DOS вместе с некоторыми прикладными программами OS/2 и, например, с LOTUS 1-2-3, то у вас с 80386 не будет никаких проблем. Настоящие 32-битные микропроцессоры 80386 серии DX выпускаются в четырех модификациях — 16, 20, 25, и 33 МГц.

Микропроцессор 80386 имеет конвейерное обращение, которое в некоторых применениях не отличалось эффективностью. Запись в конвейерном режиме затрудняется отсутствием данных до тех пор, пока не будет завершена текущая операция чтения. (Конвейерная запись не может начинаться до тех пор, пока данные предыдущего конвейерного чтения не покинут шину данных). Попытка реализации одновременной спешки и ожидания.

Еще один недостаток INTEL 8086 и 80386 состоит в том, что невозможно существование адреса свыше 1 Мбайта из-за отсутствия адресного канала A20. В микропроцессоре 80286 для этого применено маскирование (установка в неактивное состояние) адресного канала A20. Хорошо, но несовершенно.

Анализируя ситуацию, фирма «Intel» сумела разглядеть, что расширение функции буфера упреждающей выборке команд процессора

80386 – это не что иное как кэш-память. Далее все просто: помещаем кэш-память в микросхему; ликвидируем механизм упреждающей выборки в том виде, в каком он был описан выше, и получаем микропроцессор 80486.

Intel 80486 состоит из объединенных микропроцессора 80386, расширенного сопроцессора 80387, объединенных кэш-контроллера и 8-кбайтного статического ОЗУ (СОЗУ). Все это размещено в одном модуле, содержащем 1,2 млн. транзисторов.

Следуя историческому пути развития машин общего назначения, платы основной кэш-памяти СОЗУ стали обычными в высокоскоростных системах на основе 80386. Главное отличие между доступом к памяти на микропроцессорах 80386 и 80486 (исключая встроенную кэш-память 80486) состоит в том, что первый допускает только конвейерные, а второй пакетные операции.

Концепция конвейерного механизма еще не исчерпана. Может быть конвейерные пакеты будут доступны на 80586. А пока механизм пакетной передачи 80486 превосходит в 2 раза по производительности механизм передачи данных 80386.

Тактовая частота Intel 80486 может быть 33, 40 и 60 МГц. Этот микропроцессор снабжен самоконтролем при включении и контролем четности. Имеется возможность эмулировать действия канала А20 в машинах типа АТ, что дает заметный выигрыш в производительности. И последнее из преимуществ (в сравнении с 80386), – это наличие специальной BIOS, без которой невозможно воспользоваться улучшенными характеристиками кэш-контроллера и математического сопроцессора. Разработка новых версий BIOS для 80486, видимо, будут проводиться еще долго.

Микропроцессоры типа 80386 и 80486 выпускают и другие фирмы. Отметим из них лишь фирму «Advanced Micro Devices Inc.», которая примерно в 1990 г. анонсировала свой вариант 40-МГц семейства Am386. Вариант с пониженной мощностью потребления для применения в блокнотных компьютерах особенно интересен. Потребляемый ток – менее 1 мА против тогдашних 133 мА у приборов фирмы «Intel». Am386DXL выпускается в четырех вариантах с частотами 20, 25, 33 и 40 МГц.

### Другие 32-разрядные микропроцессоры

Уже в 1980 г. начали выпускать микропроцессоры, у которых внутренние шины были 32-разрядными, а внешние 16-разрядными. Однако эра полных 32-разрядных приборов началась только в

1981 г., когда на рынке появился МП iAPX 432 фирмы «Intel».

Последние разработки фирмы «Intel» – это микропроцессор i486 DX2. DX – это быстрая шина. Только в 1991 г. этих изделий было продано 2 млн. штук. Но пользователи отметили несовершенство этой разработки. В настоящее время в эксплуатации находятся i486DX на 25, 33, 50 МГц и на выходе 66 МГц. В пересчете на Mips скорость DX2-50 составляет примерно 40 Mips, DX2-66 достигает 54 Mips. Венцом этого парада новинок должен стать 586 под кодовым обозначением P5.

Несмотря на проблемы, которыми сопровождался выпуск процессоров фирмы «National Semiconductor» она была первой фирмой, объявившей о создании и начавшей реально продавать настоящие 32-разрядные микропроцессоры. Ее микропроцессор 32032, как и в свое время 16032, программно совместим с 32016. Популярность операционной системы UNIX, была основным фактором успеха микропроцессора 32032, если только на самом деле все обещания относительно того, что семейство микропроцессоров 32000 будет идеальным для операционной системы UNIX, окажутся правдой.

Фирма «Motorola» в настоящее время выпускает пробные образцы своего нового микропроцессора, являющегося полностью 32-разрядным расширением процессора 68000, – процессора 68020, который выглядит многообещающе. Этот микропроцессор имеет одну особенность, которая, возможно, позволит сразу использовать его в большом количестве разработок. Вспомним, что процессор 8088 оказался столь успешным благодаря наличию 8-разрядной внешней шины и соответственно легкости применения в самой различной аппаратной архитектуре. Микропроцессор 68020 позволяет нам динамически выбирать размер шины 8, 16 или 32 разряда по желанию. Одним из факторов, обеспечивающих его высокое быстродействие, является наличие кэш-памяти – логическое продолжение принципа очереди, использованного с большим успехом в микропроцессорах 8086/8088/80286. Отметим, что процессор MC68020 размещен на кристалле площадью примерно 20 мм<sup>2</sup>.

Для 16-разрядных процессоров (68000 или 8086) быстродействие достигает 1 Mips (1 Mips – 1 млн. инструкций в секунду). Микропроцессоры с полем данных шириной в 32 бит достигают скорости от 2 до 3 Mips.

Кэш-память в процессоре 68020 имеет объем 256 байт и работает по несколько иному принципу, чем очередь. Если выполняется команда перехода на адрес, находящийся в очереди, вся очередь сбрасывается и перезагру-

жается. В противоположность этому кэш работает в точности как память, и поэтому переход на адрес внутри кэша не означает его сброса и перезагрузки. Если цикл оказывается достаточно маленьким, то он может работать непосредственно из кэш-памяти. Преимущество состоит в том, что доступ процессора происходит гораздо быстрее, чем к внешней памяти. Таким образом, программы работают быстрее, если они хранятся в кэш-памяти.

Современная 0,8-мкм КМОП-технология в сочетании с тремя слоями металлизации позволила разработчикам Intel достичь уровня гибкости и интеграции, необходимой для размещения более 2,5 млн. транзисторов на кристалле – и благодаря этому создать новый самый мощный микропроцессор семейства i860, модель 860XP. В предыдущем варианте i860 задействовано лишь 1,2 млн. транзисторов.

Повышение плотности компоновки благодаря применению трех слоев металлизации и субмикронной технологии изготовления ИС позволило довести рабочую частоту микропроцессора до 40-50 МГц. При тактовой частоте 50 МГц вычислительная мощность микропроцессора составляет 100 Мфлопс (миллионы операций с плавающей точкой в секунду) и около 50 Mips. Благодаря 64-разрядным шинам пропускная способность шинной структуры микропроцессора составляет 400 Мбайт/с, т. е. соизмерима со скоростными показателями каналов ввода-вывода большинства суперкомпьютеров. Все это позволяет микропроцессору 860XP при работе на той же самой тактовой частоте обеспечить выигрыш в скорости вычисления программ на 30-40% по отношению к предыдущим версиям семейства 860.

МП i860 можно применять в коммерческих компьютерах, в том числе решающих задачи финансового прогнозирования и оперативной обработки транзакций (коротких сообщений, описывающих событие, после чего происходит обновление файла).

Подобный кремниевый 32-разрядный прибор для встроенных систем разработан английской фирмой «Jlmos Ltd» (Бристоль). Ее транспьютер T-9000, выпуск которого начат в 1992 г., обладает пиковым быстродействием 200-250 Mips и 25 Мфлопс. Пропускная способность каждого из четырех каналов связи составляет 100 Мбит/с.

## Новейшие, уникальные, рекордные

Разработаны кристаллы с новой так называемой RISC-архитектурой, имеющие производи-

тельность 4-10 млн. оп/с. Однако независимо от архитектурных различий 32-разрядные процессоры должны взаимодействовать с 16- и 8-разрядными периферийными устройствами, и для этого они обладают возможностью динамически изменять формат магистралей.

В настоящее время около десяти фирм поставляют традиционные и упрощенные по системе команд (RISC-архитектура) 32-разрядные микропроцессоры, а четыре фирмы предлагают микропрограммируемые 32-разрядные кристаллы.

RISC-процессоры программируются меньшим и упрощенным набором команд. Соответственно они часто обгоняют своих более именитых конкурентов за счет того, что успевают выполнить команду за каждый тактовый цикл процессора. Однако скептики не относят это к достоинству, указывая, что RISC-процессоры переносят бремя обработки информации на программное обеспечение – операционную систему и прикладные программы, но последние пока значительно отстают от аппаратной части.

Производительность компьютеров с архитектурой RISC примерно в 5 раз выше, чем для традиционной архитектуры. Для выполнения каждой команды такому компьютеру требуется ровно один такт. Для сравнения: микропроцессору MC 68000 на это нужно пять тактов.

RISC-компьютер был успешно реализован в кремнии в Беркли, и что удивительно – с первой попытки. В настоящее время создатели этой микросхемы работают над увеличением ее быстродействия. Этот процессор имеет 32-разрядную архитектуру.

И наконец, следует сказать об уникальном устройстве, которое получило название «Транспьютер». Транспьютер – это сверхбольшая ИС (СБИС), содержащая микропроцессоры, средства микропроцессорной связи, собственное ОЗУ объемом от 2 до 16 Кб и средства доступа к внешней памяти. Транспьютер предназначен для параллельной обработки. Многие годы еще займет написание программного обеспечения для этого процессора, пока удастся получить полную отдачу от него, но теоретические возможности транспьютера просто поражают воображение.

Новый микропроцессор i860XP выполняет 100 Мфлопс против 80 Мфлопс предшествующего 40-МГц прибора i860. Он имеет пропускную способность шины 400 Мбайт/с против всего лишь 160 Мбайт/с и 16-кбайт кэш-память команд против 8-кбайт. Прибор содержит 2,55 млн. транзисторов, в то время как в самом сложном из предыдущих микропроцессоров серии i860 насчитывалось около 1,2 млн. транзисторов.

По мере перехода к параллельной архитектуре будут разработаны специальные параллельные процессоры и программные языки, которые еще больше увеличат скорость и эффективность такой архитектуры. Об этом свидетельствует транспьютер компании «Inmos» — микропроцессор, использованный в супер-компьютерах серии «Т» фирмы «Floating Point Systems», имеет потрясающую скорость — 10 Mips.

Такую же скорость без особых затруднений — так заявил менеджер фирмы «Intel» Сайва Кумара, — получит на своих микропроцессорах фирма «Intel» в ближайшие несколько лет. К 2000 г. «мозг», управляющий компьютером, будет отдавать команды со скоростью, которая достигнет 30 Mips — почти половину скорости суперкомпьютера Cray-1.

## Заключение

**Тенденции.** За 20 прошедших лет микропроцессорный рынок разделился на две основные части — сектор универсальных репрограммируемых приборов и сектор специализированных встроенных процессоров. В некоторых случаях один и тот же кристалл можно использовать в обеих сферах применения, но зачастую эти сферы требуют разных наборов функциональных возможностей и встроенных функциональных блоков, а также предъявляют различные комбинации требований к техническим характеристикам кристаллов. К сожалению, сегодня перед разработчиком как никогда остро стоит проблема выбора необходимого ему ЦП.

Современные репрограммируемые приборы — важнейшие элементы настольных компьютеров, АРМ и вычислительных серверов — по производительности перекрывают диапазон от нескольких миллионов до 50 миллионов операций в секунду.

На сегодня можно отметить две тенденции развития микропроцессоров. Во-первых, происходит постепенное сближение RISC- и CISC-архитектур по мере того, как в состав каждого из типов ЦП включается все большее число аналогичных функциональных блоков — кэш-память, устройства управления памятью, ускоряющие арифметические сопроцессоры с плавающей точкой и другие. Такая однотипность функциональных блоков все более усложняет определение различий между процессорами без обширных испытаний в реальных сферах их применения. Во-вторых, в обоих классах микропроцессоров уже применяются — или вскоре будут применяться — такие усовершенствования, как супер конвейерные или суперскалярные структуры, предназначенные для исполь-

зования встроенного параллелизма и одновременного выполнения нескольких команд в каждом тактовом цикле.

В большинстве RISC-процессоров второго поколения суперскалярные или суперконвейерные структуры обеспечивают в некоторых случаях повышение предельной производительности более чем до 100 млн. команд/с. Для примера назовем ряд таких новейших RISC-кристаллов — ЦП Snakes компании «Hewlett-Packard Co.», прибор I860XP компании «Intel», кристалл 88110 компании «Motorola Inc.», R4000 компании «MIPS Inc.», SuperSparc компании «Texas Instruments Inc.».

В области CISC-процессоров основное внимание сконцентрировано на приборах семейства 386/486 компании «Intel», а также на аналогичных кристаллах других поставщиков — приборы семейства Am386 компании «Advanced Micro Devices Inc.», семейства Super386 компании «Chips and Technology Inc.», многокристалльных комплектах F86 компании «NexGen Microsystems Inc.», а также, несомненно, на новых разработках компании «Intel». Появились описания усовершенствованных процессоров с малым энергопотреблением, моделей с повышенной производительностью и приборов с еще более высокими уровнями интеграции, например однокристалльного ЦП компании «Chips and Technology» для карманных компьютеров и встроенных контроллеров.

В области процессоров для встроенных систем управления обсуждают четыре новейших RISC-кристалла. Речь идет о новых версиях прибора 29000 компании AMD, новом приборе фирмы «Advanced RISC Machines Ltd.», усовершенствованной модели прибора I960 компании «Intel» и о процессоре семейства Sparc для со-бытийного управления, представляемом отделением «Advanced Products» компании «Fujitsu Microelectronics Inc.».

За всю 20-летнюю историю развития микропроцессоров достигнуто повышение производительности этих приборов на несколько порядков величины. Рост производительности ЦП-кристаллов отчасти устранил необходимость выбора этих приборов на основе только их быстродействия. В число важнейших факторов при выборе микропроцессоров вошли также такие моменты, как разнообразие прикладного программного обеспечения и сроки подготовки производства новых изделий. В области встроенных контроллеров стали существенно выше цениться наличие простых в применении отладочных средств и возможность многократного использования уже существующих программ, поскольку они обеспечивают

сокращение сроков освоения производства и выпуска новой продукции.

**Технология.** В начале развития микропроцессоры изготавливались по р-МОП технологии, затем специалисты стали отдавать предпочтение комплементарной МОП-технологии (К/МОП). Теперь применяется множество разнообразных видов технологии и технологических приемов при изготовлении микропроцессоров: п-МОП технология с обогащением и с обеднением, биполярная технология, технология И<sup>2</sup>Л и др. Например, за 20 лет в США зарегистрированы 237 технологических нововведений, из них 67 революционных.

Основной материал – кремний. Однако ведущие производители вкладывают большие средства в разработку GaAs-процессоров. Ожидается, что этот материал сыграет ведущую роль при создании новых средств, например в условиях повышенной радиации.

Современные быстродействующие транзисторы основаны на комбинации слоев GaAs и AlGaAs. Однако в некоторых случаях эти структуры намного уступают другим структурам, например InP/GaInAs.

Одна из областей где могут быть использованы схемы на таких структурах, – это гибридные приборы, изготовленные на основе полу-

проводников и высокотемпературных сверхпроводников.

Для полупроводникового соединения JnP/GaInAs специалисты Института тонкопленочной и ионной технологии (Германия), разработали такое сочетание слоев в структуре, которое позволяет получать полупроводниковые приборы с очень высокой подвижностью электронов. Подвижность при низких температурах (50 К) достигает 450 000 см<sup>2</sup>/(В.с).

Другое преимущество комбинации InP/GaInAs состоит в том, что позволяет изготавливать ИС, формируя на одном кристалле оптоэлектронные приборы и быстродействующие транзисторы.

В кремнии при напряженности электрического поля 0,3 В/мкм электроны движутся со скоростью 40 км/с тогда как в двумерном электронном газе на границе раздела InP и GaInAs при такой же напряженности поля они достигают в 10 раз более высокой скорости – 400 км/с – при той же напряженности электрического поля.

\*\*\*

А между тем Марк Герец утверждает, что микропроцессор 4004 до сих пор управляет светофором неподалеку от его конторы. И делает вывод, что в будущем он, видимо, заткнет за пояс любые сверхумные новинки. Похоже, так и будет!

## ВНИМАНИЮ ПОЛЬЗОВАТЕЛЕЙ УКНЦ, «Агат-7» и ИВМ

Творческое объединение «ШТУРМ» предлагает для школ, лицеев, гимназий, педучилищ и техникумов следующее программное обеспечение:

- комплекс тестов для учащихся (7-8 лет, 6-11 классы);
- обучающие и контролирующие курсы: математики и русского языка (2-3 классы), математики (9-11 классы) и другие учебные программы, а также системное программное обеспечение.

**БЕСПЛАТНО ВЫШЛЕМ КАТАЛОГ И НАЛОЖЕННЫМ ПЛАТЕЖОМ  
ДЕМОНСТРАЦИОННУЮ ДИСКЕТУ**

**Заказы по адресу:** Россия, 346428, Ростовская обл.,  
г.Новочеркасск-28, а/я 79.

Ю.И.Буцкий

## УЧЕБНЫЕ ПРОГРАММЫ ИНТЕРПОЛЯЦИИ КУБИЧЕСКИМИ СПЛАЙНАМИ

В учебном пособии [1] в главе «Интерполирование функций» уделено большое внимание традиционным методам Лагранжа и Ньютона и значительно меньшее – сплайн-функциям. В то же время практическое применение сплайнов очень широко. Это прежде всего относится к системам автоматизированного проектирования (САПР). Говоря о сплайнах, авторы [1] рекомендуют пользоваться программами, описанными, например, в [2]. Однако в технических приложениях сплайн-интерполяции чаще используется алгоритм Алберга [3]. В настоящей статье предлагается реализация этого алгоритма в виде программ на языке Си для ЭВМ IBM PC/XT/AT и совместимых с ними. Программы могут быть использованы при проведении лабораторных работ в вузах и колледжах, а также в школах и лицеях с углубленным изучением математики и информатики.

При стандартной постановке задачи интерполяции [1] алгоритм построения сплайна, предлагаемый Албергом [3], разбивается на два основных этапа:

**Этап 1.** Вычисление первых производных сплайна  $\dot{S}(x_i)$  в узловых точках  $x_i$ ,  $i = 1, \dots, n$ , где  $n$  – общее число узловых точек.

**Этап 2.** Собственно интерполяция, т.е. вычисление значений сплайна  $S(x_j)$  для любого  $x_j$ , входящего в отрезок  $[x_1, x_n]$ , где  $j$  – счетчик интерполированных точек. Исходными данными служат значения координат узловых точек  $x_i$ ,  $y_i$  и производных сплайна  $\dot{S}(x_i)$ .

Этап 1 реализуется программой splider, этап 2 – программой cubic. Таким образом, пользователю остается написать головную программу, которая будет вводить исходные данные (координаты узловых точек), обращаться к подпрограммам splider и cubic и выводить результаты (координаты интерполированных точек) на печать или экран в виде таблиц или графиков.

### Описание программы SPLIDER

Программа splider реализует решение системы линейных уравнений [3]:

$$\overline{A} \cdot \overline{M} = \overline{C},$$

где  $\overline{A}$  – трехдиагональная матрица, элементами которой являются линейные комбинации абсцисс узловых точек,  $\overline{M}$  – столбец неизвестных (производных сплайна в узлах),  $\overline{C}$  – столбец свободных членов, элементами которого являются линейные комбинации абсцисс  $x_i$  и ординат  $y_i$  узловых точек. Программа формирует матрицу  $\overline{A}$ , столбец  $\overline{C}$  и решает систему методом прогонки.

Поскольку для определения сплайна необходимы граничные условия, на пользователя возлагается их задание через параметры программы splider. Программа имеет следующие формальные параметры:

```
void splider ( int l, int n, double * x, double * y,
              double gran1, double gran2, double *diff1 );
```

**Входные параметры:**

- 1)  $l$  – признак задания граничных условий по следующим пяти вариантам:
  - $l = 0$  – сплайн периодический, задания граничных-условий не требуется;
  - $l = 1$  – на концах сплайна (в первой и последней узловых точках) задаются первые производные;
  - $l = 2$  – на концах сплайна задаются вторые производные. Если они равны нулю, значит, строится "естественный кубический сплайн", реализующий минимум кривизны [3];
  - $l = 3$  – в первой узловой точке задается первая производная, в последней точке – вторая производная;
  - $l = 4$  – в первой точке задается вторая производная, в последней точке – первая производная;
- 2)  $n$  – число узловых точек;
- 3)  $*x$  – адрес массива абсцисс узловых точек;
- 4)  $*y$  – адрес массива ординат узловых точек;
- 5)  $gran1$  – граничное условие в первой узловой точке;
- 6)  $granp$  – граничное условие в последней узловой точке.

**Выходные параметры:**

- $*diff1$  – адрес массива первых производных сплайна в узловых точках (результаты расчета).

**Пример обращения к программе SPLIDER**

Пусть требуется вычислить первые производные сплайна в пяти узловых точках для естественного кубического сплайна (наиболее распространенный на практике случай). Фрагмент головной программы может быть следующим:

```

.....
/* инициализация массивов x и y */
.....
spllder ( 2, 5, x, y, 0., 0., diff1 );
/* вывод результатов, содержащихся в diff1 */
.....

```

**Описание программы CUBIC**

Как уже говорилось, программа cubic вычисляет для заданного  $x_j$  значение сплайна  $S(x_j)$ . По желанию пользователя могут быть вычислены также значения первой и второй производной  $\dot{S}(x_j)$  и  $\ddot{S}(x_j)$  в  $j$ -й интерполированной точке. Программа cubic имеет следующие формальные параметры:

```

int cubic ( int n, double *x, double *y, double *diff1,
           double xj, double *ft, double *f1, double *f2 );

```

**Входные параметры:**

- 1)  $n$  – число узловых точек;
- 2)  $*x$  – адрес массива абсцисс узловых точек;
- 3)  $*y$  – адрес массива ординат узловых точек;
- 4)  $*diff1$  – адрес массива первых производных сплайна в узловых точках;
- 5)  $x_j$  – задаваемая пользователем абсцисса.

**Выходные параметры:**

- 1) \*ft – адрес переменной для записи значения сплайна (результат интерполяции);
- 2) \*f1 – адрес переменной для записи значения первой производной сплайна в интерполированной точке;
- 3) \*f2 – адрес переменной для записи значения второй производной сплайна в интерполированной точке.

Программа возвращает своим значением 0 при нормальном завершении и -1, если задаваемая абсцисса  $x_j$  не принадлежит отрезку  $[x_1, x_n]$ .

Проверить правильность работы программ splider и cubic можно на простых аналитических примерах. Поскольку кубический сплайн представляет собой отрезок кубической параболы, его значения должны точно совпадать со значениями кубического полинома. Рассмотрим, например, уравнение:

$$y = x^3 - x^2 + x. \quad (1)$$

Вычислим значение функции, ее первой и второй производных для  $x = -2, 1, 0, 1, 2$ . Результаты расчета сведем в таблицу:

i	1	2	3	4	5
x (i)	- 2	- 1	0	1	2
y (i)	- 14	- 3	0	1	6
$\dot{y}$ (i)	17	6	1	2	9
$\ddot{y}$ (i)	- 14	- 8	2	4	10

Напишем обращение к программе splider, задав в качестве граничных условий первые производные, т.е.  $l = 1$ ;  $gran1 = 17$ ;  $grann = 9$ :

```
void main ( void )
{
  int n, l;
  double x[5] = { -2., -1., 0., 1., 2. };
  double y[5] = { -14., -3., 0., 1., 6. };
  double diff1[5];
  double gran1, grann;
  l = 1; n = 5; gran1 = 17.; grann = 9.;
  splider ( l, n, x, y, gran1, grann, diff1 );
  .....
```

На выходе получим:

$diff1[0] = 17.$ ;  $diff1[1] = 6.$ ;  $diff1[2] = 1.$ ;  $diff1[3] = 2.$ ;  $diff1[4] = 9.$

Зададим в качестве граничных условий вторые производные, т.е.:

```
.....
l = 2; n = 5; gran1 = -14.; grann = 10.;
splider ( l, n, x, y, gran1, grann, diff1 );
.....
```

На выходе получим:



```
diff1[0] = 17.; diff1[1] = 6.; diff1[2] = 1.; diff1[3] = 2.; diff1[4] = 9.
```

Оба варианта задания граничных условий дают полное совпадение результатов аналитического расчета с расчетом на ЭВМ.

Определим значение сплайна в приведенном примере для  $x_j = 0.5$ :

```
.....
xj = 0.5;
cubic ( n, x, y, diff1, xj, &ft, &f1, &f2 );
.....
```

Результаты расчета: ft = 0.375; f1 = 0.75; f2 = 1. Легко убедиться, что аналитическое вычисление по уравнению (1) дает аналогичные результаты.

Использование сплайна со свободными концами, проходящего через те же точки:

```
.....
l = 2; n = 5; gran1 = 0.; grann = 0.;
splider ( l, n, x, y, gran1, grann, diff1 );
xj = 0.5;
cubic ( n, x, y, diff1, xj, &ft, &f1, &f2 );
.....
дает: diff1[0] = 12.929; diff1[1] = 7.143; diff1[2] = 0.5;
diff1[3] = 2.857; diff1[4] = 6.071; ft = 0.205; f1 = 0.661;
f2 = 2.357.
```

Описанные программы можно также использовать для интерполяции плоских кривых, имеющих более одной ординаты для одной и той же абсциссы, например, траекторий, замкнутых на плоскости. В этом случае строят параметрические сплайны. В качестве параметра чаще всего используют длину  $t$  ломаной, соединяющей узловые точки  $x_i, y_i$ . Задав в первой точке  $t_1 = 0$ , рассчитывают  $t_i$  по формуле

$$t_i = t_{i-1} + \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}.$$

Принимая  $t$  за независимую переменную, можно построить сплайны  $S_x(t)$  и  $S_y(t)$ . Обращаясь два раза к программе splider, можно вычислить значения производных  $\dot{S}_x(t)$  и  $\dot{S}_y(t)$  в узловых координатах  $(t(i), (x(i))$  и  $(t(i), y(i))$  соответственно. Затем можно вычислить интерполированные значения  $S_x(t_j)$  и  $S_y(t_j)$  для любого  $t_j$  из отрезка  $[0, t_n]$ , два раза обращаясь к программе cubic. Пара значений  $S_x(t_j), S_y(t_j)$  дает декартовы координаты интерполированной точки. Для замкнутых кривых строят периодические параметрические сплайны (первая и последняя узловые точки совпадают).

Таким образом, описанные программы можно использовать при интерполировании таблично заданных функций, аппроксимации плоских кривых и решении других задач. Подключение графических функций (например, графических пакетов QC или TURBO C) позволяет иллюстрировать применение программ на экране монитора.

Тексты описанных программ splider и cubic на языке Си приведены ниже. Программы компилировались и тестировались в среде Quick C Microsoft и среде TURBO C++ Borland Inc.

У автора имеется также реализация описанных программ на языках FORTRAN, PL/1, BASIC, которые легко могут быть адаптированы под имеющуюся ЭВМ.

## Текст программы SPLIDER:

```

/...../
void splider ( int l, int n, double *x, double *y, double gran1,
              double grann, double *diff1 )
{
double lamb, mu, l1m1, l1m2, l1m3, l1m4, lalim, mulim, limit, limes,
    h1, hk, cn, hj, hjy, ci, q, u, s, c, a, p, hky, d, xnew, v, t;
int    strike, i, j, k, k1, meter, n1;
static double work1[50], work2[50], work3[50];
--n;
n1 = n - 1;
h1 = x[1] - x[0];
hk = x[n] - x[n1];
limit = 3 * ( y[1] - y[0] ) / h1 - h1 / 2 * gran1;
limes = 3 * ( y[n] - y[n1] ) / hk + hk / 2 * grann;
switch (l)
{
case 0:
    lalim = h1 / ( h1 + hk ); mulim = 1 - lalim; work1[n] = mulim;
    cn = 3 * lalim * ( y[n] - y[n1] ) / hk + 3 * mulim * ( y[1] - y[0] ) / h1;
    work2[n] = cn;
    goto help;

case 1:
    lim1 = 0.; lim2 = 2 * gran1;
    lim3 = 0.; lim4 = 2 * grann;
    break;

case 2:
    lim1 = 1.; lim2 = limit;
    lim3 = 1.; lim4 = limes;
    break;

case 3:
    lim1 = 0.; lim2 = 2 * gran1;
    lim3 = 1.; lim4 = limes;
    break;

case 4:
    l1m1 = 1.; lim2 = limit;
    lim3 = 0.; lim4 = 2 * grann;
    break;
}

work1[0] = lim1; work2[0] = lim2;
work1[n] = 1 - lim3; work2[n] = lim4;

help:
for ( i = 1; i < n; i++ )

```

```

{
j = i - 1; k = i + 1;
hj = x[i] - x[j]; hk = x[k] - x[i];
lamb = hk / ( hj + hk ); mu = 1 - lamb;
h jy = ( y[i] - y[j] ) / hj; h ky = ( y[k] - y[i] ) / hk;
ci = 3 * lamb * h jy + 3 * mu * h ky;
work1[i] = mu; work2[i] = ci;
}
k1 = ( l == 0 ) ? 1 : 0;
q = 0.; u = 0.; s = 1.;
for ( k = k1; k < n; k++ )
{
c = work1[k]; a = 1 - c; p = a * q + 2;
d = work2[k]; u = (d-a*u)/p;
q = - c / p; work1[k] = q;
work2[k] = u; s = - a * s / p;
work3[k] = s;
}
if ( l )
{
xnew = work2[n]; diff1[n] = xnew;
for ( strike = 0; strike < n; strike++ )
{
meter = n1 - strike;
xnew = work1[meter] * xnew + work2[meter];
diff1[meter] = xnew;
}
}
else
{
v = 0.; t = 1.;
for ( k = k1; k < n; k++ )
{
meter = n1 - k + 1; q = work1[meter];
t = q * t + work3[meter]; v = q * v + work2[meter];
work2[meter] = v; work3[meter] = t;
}
a = lalim;
xnew = ( cn - mulim * v - a * work2[n1] ) /
( mulim * t + a * work3[n1] + 2 );
diff1[0] = xnew; diff1[n] = xnew;
for ( i = 1; i < n; i++ )
diff1[i] = work3[i] * xnew + work2[i];
}
}

```

...../

## Текст программы CUBIC:

```

/...../
int cubic ( int n, double *x, double *y, double *diff1, double t,
           double *ft, double *f1, double *f2)
{
double h, a, b, q, a2, b2, c, d, f, g, r, xi, yi, dl, dj, xj, yj, e;
int i, j = 0;

if ( t x[0] || t x[n-1] )
return -1;
for ( i = 1; t x[i]; ) ++i;

j = i - 1;
xi = x[i]; yi = y[i]; xj = x[j]; yj = y[j];
di = diff1[i]; dj = diff1[j];
h = xi - xj; d = h * h;
a = xi - t; a2 = a * a;
b = t - xj; b2 = b * b;
c = 2 * xi + xj - 3 * t; f = d * h; e = c / d;
g = 6. * ( yi - yj ) / f; q = xi + xj - 2 * t;
r = ( 2 * xj + xi - 3 * t ) / d;
*f1 = dj * a * r - dl * b * e + g * a * b;
*f2 = -2 * dj * e - 2 * dl * r + g * q;
*ft = dj * a2 * b / d - dl * b2 * a / d +
      yj * a2 * ( 2 * b + h ) / f + yi * b2 * ( 2 * a + h ) / f;

return 0;
}
/...../

```

## ЛИТЕРАТУРА:

- [1]. *Заварыкин В. М., Житомирский В. Г., Лапчик М. П.* Численные методы. — М.: Просвещение, 1991.
- [2]. *Форсайт Дж., Малькольм М., Моулер К.* Машинные методы математических вычислений. — М.: Мир, 1980.
- [3]. *Алберг Дж., Нилсон Э., Уолш Дж.* Теория сплайнов и ее приложения. — М.: Мир, 1972.

В. Е. Гершензон, Е. Ю. Зейналова, В. А. Кошевой,  
И. В. Левченко, О. Н. Тараканова

## ПРОЕКТ «ПОДСОЛНУХ» — ОБРАЗ НОВОЙ ОБУЧАЮЩЕЙ СРЕДЫ

В настоящее время в системе образования разных стран, в том числе и в России, внедряются новые информационные технологии, предусматривающие работу с приборами и компьютерами. Этот процесс требует перестройки мышления преподавателей, пересмотра содержания учебных программ, методов и организационных форм обучения, разработки и внедрения интегрированных курсов обучения и, в конечном итоге, создания новой обучающей среды.

Под новой обучающей средой мы понимаем особую технологию обучения, которая предусматривает познание окружающего мира путем постановки и решения экспериментально-исследовательских задач с помощью современной техники для сбора, передачи и обработки информации, что способствует развитию творческого потенциала обучаемого, способностей к коммуникативным действиям, интенсификации всех уровней учебно-воспитательного процесса, повышению его эффективности и качества. Одним из примеров создания новой обучающей среды может служить проект «SunFlower» (от англ. — подсолнух). В его основе лежит климатический мониторинг, осуществляемый с помощью метеосистемы «SunF» — недорогой, изящной, современной и надежной станции, подключаемой к персональному компьютеру.

Работы по осуществлению проекта начаты в 1991 г. в рамках сотрудничества Центра Учебных Компьютерных Комплексов Института Новых Технологий, Инженерного Центра «Скан», метеорологической Обсерватории МГУ, творческого коллектива «Геос» и активно ведутся в настоящее время.

Проект «SunFlower» может быть реализован в разных странах и системах образования. Его можно использовать с учетом указаний, содержащихся в образовательных программах разных типов. Он предполагает виды учебной работы и задания, ориентированные на образова-

тельные учреждения различных профилей с учетом особенностей возрастных групп. Конечная цель проекта — не только давать готовые знания, но ученикам в союзе с учителем ставить и решать новые проблемы, принимать или отвергать научные гипотезы, формировать у учащихся навыки исследовательской работы. Все это ориентирует учащихся на самостоятельную творческую деятельность, развивает их индивидуальность и способности, стимулирует постоянный интерес к процессу познания окружающего мира, истории и культуры народов.

Спектр применения системы «SunF» в учебном процессе может быть достаточно широким. Она позволяет не только расширить круг тем таких предметов как «Основы информатики и вычислительной техники», «Физика», «География» и других, но и разработать конкретные направления применения этой системы в новых интегрированных курсах.

Возможности, предоставляемые комплексом «SunF», позволяют внедрять игровые формы обучения не только в системе «ученик» — «SunF», но и реализовывать разнообразные коллективные проекты, например проект «SunFlowersField». Он дает возможность расширить рамки новой обучающей среды и организовать творческую коллективно-игровую деятельность в разных регионах одной или нескольких стран. Примером такой совместной работы учащихся могут быть актуальные исследования по программе экологического мониторинга или проверка гипотезы о достоверности народных примет о погоде у разных народов.

Методическая основа работ по проекту «SunFlower» включает несколько самостоятельных блоков процесса обучения:

- целевой;
- научно-методический;
- учебно-теоретический;
- инструментальный;
- учебно-практический;
- аналитический.

Целевой блок направлен на активизацию творческих усилий учащихся по самостоятельной постановке исследовательских задач и их мотивации. Например, по проверке гипотез и теоретических моделей типа «Погода – растения», «Атмосфера – человек» и др.

Научно-методический блок позволяет познакомить учащихся с организацией научных исследований, их этапами и методами. Например, с методами статистического анализа при упорядочении и обобщении выборочных измерений, проведенных с помощью системы «SunF».

Учебно-теоретический блок предполагает получение и закрепление знаний по разным учебным дисциплинам (география, естествознание, информатика, физика, математика, история и т. д.). Так учащиеся могут получить знания о физических процессах в атмосфере (движения воздуха, нагревание, излучение и др.), об истории развития наук о Земле и цивилизации и т. п.

Инструментальный блок предполагает ознакомление и изучение устройства, принципов работы метеосистемы «SunF» и других технических средств и приборов.

Учебно-практический блок предусматривает развитие умений и навыков учащихся по работе с системой «SunF», сбор и обработку разных видов информации. Источником могут служить личные наблюдения, банки данных разных метеостанций, географические карты, опросные листы, публикации периодической печати и другие.

Аналитический блок включает в себя выявление количественных и пространственных взаимосвязей в природе, принятие гипотезы или отказ от нее, выводы и обобщения в виде отчета о проделанной работе.

Методическая основа проекта работ «SunF» базируется на принципе проблемного подхода в обучении, где учащиеся сами становятся исследователями, открывающими законы. Основной инструмент для создания новой среды – система «SunF».

Структурно комплекс «SunF» состоит из персонального компьютера и системы датчиков для измерения основных метеопараметров: солнечной радиации, атмосферного давления, температуры воздуха и почвы, скорости и направления ветра. Единый стандарт крепления и подключе-

ния датчиков дает возможность пользователю создать собственную архитектуру системы. Передача данных от каждого датчика к комплексу осуществляется специальная электронная система. Метеокомплекс имеет 30 каналов для считывания информации и может быть легко дополнен новыми датчиками. Он работает как в автономном режиме, так и от сети. Унифицированное питание – 12 В, а потребление тока 100 мА. Стыковка станции и ЭВМ происходит через унифицированный порт RS 232.

Авторами проекта создано программное обеспечение, которое реализует запись, хранение и обработку полученных данных. Оно позволяет поддерживать различные режимы работы с метеостанцией, а дружелюбный пользовательский интерфейс позволяет работать с комплексом человеку любого возраста и профессии.

При разработке программного обеспечения стала очевидной необходимость выделения специальных программ-инструментариев – «кирпичников», которые можно использовать при создании различных прикладных программ. В состав этого комплекса входят графические модули, системы управления диалогом, блоки работы с окнами и ряд других модулей.

Созданное программное обеспечение позволяет работать с метеостанцией как в режиме реального времени (когда измеряемые параметры сразу отображаются в графическом виде на мониторе), так и в резидентном режиме (когда в память загружается резидентная программа, в фоновом режиме запрашивающая метеостанцию и записывающая данные на диск). Резидентный режим позволяет использовать компьютер для других программ. Моменты опроса и записи данных со станции для пользователя не заметны и не мешают текущей работе.

Еще один модуль позволяет работать с накопленными данными, проводить обработку и сравнение временных рядов. Эта программа позволяет анализировать данные в различных видах (в виде графиков, гистограмм, сглаженных кривых, таблиц). Существует возможность совмещения в одном окне графиков двух различных параметров, сдвига одного временного ряда относительно другого, проведения регрессивного анализа.

Пользователь может легко изменять экранное представление данных на мони-

торе (открывать нужное число окон, задавать их размеры и местоположение на экране, изменять цвета графики, редактировать надписи и т. д.).

Первая версия программного обеспечения реализована для IBM-совместимого PC/XT-AT. Авторы программного обеспечения стремились сделать работу с комплексом «SunF» максимально простой, не требующей специальных знаний в области компьютерной техники и программирования.

Как уже отмечалось, проект «SunF» апеллирует к нескольким школьным дисциплинам. С точки зрения преподавания физико-технических дисциплин полезным представляется наличие некоторых датчиков, не установленных в режиме постоянных измерений. Учащиеся могут ознакомиться с принципами их действия, исследовать в лабораторных условиях влияние различных физических факторов на показания датчиков. Так, с помощью включенного датчика температуры можно изучать процессы передачи тепла на расстояние, теплопроводность различных материалов и т. п.

Возможность совместной творческой работы учащихся и преподавателя существенно расширяются за счет банка данных многолетних наблюдений за погодой. Это позволяет сопоставлять личные инструментальные наблюдения на комплексе «SunFlower» с архивными метеоданными, а также делать выводы о типичности или нетипичности наблюдаемых вариаций, о рекордных значениях метеопараметров, о

наличии или отсутствии общих тенденций измерения климата, о характерных отличиях метеохарактеристик в различных климатических регионах.

Дальнейшее развитие проекта «SunFlower» предусматривает, прежде всего, пополнение системы сбора информации новыми датчиками, позволяющими перейти от станции погоды к станции экологического мониторинга атмосферы, а также к системе таких станций, объединяющих различные географические регионы. Это принесет в процесс обучения пространственный аспект и придаст ему практическую значимость.

Развитие проекта «SunFlower» предусматривает также разработку нетрадиционных учебных программ, базирующихся на возможностях информационных технологий и позволяющих создать новую обучающую среду. Конкретным примером может служить разрабатываемый в настоящее время проект работ «Народные приметы о погоде». Выбор этой оригинальной темы позволяет не только познакомиться с научными основами географии, астрономии, физики, информатики, лингвистики, обычаев и фольклоре разных народов, о природных особенностях и местах проживания.

Проект «SunFlower» — это вклад его авторов в модернизацию процесса обучения, реальная попытка преодоления границ, разделяющих измерительные и естественные науки.

---

---

*Предлагаем программные средства и периферию для  
БК-0010(01), БК-11М,  
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.Е. Платонов,**

*Республиканский Центр Интерактивных Средств Обучения*

*Министерства Образования РФ*

## **СРЕДСТВА МУЛЬТИМЕДИА В ОБРАЗОВАНИИ**

Данная публикация продолжает тему, начатую статьей «CD ROM технологии» (см. «ИНФО» №4, 1993 г.), и рассматривает вопросы практического применения CD ROM в образовательных комплексах мультимедиа.

Все более широкое применение в области обучения находят средства мультимедиа. Под мультимедиа понимают такой способ и такие средства донесения информации до потребителя, при которых используется несколько сред, например, компьютерная графика, фотография, фрагменты видео, текст, звуковое сопровождение. При этом, как показали исследования, проведенные в ряде стран, эффективность восприятия материала резко возрастает и, соответственно, в несколько раз возрастает эффективность обучения. Однако такой подход предъявляет высокие требования к носителю информации по объему и быстрдействию, а также к средствам вывода информации. Таким требованиям, с учетом доступной цены, на сегодня отвечает комплекс мультимедиа, который включает в себя следующие компоненты:

- персональный компьютер;
- проигрыватель компакт-дисков (CD ROM Drive);
- звуковая плата с наушниками или динамиками;
- компакт-диски (CD ROM) с записанными на них программами.

Рассмотрим более подробно каждый из компонентов. В качестве базового персонального компьютера лучше использовать широко распространенные модели, совместимые с IBM PC AT. Желательно, чтобы процессор был класса Intel 80386 и выше, оперативная память не менее 2 Мб. Для вывода визуальной информации требуются мониторы с хорошим разрешением, как правило, не ниже 640x480 точек на экран, 256 цветов. Такой режим обеспечивается видеоадаптером SVGA. Возможны и другие варианты, в том числе и с процессором 80286, но при этом возможна несовместимость адаптера с программным обеспечением некоторых дисков.

Проигрыватель компакт-дисков достаточно подробно описан в предыдущей публикации. Его главная роль – обеспечить компьютеру доступ к информации, хранящейся на CD ROM. Обычно это плата контроллера, которая вставляется в свободный слот расширения материнской платы компьютера, собственно проигрыватель (CD ROM Kit) и соединительный кабель. Конструктивно CD ROM Kit может быть выполнен в виде устройства (external CD ROM Drive), либо его встраивают внутрь корпуса компьютера (internal CD ROM Drive). В последнем случае установочные размеры устройства часто совпадают с дисководом 5.25". Обращение компьютера к CD ROM производится аналогично обращению к другим дисководам. Сами компакт-диски сменяются как в обычном проигрывателе лазерных компакт-дисков.

Звуковая плата (Sound Card, Soudblaster и т.п.) представляет собой адаптер, который также вставляется в свободный слот расширения материнской платы компьютера и имеет выходной разъем для подключения наушников или динамиков. Функционально это устройство, которое преобразует цифровую информацию, поступающую от компьютера, в аналоговый сигнал, достаточный по мощности для работы наушников или динамиков. Отличаются такие платы по следующим характеристикам:

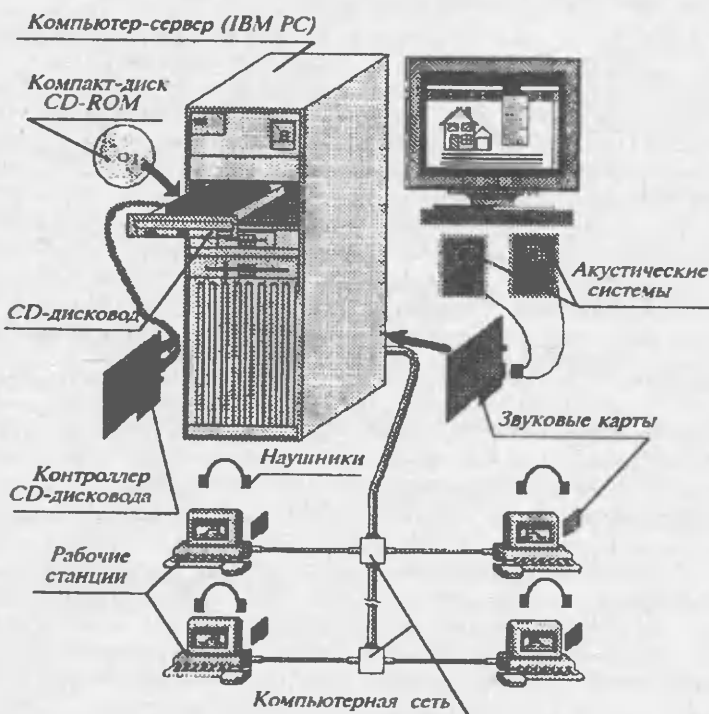
- разрядность преобразования цифрового кода в аналоговый сигнал (желательно, чтобы эта разрядность была не ниже 8);
- разрядность шины данных адаптера – 8 или 16;
- программный драйвер, обеспечивающий работу звуковой карты. Как правило, звуковые карты совместимы с несколькими распространенными драйверами, такими, как AdLib и др.



Качество воспроизведения звука в значительной мере определяется характеристиками наушников или динамиков, рекомендации по выбору которых такие же, как для бытовой звуковоспроизводящей аппаратуры. Некоторые звуковые карты имеют также линейный выход для подключения усилителя и микрофонный вход. С микрофона звуковой сигнал может быть введен в компьютер, что используется, например, для самоконтроля произношения при изучении иностранных языков. Звуковые файлы можно также вводить в свои компьютерные программы.

Следует отметить, что проведение вышеуказанной аппаратной модернизации не требует какой-то особенной конструкции компьютера и специальной квалификации. Для выполнения такой модернизации понадобится всего лишь обычная отвертка.

Завершающим компонентом системы мультимедиа являются сами компакт-диски (CD ROM), на которых записана вся информация. Особенности и достоинства такого способа хранения информации подробно рассматривались в предыдущей публикации. Использование CD ROM обусловлено необходимостью работы с большими объемами информации, что является характерной чертой программного обеспечения мультимедиа. Компании США и Европы (в основном Великобритании) выпустили уже сотни наименований справочных и учебных компакт-дисков: от справочников номеров телефонов и телефаксов до всевозможных иллюстрированных и озвученных энциклопедий, от курсов иностранных языков до развлекательных и познавательных игр. На сегодняшний день Республиканский Центр Интерактивных Средств Обучения (РЦИСО) Министерства образования Российской Федерации установил деловые контакты с десятками фирм, производящих CD ROM. Недавно РЦИСО приступил к созданию отечественного диска. На основе собранной информации сформирован каталог мультимедиа, куда вошли апробированные РЦИСО аппаратные средства мультимедиа и наиболее интересные, с нашей точки зрения, а также доступные по цене диски. Первая версия каталога содержит свыше 200 наименований CD ROM с краткими аннотациями.



Компьютерный класс мультимедиа

В данной публикации мы приводим обзор нескольких зарубежных компакт-дисков с курсами иностранных языков, которые могут быть рекомендованы РЦИСО для системы образования. Их использование представляется наиболее целесообразным, так как они разработаны в сотрудничестве с ведущими университетами, имеют солидную методическую основу и прошли практическую проверку. Для озвучивания дисков привлекались дикторы, для которых данный язык является родным, что гарантирует правильность произношения изучаемой лексики.

«Вводные игры по английскому (французскому, немецкому, испанскому, японскому) языку» (Introductory Games in English (French, German, Spanish, Japanese)). Диски предназначены для начального изучения языков в младших классах школы и для дошкольных заведений. Каждый диск содержит 27 обучающих игр трех уровней сложности, озвученных на одном из вышеназванных языков. В каждой игре на экране компьютера появляется совокупность объектов и в звуковой форме предлагается выбрать один из них. Правильно определенный объект исчезает с экрана. При неверном ответе произносится название указанного объекта, и поиск продолжается. Игра длится до исчезновения всех предметов. При каждом перезапуске объекты переставляются и меняется порядок вопросов. Например, при изучении цветов на экране появляется ряд мячей разного цвета. Перед началом игры можно установить курсор на любой мяч, и диктор будет называть соответствующий цвет. Во время этого режима учащийся знакомится с произношением названий цветов или повторяет пройденное. Затем запускается собственно игра. Теперь голос диктора предлагает учащемуся найти белый мяч, затем черный, затем желтый и т. д. После правильного определения всех цветов можно либо повторить игру на том же уровне, при этом меняется расположение мячей и порядок вопросов, либо перейти на следующий уровень, где появляются новые цвета, такие, как розовый, пурпурный, оранжевый и т. д., в том числе двухцветные мячи.

В других играх аналогичным образом вводятся понятия размера (большой, маленький), формы (круг, квадрат, треугольник и т. д.), а также их сочетания. Например, надо найти на экране маленький оранжевый круг.

Кроме указанных тем рассматриваются предметы одежды, детали лица, домашние животные, числа, отсчет времени и т. д. Общий объем программного обеспечения на диске составляет около 117 Мб. Курс позволяет усвоить сотни слов и выражений. Игры не требуют от ребенка знания букв. Курс может быть рекомендован для детей, начиная с 4-х лет. Предлагаемые игры и истории можно использовать и в домашних условиях, и включать в школьный курс. Данный подход к изучению второго языка был разработан в Сиракузском университете, одним из крупнейших мировых научных центров.

«Открытие языка» (Language Discovery). Диск содержит курс обучения нескольким языкам (английский, французский, немецкий и испанский). Как и вышеописанный курс, он предназначен для младших школьников, но, помимо устной речи, здесь изучается и написание слов. Рассматриваются 40 тем, таких, как «в школе», «на природе», «на кухне», «еда», «в детском парке» и т. п. По каждой теме дается несколько десятков картинок, на каждой из которых изображен какой-либо предмет, относящийся к избранной теме.

Курс предусматривает четыре основных режима работы:

1 — «Выучи слова». Выбрав ученик при помощи «мыши» может указать на любой предмет из изображенных на экране и услышать его название на выбранном языке.

2 — «Найди слово». Вновь задается тема, и появляется картинка с изображением совокупности предметов. Далее возможны два порядка работы. Первый режим — «знакомство с картинкой», в котором ученик, указывая при помощи «мыши» на объект, слушает и запоминает его название. При этом в нижней части экрана дается написание данного слова. Во втором режиме на экране появляется название объекта, который надо найти на картинке. Если объект указан неверно, произносится наименование данного предмета, и поиск продолжается. При правильном ответе задается новое слово для поиска.

3 — «Словарь». На экран выводится весь список слов. При выборе конкретного слова при помощи «мыши» оно произносится, и на экране появляется изображение данного предмета. Можно также увидеть этот предмет на общей картинке по одной из тем.

4 – «Игра». Выбирается тема, и на экране дается совокупность полей, в которых последовательно появляются изображения предметов и шесть возможных ответов, один из которых правильный. При неправильном выборе то поле, где появилось изображение предмета закрывается 20 очков. Если закрытыми оказываются все поля, происходит переход на другой уровень с меньшим числом полей и т. д. Итоги игры заносят в таблицу результатов.

Все режимы взаимосвязаны, и возможен прямой переход от изучения слов по избранной теме к словарю, поиску или игре по этой же теме. В среднем диск охватывает порядка 1000 слов по каждому языку.

«Учимся говорить по-английски» (Learn To Speak English). Этот курс ориентирован на учащихся, уже имеющих начальные знания английского языка, например, в рамках вышеописанного курса. Данный курс затрагивает 30 тем, таких как «знакомство», «встречи», «телефонные переговоры», «покупки», «посещение врача», «общественный транспорт», «паспортный контроль и таможня», «обмен денег» и т. п. Вначале можно прослушать как весь диалог, так и отдельные слова и выражения по выбранной теме. Затем на экран компьютера выводится текст прослушанного диалога, в котором пропущены некоторые слова. Учащимся предлагается восстановить диалог, ввода недостающие на экране слова. При каждом новом запуске меняется набор пропущенных в диалоге слов.

В процессе обучения даются необходимые грамматические пояснения. Предусмотрена возможность ввода в компьютер голоса обучаемого для самоконтроля. Курс может быть рекомендован для изучения языка в старших классах школы, ПТУ, техникумах.

«Школьные проблемы Артура» (Arthur's Teacher Trouble). Этот диск полезно использовать как иллюстративный материал при изучении английского и испанского языков. Предлагается занимательный рассказ для детей по известной книге М. Брауна из серии «Живая книга». История о подготовке к экзамену по грамматике и проблемах, с которыми при этом столкнулся ученик Артур. Помимо прослушивания озвученного текста ребенок может указать «мышью» практически на любое место в картинке, и на экране произойдет какое-нибудь веселое событие. Звуковые эффекты, музыка и много мультимедийных превращают знакомство с языком в интересную игру. Весь текст произносится по выбору на английском или испанском языках.

Рассмотренные выше диски расширяют и дополняют друг друга, что позволяет построить на их основе комплексные курсы изучения иностранных языков с учетом контингента учащихся, переходя от языковых CD ROM. Курсы отличаются друг от друга по методике, темам, объему, звуковому и графическому оформлению и т. д., но всех их объединяет наличие активного режима обучения, что обеспечивает высокую мотивацию, скорость и продуктивность обучения.

Рассмотренные CD ROM представляют лишь малую долю программ мультимедиа. Есть диски, заполненные всевозможными словарями, автоматические переводчиками, контроллерами правописания и т. д. Другие заполнены программами по астрономии, биохимии, математике, истории, химии и пр. Много компакт-дисков с системным и прикладным программным обеспечением. Имеются обширные энциклопедии, вместившие на одном диске десятки томов всемирных музеев. В последующих публикациях по мультимедиа намечено провести следующие:

- энциклопедии и справочники на CD ROM;
- CD ROM для изучения естественных наук;
- оборудование мультимедиа;
- средства создания программ мультимедиа и др.

Это новый информационный мир – мир мультимедиа. Откройте его для себя и своих учеников!

В.Ф. Очков

## ХИМЕРЫ ПРОГРАММИРОВАНИЯ

### Первые ретроспективы и первые химеры

Первую химеру — антирыбу подарил миру Поль Дирак (1902–1984), по-новому решив старую английскую задачу о рыбаках и рыбке:

*«Три рыбака легли спать, не поделив улова. Проснувшийся ночью первый рыбак решил уйти, взяв свою долю. Но количество рыб не делилось на три. Тогда он выбросил одну, а из остатка забрал треть. Второй и третий рыбаки поступили так же. Спрашивается, какое наименьшее количество рыб удовлетворяет условиям задачи».*

Поля Дирака хлебом не корми, а только позволь ему давать разным существительным приставку «анти» — античастица яркий тому пример. В задаче о рыбаках и рыбке Дирак, по-видимому, не изменил своей привычке, дав оригинальный ответ — минус две рыбы. Выбрасываем одну антирыбу — получаем три, забираем одну (треть улова) — получаем минус две и так далее. Но в ответ вкралась ошибка, которую можно объяснить только тем, что у Дирака не было компьютера [1].

Программа 1 решает задачу о рыбаках и рыбке методом ретроспективного перебора, когда задается начальное число рыб (StartFish) и проверяется, удовлетворяет ли оно условиям честного дележа. Если да, ответ готов, нет — начальное число рыб уменьшается на единицу, а расчет повторяется.

Рано или поздно задача будет решена. При этом компьютер, как и Поль Дирак, не смутится фактом появления химер — антирыб. Нормального человека такой нюанс сбил бы с толку, но нормальный человек и до античастиц бы не додумался: талант — это аномалия.

#### Листинг 1

```

Рыбный ряд Дирака (QBasic)
INPUT "Число рыбаков"; Fishers
INPUT "Дельта рыб (с минусом - выбрасывают)"; DeltaFish
INPUT "Начальное число рыб"; StartFish
DO ' Цикл с уменьшением начального числа рыб
  P = StartFish
  FOR I = 1 TO Fishers ' Перебор рыбаков
    P = P + DeltaFish ' Каждый выбросил лишнюю рыбу
    P = P - P / Fishers ' и взял свою долю
  NEXT
  IF P = INT(P) THEN EXIT DO ' Дележ получился
  StartFish = StartFish - 1 ' Дележ не получился
LOOP
PRINT "Ответ "; StartFish; "рыб"
END

```

Расчеты по этой программе дают не одно, а целый ряд решений задачи (назовем его рыбным рядом Дирака):

..., -110, -83, -56, -29, -2, 25, 52, ...

Никто не будет спорить, что минус 29, а тем более минус 110 меньше, чем минус 2. Жаль, что у Дирака не было компьютера.

Если рыбак перед своим уходом не выбросит рыбку, а, наоборот, подловит, то элементы рыбного ряда Дирака поменяют свой знак на противоположный.

Задачу о рыбаках и рыбах можно решить намного быстрее, чем это получается по программе 1. Для этого нужно взять другую опорную точку и «шагать» в противоположном направлении. Вот как, например, выглядит расширенный рыбный ряд Дирака при пяти рыбаках, каждый из которых перед своим уходом по-английски выбрасывает по одной рыбке:

... -6254/-2052, -3129/-1028, -5/-4, 3121/1020, 6246/2041 ...

Этот ряд (числитель — начальный улов, знаменатель — остаток для последнего рыбака) подсказывает более быстрый алгоритм решения задачи Дирака, чем заложенный в программу 1: за остаток рыб для последнего рыбака нужно взять единицу, а потом, если условие честного дележа не соблюдается, к остатку прибавлять по той же единице. Но в этом случае, хоть задача и будет решена быстрее, антирыб не получится: Дирак перешагнул нуль по своей гениальности, а машина, работая по программе 1, — по своей «наивности», не нарушенной приемами оптимизации.

*«Описывай, не мудрствуя лукаво,  
Все то, чему свидетель в жизни будешь...»*

— советовал Григорию Отрепьеву отец Пимен, наставляя его искусству ретроспективы, то есть искусству летописи.

KISS-принцип программирования требует, чтобы поставленная задача решалась наиболее простыми алгоритмическими приемами. С поцелуями термин «KISS» ничего общего не имеет, хотя хорошее отношение к компьютеру в нем просматривается. KISS — это аббревиатура английской фразы «Keep It Simple, Stupid» — «Делай это проще, дурачок!», призывающей решать задачи простыми методами и прибегать к изощренным алгоритмам и программным средствам только в крайних случаях. Нужно всегда помнить, что любой наисложнейший алгоритм машина с помощью транслятора сведет к комбинации полусотни тех простейших операций, которым обучен ее процессор. Чем проще алгоритм, тем быстрее и эффективнее машина это сделает. Попытки ускорить счет или сэкономить память хитрыми приемами не только чреваты ошибками, но и могут привести к обесцениванию результата.

Этим же KISS-принципом руководствуются непрофессиональные программисты, выбирая для решения своих прикладных задач не язык C, не Pascal, не Fortran, а отвергнутый профессиональными программистами BASIC.

### Вторые ретроспективы и вторые химеры

Как запомнить число  $e$  (основание натуральных логарифмов) с десятью цифрами после запятой? Очень просто — две целых семь десятых плюс два раза Лев Толстой: 2.718281828, то есть к числу 2.7 нужно прибавить два раза год рождения писателя. Остается только самая малость — запомнить, в каком году родился Лев Толстой, или, на худой конец, сообразить, что это случилось в прошлом веке, чтобы вспомнить хотя бы три знака числа  $e$  после запятой: 2.718.

Ретроспектива, обращенная в 19-й век, поможет запомнить довольно-таки точное значение одной из важных констант математики<sup>1</sup>.

Как запомнить, что факториал нуля равен не нулю (типичная ошибка), а единице? Очень просто. Нужно применить ретроспективный метод поиска факториала числа: сообщить машине факториал какого-либо положительного числа  $N$  (например,  $5! = 120$ ) и то, что факториал предыдущего числа  $N-1$  равен факториалу первого ( $N!$ ), разделенному на  $N$ .

Компьютер, работая по программе 2, не только правильно определит факториал нуля («плюс единица»), но и родит новых химер — факториалы отрицательных чисел.

```

Листинг 2
Факториал (DBasic)
DECLARE FUNCTION Fakt$(N%)
INPUT "N"; N%; PRINT "Факториал = "; Fakt$(N%)
END

FUNCTION Fakt$(N%)
IF N% = 5 THEN Fakt$ = "120"; EXIT FUNCTION
IF N% > 5 THEN Fakt$ = STR$(N% * VAL(Fakt$(N% - 1))); EXIT FUNCTION
IF N% < 5 AND N% > -1 THEN Fakt$ = STR$(CINT(VAL(Fakt$(N% + 1)) \ (N% + 1))); EXIT FUNCTION
Fakt$ = "минус бесконечность"
END FUNCTION
    
```

Машину нужно только обучить этому — заложить в программу перспективно-ретроспективно-рекурсивный метод поиска факториала на всем целочисленном диапазоне.

### Третьи ретроспективы и третьи химеры

В 1202 году (опять ретроспектива!) Леонардо Пизанский (1180—1240) описал одну из самых первых моделей развития замкнутой биологической системы, населенной условными кроликами [2]. Если

<sup>1</sup> *Есть и другие веселые правила, позволяющие запоминать достаточно сложные последовательности и значения констант. Вот некоторые из них*

1. Цвета радуги: «Каждый Охотник Желает Знать, Где Сидит Фазан» и «Как Однажды Жак-Звонарь Городской Сломал Фонарь».
  2. Число Пи: «Чтобы нам не ошибаться, надо правильно прочесть — 3, 14, 15, 92 и 6».
  3. Порядок дизелей и бемолей: «Жили-были два испанца. Один — высокий и худой, и звали его Фадосоль Релямиси (дизель фа, до, соль, ре, ля, ми, си). А другой низкий и толстый — Симиляре Сольдофа (бемоль си, ми, ля, ре, соль, до, фа)».
  4. Спектральные классы звезд в перечислении от наиболее горячих до самых холодных: «Oh, Be A Fine Girl, Kiss Me» — «О, будь хорошей девочкой, поцелуй меня» (обозначения классов O, B, A, F, G, K, M).
- Читатели могут сами продолжить этот перечень, мы же пока этим ограничимся.*

соответствующим образом определить их плодовитость и долголетие, то численность популяции кроликов будет меняться из поколения в поколение по строгому закону:

Поколение: ... 4 5 6 7 8 9 10 11 ...

Число кроликов: ... 3 5 8 13 21 34 55 89 ...

Читатель, конечно, уже догадался, что речь пойдет о числах Фибоначчи: Леонардо Пизанский более известен под именем Фибоначчи (Fibonacci – сокращение от filius Bonacci, сын Боначчи). Приведенный ряд специально начат не с традиционного места (первое поколение), а с четвертого поколения для того, чтобы задать читателю вопрос, подобный тому, какой стоял в задаче о рыбаках и рыбке и в задаче о факториале: «Чему равно минимальное число кроликов в популяции, каково наименьшее число Фибоначчи?» Нормальный ответ, приводимый в учебниках, – ноль.

Но не будем спешить, вспомним о лаврах Поля Дирака и составим ретроспективно-рекурсивную программу (листинг 3), взяв за базовые числа Фибоначчи не традиционную пару 0 и 1, а 3 и 5.

### Листинг 3

```

Числа Фибоначчи (QBASIC)
DECLARE FUNCTION Fibok (N%)
  INPUT "N"; N%; PRINT "Число Фибоначчи="; Fibok(N%)
END

FUNCTION Fibok (N%)
  IF N% = 4 THEN Fibok = 3; EXIT FUNCTION
  IF N% = 5 THEN Fibok = 5; EXIT FUNCTION
  IF N% < 4 THEN Fibok = Fibok(N% + 2) - Fibok(N% + 1); EXIT FUNCTION
  Fibok = Fibok(N% - 2) + Fibok(N% - 1)
END FUNCTION

```

Машина, как и Поль Дирак, «без страха и упрека» перешагнет ноль на числовой оси и родит новых химер – антикроликов Фибоначчи:

...-13 -8 5 -3 2 -3 2 -1 1 0 1 1 2 3 5 8 13 21 34 55 89 ...

Никто, как и в случае с рыбным рядом Дирака, не будет спорить, что минус один, а тем более минус 13 кроликов меньше, чем ноль кроликов. Да, жаль, что папа Боначчи не мог купить сыну компьютер.

Метод опорных точек (число 120 в программе 2, числа 3 и 5 в программе 3) годен для расчета не только факториала и чисел Фибоначчи, но и рыбного ряда Дирака. По программе 1 достаточно рассчитать только два смежных числа рыб, удовлетворяющих условиям задачи о рыбаках и рыбке. Остальные числа рыбного ряда можно легко рассчитать, приняв во внимание периодичность их появления с интервалом 27 (при трех рыбаках) или 3126 (при пяти рыбаках), когда выбрасывается или поддавливается одна рыбка.

До сих пор разговор шел не о химерах программирования, а только о химерах, генерируемых программами «без шор», которые люди часто надевают на свои глаза и которые мешают им видеть отрицательные числа в ряду минимальных: антикролики, антифакториалы и другие диковинки. Приведенный ряд кроликов и антикроликов позволит нам «родить» чистокровную программистскую химеру, полностью соответствующую принципу единства формы и содержания.

Листинг 4 – это Pascal-программа, преобразованная из BASIC-программы 3 таким образом, что строки в ней – это ... замаскированные числа Фибоначчи от минус седьмого до плюс одиннадцатого. Программа 4, начиная с шестой строки, вполне работоспособна. Первые ее строки, включая и химерические с отрицательным и нулевым числом знаков на строке, не работают, что лишний раз подчеркивает, что в вычислительной математике, как правило, первые семь, а тем более отрицательные, числа Фибоначчи не используются. Полное единство формы и содержания!

Со временем про кроликов Фибоначчи забыли (о существовании химерических антикроликов вообще не упоминали), а числа Фибоначчи нашли применение в прикладной математике.

Но сначала еще одна маленькая ретроспектива.

История приписывает Колумбу два дела: открытие Америки и решение задачи о колумбовом яйце, которое требовалось поставить на острый конец. Многие для этого делали незаметные подпорки или раскручивали яйца... Колумб же просто надломил скорлупу и поставил яйцо торчком.

Языку BASIC история также приписывает два неравнозначных дела. Во-первых, этот язык открыл для многих прекрасный мир программирования, а во-вторых, с его помощью была по-колумбовски решена задача о самопечати программ. Многие, работая с языками C, PL, Fortran, Pascal, пытались

```

Листинг 4
{ Фибоначчи }
function Fibo(
N :
in
t
e
( Минус первая программная строка. Число Фибоначчи = 1 )
( Здесь находится невидимая программная строка с нулевым числом знаков, отмечающим нулевое число Фибоначчи)
g
e
r)
:lo
ngint
) begin
( Начиная с местной строки программа вполне работоспособная. Число Фибоначчи = 8 )
if N < 1 then
case N of 0 :Fibo:=0;
-1 : Fibo:=1; else Fibo:=Fibo(N+2)
- Fibo(N+1); end (case) else if (N = 1) or (N = 2) then
Fibo:=1 else Fibo:=Fibo(N-2)+Fibo(N-1);end;var N:integer; begin read(N);write(Fibo(N));end. (11 89)

```

хитрыми приемами заставить программу при ее прогонке печатать свой собственный листинг. На языке BASIC (версии BASICA, GW-BASIC) это делается одним-единственным оператором LIST.

Колумба и язык BASIC, кроме этого, объединяет также и принцип «Ни одно доброе дело не может остаться безнаказанным!»: Колумб остаток дней провел в тюрьме, с BASIC'ом же работают многие, но зачастую стесняются этого. В этом смысле обиду затаили и сами разработчики BASIC'а – Дж. Кемени и Т. Курц, сотрудники Дартмутского колледжа (США, 60-е годы). Они считают, что все версии языка BASIC, какие сейчас в ходу (с приставками GW, Q, Quick, Visual, Turbo, Power и т.д.) – это своего рода самозванцы, а работать нужно с языком True BASIC – с ортодоксальным Бейсиком, который культивируют Дж. Кемени и Т. Курц, сохраняя чистокровную линию того самого Бейсика – «шестидесятника».

Задача о самопечати – это типичная задача о единстве алгоритма и листинга, о единстве формы и содержания программы. Программа 5 из той же «оперы».

Программа 5 при ее прогонке решает две задачи: основную (поиск минимума функции методом Фибоначчи) и вспомогательную (генерация не всего листинга программы, а только номеров ее строк). В BASIC-программе 5 для расчета двух смежных чисел Фибоначчи, помогающих

```

Листинг 5
* Поиск минимума функции Y методом Фибоначчи (QBASIC)
1 DEF FN Y (X)
2 FN Y = X ^ 2 + X - 3 ' Аналитическая функция
3 END DEF
5 INPUT "A, B, Число приближений к минимуму"; A, B, N
6 Fibo1 = 0 ' Первое базовое число Фибоначчи
13 Fibo2 = 1 ' Второе базовое число Фибоначчи
21 FOR I = 2 TO N ' Расчет чисел Фибоначчи рекуррентным методом
34 Fibo3 = Fibo1 + Fibo2
55 Fibo1 = Fibo2
89 Fibo2 = Fibo3
144 NEXT
233 X2 = A + (B - A) * Fibo1 / Fibo2
377 Y2 = FN Y(X2)
610 FOR I = 1 TO N ' Циклы приближения к минимуму
987 X1 = A + B - X2; Y4 = FN Y(X1)
1597 DO ' Реализация сложного ветвления через однократный цикл
2584 IF X2 > X1 AND Y2 > Y4 THEN B = X2; X2 = X1; Y2 = Y4; EXIT DO
4181 IF X2 <= X1 AND Y2 > Y4 THEN A = X2; X2 = X1; Y2 = Y4; EXIT DO
6765 IF X2 > X1 AND Y2 < Y4 THEN A = X1; EXIT DO
18946 IF X2 <= X1 AND Y2 <= Y4 THEN B = X1; EXIT DO
17711 LOOP UNTIL 1 = 1 ' UNTIL 1 = 1 - это для перестраховки
28657 NEXT
46368 X = (A + B) / 2
73825 PRINT "Y миним. = "; FN Y(X); " при X = "; X
121393 STOP
196418 END

```

оптимальным образом делить интервал неопределенности A-B, заложен не рекурсивный, а рекуррентный метод. Если рекурсивность (листинги 2, 3 и 4), примененную для поиска чисел Фибоначчи, вполне обоснованно сравнивают со стрельбой из пушки по воробьям [3], то рекуррентность в подобной ситуации можно уподобить стрельбе по воробьям из ... крупнокалиберного пулемета. Рекуррентный, а тем более рекурсивный расчет имеет альтернативу и даже две – выживание чисел из таблицы и использование специальных формул.

Калькуляторы отучили нас от устного счета, компьютеры же заставляют забыть полезные методы и формулы, широко использовавшиеся в докомпьютерную эру.

Программа 5 приводится не только для оправдания третьего варианта ее названия «Единство формы и содержания»: номера строк программы 5 – это первые 26 чисел Фибоначчи. Эта колонка цифр подсказывает более оптимальный метод определения смежной пары чисел Фибоначчи – выживание данных из таблицы: вертикальный ряд номеров строк необходимо развернуть на 90 градусов (современные версии языка BASIC – Turbo BASIC, QuickBASIC, QBasic, Visual Basic и другие не нуждаются в нумерации строк), разделить их запятыми, а в начале новой программной строки написать DATA. «Читатель извинит меня от излишней обязанности описывать развязку». Так А.С. Пушкин закончил свою «Барышню-крестянку». Мы же продолжим статью дальше.

В программе 5 алгоритмическое «содержание» выбора (строки с 1597 по 1771) оказалось не в своей форме (SELECT CASE ... END SELECT), а в оболочке одноразового цикла с четырьмя выходами из середины (DO ... LOOP). И это неспроста. Традиционная BASIC-конструкция SELECT CASE ... END SELECT здесь не годится, так как она не может работать со сложными булевыми выражениями, записанными в строках 1597-1771 программы 5. Конструкция IF ... THEN ... ELSE IF ..., предназначенная для решения подобных программистских задач, заставляет забираться в такие дебри вложенных блоков, где очень просто можно запутаться. А вот цикл с выходом из середины – это универсальная управляющая конструкция алгоритмов, годная на все случаи жизни [4].

Избыток логических выражений строк 2584-10946 тоже не случаен. Он дает возможность без комментариев ясно показать (конечно, не машине, а человеку) механизм сужения интервала неопределенности A-B при реализации метода Фибоначчи.

Если рекурсию использовать нельзя (из-за ее неоптимальности), но очень хочется (из-за ее изящества), то можно ... порекомендовать в качестве опорных чисел Фибоначчи брать не 0 и 1 (как в программе 4) и не 3 и 5 (как в программе 3), а 377 и 610, а затем шагать направо прогрессивно-рекурсивным или налево ретроспективно-рекурсивным способом. Выбор опорной смежной пары (377 и 610 или какой-либо другой) – это новая задача внутри старой (поиск минимума функции методом Фибоначчи). Она схожа с другой популярной оптимизационной задачей – поиск в поселке места для магазина (почты, аптеки), чтобы среднее время возвращения из него жителей было минимальным. Интервал нахождения оптимума опорной пары чисел Фибоначчи ограничен с правой стороны закономерностью приближения дроби  $Fibo(N)/Fibo(N-1)$  при больших N к золотому соотношению – 1.61803... Вследствие этого нет нужды рассчитывать большие числа Фибоначчи, а при поиске минимума функции можно громоздкий метод Фибоначчи заменить на более простой – метод золотого сечения.

Попросите знакомого художника разделить отрезок на две неравные части. Можно ожидать, что он сделает это в золотом соотношении.

Подобное чувство меры прививается компьютеру программой 6: интервал неопределенности A-B делится в золотом соотношении, что позволяет при новом приближении к минимуму анализируемой функции опираться на предыдущий расчет и тем самым ускорять поиск.

```

*Поиск минимума у функции Y методом золотого сечения (QBasic)
DECLARE FUNCTION Y (X) : DECLARE FUNCTION Golden (A, B)
INPUT "A . B . Погрешность": A, B, Eps
X2 = Golden(B, A) : Y2 = Y(X2) : X1 = Golden(A, B) : Y1 = Y(X1)

DO WHILE ABS(B - A) >= Eps
    IF Y1 > Y2 THEN
        A = X1 : X1 = X2 : Y1 = Y2
        X2 = Golden(B, A) : Y2 = Y(X2)
    ELSE B = X2 : X2 = X1 : Y2 = Y1
        X1 = Golden(A, B) : Y1 = Y(X1)
    END IF
LOOP

X = (A + B) / 2 : PRINT "Y миним.=" : Y(X) : " при X =" : X
END

FUNCTION Golden (A, B) : Golden = .618 * A + .382 * B : END FUNCTION
FUNCTION Y (X) : Y = X ^ 2 + X - 3 : END FUNCTION

```



Программа б едина по форме и по содержанию – она вписана в рамки структурной диаграммы, стороны которой находятся в золотом соотношении.

Но самое удивительное в том, что и антифакториал и антикролики Фибоначчи «в зачаточном состоянии» существуют уже несколько веков – «ищем рукавицы, а обе за поясом». Дело в том, что у целочисленного факториала и у целых чисел Фибоначчи есть вещественные «родственники» – гамма-функция и специальная формула для расчета чисел Фибоначчи, соответственно.

История гамма-функции похожа на историю языка С (Си): был язык А (Эй) – не пошел, разработали язык В (Би) – тоже не привился, а язык С приобрел огромную популярность. Была альфа-функция, была бета-функция, но математический мир запомнил и широко использует в расчетах только гамма-функцию.

Ни факториала, ни гамма-функции нет среди встроенных функций четверки самых популярных языков программирования (BASIC, Pascal, C, Fortran), хотя сам факториал незримо там присутствует – без него не вычислить ни логарифма, ни какой-либо тригонометрической функции. Факториал и гамма-функция непременно входят в состав встроенных функций четырех самых популярных математических пакетов «программирования без программирования» (MathCAD, Maple, MatLab, Mathematica) [5]. Гамма-функция в целых отрицательных точках стремится либо к «плюс-», либо к «минус бесконечности», формула же для расчета чисел Фибоначчи прекрасно работает и на отрицательной половине числовой оси – см. рисунок, выполненный в среде пакета MathCAD 3.0 for Windows.

Работая в среде MathCAD, можно доказать существование не только антифакториалов и антикроликов, но и суперантифакториалов и суперантикроликов, свернув с действительной числовой оси в комплексную область.

Резюме: «Нет ничего практичной хорошей теории», а «самая лучшая программа – это формула».

Среда MathCAD при анализе функций позволяет не только свободно гулять вдоль всей числовой оси в прямом и обратном направлениях, но и переходить в область комплексных чисел, где суперантикролики не просто обитают, но и группируются в изящные поверхности.

Мы не будем лишать читателя удовольствия быть первооткрывателем суперантирыб Дирака и оставляем ему возможность переписать и прогнать программу 1 на языках С или Fortran, где есть комплексный тип переменных. Статью же на этом месте заканчиваем, опасаясь «забраться в дебри, в которые может забираться, не рискуя свернуть себе шею, лишь очень образованный человек» (М. Булгаков, «Мастер и Маргарита»).

**Литература**

1. *Очков В.Ф.* «Двенадцать программ с дублями и эпиграфами, или три триады программирования, а точнее, третий лишний, а второй неправильный». Монитор N4, 1993.
2. *Гарднер М.* Математические новеллы. Пер. с англ. — М.: «Мир», 1974.
3. *Бройтман Д.Э.* «Рекурсия или Не стреляйте из пушки по воробьям». Монитор N1, 1992.
4. *Очков В.Ф.* «Turbo Pascal 7.0. Взгляд со стороны». КомпьютерПресс N7, 1993.
5. *Очков В.Ф., Пухначев Ю.В.* Уроки для пользователей IBM PC. — М.: «Финансы и статистика», 1992.

Суперкалькулятор MathCAD 3.0 for Windows  
 1. Гамма-функция, факториал и антифакториал  
 $X = -5.05..-4.95..5.05 \quad N = 1..5$

2. Функция Фибоначчи и интегралов  
 $Fibo(X) = \frac{1}{\sqrt{5}} \left[ \left( \frac{1+\sqrt{5}}{2} \right)^X - \left( \frac{1-\sqrt{5}}{2} \right)^X \right] \quad X = -10..-9.9..10$   
 $N = -5..5$

3. Гамма-функция комплексной переменной и суперантифакториал  
 $i = \sqrt{-1}$  Минимая единица  
 $\Gamma(i) = -0.155 - 0.498i$  Первый суперантифакториал  
 $\Gamma(-i) = -0.155 + 0.498i$  Второй суперантифакториал  
 $X = -10..-9.9..10$

4. Функция Фибоначчи комплексной переменной и суперантикролики  
 $Fibo(i) = 0.379 + 0.216i$  Первый суперантикролик  
 $Fibo(-i) = -0.379 - 0.216i$  Второй суперантикролик  
 $k = 0..25 \quad l = 0..10 \quad M_{\{k,l\}} = Fibo \left[ \frac{k-S}{5} + \frac{l-20}{10} - 1 \right]$

# МУЛЬТИМЕДИА

## ОБОРУДОВАНИЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:

### РАБОЧЕЕ МЕСТО МУЛЬТИМЕДИА

**Состав:**

PC AT 386SX-33MГц/RAM 4M6/HDD 120M6/FDD 1.44M6 & 1.2M6/SVGA/2S/1P/101 keys  
l/c keyboard/CD-ROM Drive/Sound card/наушники

Цена \$ 1700

### СЕТЕВОЙ КЛАСС МУЛЬТИМЕДИА

Комплекс предполагает установку лазерных дисков (CD-ROM) на учительском рабочем месте. Ученики получают доступ к информации по локальной сети и могут прослушивать звуковое сопровождение при помощи наушников.

**Состав**

*Учительское рабочее место(1)*

PC AT 386SX-33MГц/RAM 4M6/HDD 120M6/FDD 1.44M6 & 1.2M6/SVGA/2S/1P/101 keys l/c  
keyboard/Ethernet card/CD-ROM Drive/Sound card/наушники

*Ученическое рабочее место(10)*

PC AT 386SX-33/RAM 2M6/SVGA /2S/1P/101 keys l/c keyboard /Ethernet card /Sound card/  
наушники

Цена базового комплекта

\$ 12136

Цена дополнительного рабочего места

\$ 987

### ДОПОЛНИТЕЛЬНАЯ КОМПЛЕКТАЦИЯ МУЛЬТИМЕДИА

Предлагается дополнительная  
доукомплектация имеющихся персональных  
компьютеров (IBM PC) средствами

мультимедиа:

- ✓ CD-дисковод (CD-ROM Drive) \$ 400
- ✓ звуковая карта \$ 120
- ✓ динамики \$ 12
- ✓ наушники \$ 8

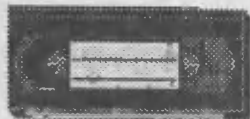
### ШИРОКИЙ ВЫБОР КОМПЬЮТЕРНЫХ ДИСКОВ CD-ROM

по следующим темам:

- ✓ иностранные языки (английский,  
немецкий, французский и др.)
- ✓ энциклопедия (мировой атлас,  
животные, растения и др.)
- ✓ естественные науки
- ✓ искусство

Цена \$50-\$150

## В ПОМОЩЬ УЧИТЕЛЮ



Имеется в продаже видеокассета в поддержку  
школьного курса *КАНОНИКИ*  
продолжительность - 90 мин.

Цена \$15

Оплата в рублях по текущему курсу ММВБ

**Наш адрес:** 125315, г. Москва, ул. Часовая, 21-6, РЦИСО

**Банковские реквизиты для Москвы и Московской обл:**

расч. счет 1609325 в Ленинградском отделении МББ. МФО 201694

**Для других регионов:** кор.счет 48616100 в РКЦ ГУ ЦБ РФ МФО 201791

☎ Телефоны (095)155-87-47 (095)155-87-37

Телефакс: (095)155-87-27

E-mail: rcime@aha.msk.su

# КЛУБ УКНЦ

В. И. Сулханов

## «ЭЛЕКТРОНИКА БК-0010» и УКНЦ

Многие владельцы БК, желая работать на профессиональной машине под стандартной ОС, приобрели УКНЦ (МС-0511). Продавать БК не хочется, уже накоплено несколько тысяч программ, да и детишки (внуки, племянники, гости) с удовольствием на ней играют.

А не создать ли многопроцессорный комплекс? Технически это сделать совсем не сложно, — достаточно соединить параллельные порты машин.



Теперь, если написать соответствующее программное обеспечение на обе машины, появятся следующие возможности:

- можно хранить программы БК на флоппи-дисках УКНЦ, что несомненно дешевле и оперативнее.
- можно писать и транслировать БК-ные программы на УКНЦ, и сразу же исполнять их на БК.
- можно (для виртуозов) для ускорения выполнения задания исполнять его сразу на всех процессорах.

Ниже приводится текст программы на ассемблере для БК, которая проверяет состояние порта УКНЦ (3-я и 5-я строки программы), и при получении от УКНЦ соответствующей команды либо читает файл из УКНЦ, либо читает первый встреченный файл с магнитофона и направляет его в УКНЦ. Программу разумнее всего "зашить" в ПЗУ типа 573РФЗ. Можно использовать ОЗУ на микросхеме 537РУ11А с 3-х вольтовым аккумулятором или миниатюрной батареей. Данная микросхема имеет объем 512 байт и потребляет в режиме хранения доли микроампера. Эти микросхемы имеют встроенный интерфейс МПИ и непосредственно устанавливаются в свободное место в блоке МСТД. Мы рекомендуем программировать микросхему на начальный адрес 120000 и отключить ФОКАЛ. Адрес 120000 удобен тем, что после включения машины программа сразу начинает работать и не требуется никаких действий на БК. Если это вас не устраивает, то лучше использовать для ОЗУ микросхему 537РУ11А, начальный адрес которой легко изменять, записывая в ячейку 177760 значение необходимого вам начального адреса.

;программа на стороне БК -----

```

BK:  MOV  #177714,R1
      MOV  (R1),R0      ;ЧИТАЕМ ПОРТ
      CMP  #333,R0B
      EQ   BKU
      CMP  #111,R0
      BEQ  UKB
      BR   BK

```

;----- ЧИТАЕМ ИЗ УКНЦ

```

UKB:  MOV  #177714,R1
      JSR  PC,CH
      RTS  PC
      HALT

```

;----- ПИШЕМ В УКНЦ

```

BKU:  JSR  PC,ZAP      ;ЧИТАЕМ ПЕРВЫЙ ВСТРЕЧНЫЙ
      TSTB @#301      ;ЕСЛИ ПЛОХО ПРОЧИТАЛОСЬ
      BEQ  MB3
      MOV  #40,R1
      MOV  #7,R0
MB2:  EMT  16          ;ТО ГУДИМ 40 РАЗ
      SOB  R1,MB2
      BR   BK          ;И СНОВА ЧИТАЕМ
MB3:  MOV  #12,R3      ;КОЛ-ВО СЛОВ В ИМЕНИ...
      MOV  #346,R2     ;НАЧАЛО ИМЕНИ
      MOV  #1000,R0    ;НАЧАЛЬНЫЙ АДРЕС ПРОГРАММЫ
      MOV  R0,R4       ;ЗАПОМНИЛИ ДЛЯ П/П ПЕРЕПИСИ НА ДИСК
MB4:  MOV  (R2)+,(R0)+ ;ДАЛЕЕ ИМЯ,
      SOB  R3,MB4
      MOV  @#266,R5    ;ДЛИНА МАССИВА
      ADD  #24,R5      ;ПОЛНАЯ ДЛИНА МАССИВА В БАЙТАХ.
      MOV  #222,(R1)   ;СООБЩАЕМ О ЗАВЕРШЕНИИ ЧТЕНИЯ С МЛ
MB5:  CMP  #123,(R1)
      BNE  MB5
      JSR  PC,PER      ;ПЕРЕДАЕМ НА ДИСК И...
      JMP  BK          ;ВСЕ С НАЧАЛА...
;П/П ПЕРЕДАЧИ НА ДИСК ЧЕРЕЗ ПАРАЛЛЕЛЬНЫЙ ПОРТ
PER:  MOV  #177714,R1
PE2:  BIT  (R1),#400   ;ПРОВЕРЯЕМ ГОТОВНОСТЬ УКНЦ
      BEQ  PE2
      MOVB (R4)+,R0   ;БЕРЕМ ОЧЕРЕДНОЙ БАЙТ
      BIS  #400,R0    ;ДОБАВЛЯЕМ СТРОБ
      MOV  R0,(R1)    ;И ПЕРЕДАЕМ В ПОРТ.
P2:   BIT  (R1),#400  ;ПРОВЕРЯЕМ СНЯТИЕ ГОТОВНОСТИ УКНЦ

```

```

BNE P2
CLR (R1) ;ОЧИЩАЕМ ПОРТ
SOB R5,PER ;И ВСЕ ПОВТОРЯЕМ, ПОКА НЕ КОНЧИТСЯ ФАЙЛ.
RTS PC ;И ВСЕ....

```

## ;П/П ЧТЕНИЯ ПЕРВОГО ФАЙЛА

```

ZAP: JSR R4,@#110346 ;ЗАПОМИНАНИЕ R0-R4
      MOV #320,R1 ;АДРЕС БЛОКА ПАРАМЕТРОВ
      MOV R1,@#306
      MOV #177716,R3 ;АДРЕС ПОРТА
      CLR @#300 ;ЗДЕСЬ ТИП ЗАПИСИ И ОТВЕТ
      CLR @#302 ;ПОДСЧИТЫВАТЬ КС
      MOV @#4,-(SP)
      MOV #116214,@#4 ;НОВЫЙ ВЕКТОР ПО СТОП
      MOV SP,@#310 ;УКАЗАТЕЛЬ SP ПРИ ВХОДЕ
      MOV #20,(R3) ;ПУСК ДВИГАТЕЛЯ
      MOV #1024,@#264 ;В ОЗУ
      MOV #20040,@#326 ;ИМЯ НАЧИНАЕТСЯ С ПРОБЕЛА
      MOV #1,@#304 ;ШАГ ПО ОЗУ
      MOV #40,R5
      JSR PC,@#116712 ;ПОИСК НАЧАЛА ФАЙЛА
      MOV @#306,R1
      ADD #26,R1
      MOV #24,R2
      JSR PC,@#117260
      MOV #10,R2
M1: MOV -(R1),177754(R1)
     SOB R2,M1
     MOV -(R1),@#266
     JSR PC,@#117154 ;СОБСТВЕННО ЧТЕНИЕ ФАЙЛА
     MOV #220,(R3) ;ОСТАНАВЛИВАЕМ МОТОР И
     JMP @#116212 ;УХОД, ТАМ ЕСТЬ RTS PC

```

## ;П/П ЧТЕНИЯ ИЗ УКНЦ

```

CH: MOV #346,R2
     MOV #24,R3
MC1: JSR PC,СВ
      MOVB R0,(R2)+
      SOB R3,MC1 ;РАСПИСАЛИ ПО БЛОКУ ПАРАМЕТРОВ
      MOV @#346,R2
      MOV R2,@#264
      MOV @#350,R3
      MOV R3,@#266 ;ЗДЕСЬ ДЛИНА В БАЙТАХ
MC2: JSR PC,CS
      MOVB R0,(R2)+

```

```

SOB   R3,MC2
RTS   PC
;П/П ЧТЕНИЯ ОДНОГО БАЙТА
CB:   MOV   #200,(R1)      ;ШЛЕМ ГОТОВНОСТЬ
CS1:  BIT   (R1),#400     ;ЕСТЬ СОПРОВОЖДЕНИЕ?
      BEQ   CS1
      MOV   (R1),R0       ;ЗАБИРАЕМ КОД
      CLR   (R1)         ;И СНИМАЕМ ГОТОВНОСТЬ
CS2:  BIT   (R1),#400
      BNE   CS2         ;СНЯЛИ СОПРОВОЖДЕНИЕ?
      BIC   #177400,R0   ;ЗАПОМИНАЕМ БАЙТ
      RTS   PC          ;И ВСЕ...
.END   BK

```

Вся программа занимает менее 310 байт.

На стороне УКНЦ создана программа (BUB.SAV), высвечивающая директорию рабочего диска. Одно из имен подсвечено. Перемещая подсветку, вы выбираете нужную программу для исполнения на БК.

Работа с системой исключительно проста. Включаются обе машины. Дисплей подключается к УКНЦ. Если БК используется для игр, то ее можно подключить к цветному телевизору. Предполагаем, что на рабочей дискете предварительно записаны программы БК. Загружаем систему на УКНЦ и вызываем программу BUB.SAV. Перемещаясь по директории, находим нужную игру и нажимаем <ВВОД>. УКНЦ переписывает программу с дискеты в БК и сообщает вам об этом. Если загруженная программа с автозапуском, она немедленно начинает исполняться. Никаких специальных действий на БК производить не нужно, т.к. БК после включения или сброса автоматически переходит на выполнение программы BK.SAV.

Если вы еще не создали дискету с программами БК, то следует подключить к БК магнитофон, установить кассету с программами, которые необходимо переписать на дискету, и включить систему. Запускается программа BUB.SAV, в директории выбирается последняя пустая строка и нажимается <ВВОД>. Все остальное сделает УКНЦ. БК начнет читать с кассеты, и после каждого прочитанного файла на дисплее УКНЦ будет появляться имя прочитанного файла. Вам остается принять решение, записывать ли прочитанную программу на диски УКНЦ и не следует ли изменить имя программы. Все необходимые подсказки появляются на экране. Корректировать имена приходится потому, что УКНЦ хранит файлы с именами размером не более 6 букв, и только латинских. Впрочем, уникальное имя программы БК не теряется и при переписывании с диска УКНЦ на БК оно, как и положено, занимает свое место начиная с адреса 352.

Если вы решили соединить свои УКНЦ и БК и у вас возникли проблемы, вам помогут в клубе УКНЦ, информация о котором была опубликована в «ИНФО» №6 за 1992 г.

А.И.Акулин

## УКНЦ: ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ РАБОТЫ

Компьютерные классы УКНЦ (МС0511, МС0202) составляют примерно треть парка школьных ЭВМ в России. Несмотря на неплохие характеристики, многие школы начинают отказываться от этих компьютеров, заменяя их на дорогостоящие IBM-совместимые машины.

Цель данной статьи — показать, как без существенных материальных затрат превратить УКНЦ в действительно работоспособный и эффективный компьютер, значительно превосходящий IBM PC AT-286 по соотношению мощность / стоимость.

Вашему вниманию предлагаются доступные в настоящее время периферийные устройства, использование которых и приводит к желаемому результату.

### Электронный диск (ЭД)

Читатель, вероятно, знаком с драйверами системы RT11 типа KD, VM, RD, позволяющими за счет урезания видео-памяти УКНЦ создавать небольшой квази-диск (объемом 100-150 блоков), на котором можно быстро работать с программами и данными. Однако такого объема недостаточно для нормальной работы с операционной системой, да и цветов на экране остается два вместо восьми.

Все эти недостатки пропадают при установке на УКНЦ периферийного устройства «Электронный диск» объемом 512 Кб либо 1 Мб. Электронный диск представляет собой электронную плату памяти, которая устанавливается в свободный слот (раз'ем) РМП (рабочего места преподавателя) или РМУ (рабочего места ученика). После включения питания вы копируете файлы операционной системы и программ с НГМД на Электронный диск, перезапускаете систему с него и загружаете программу сетевой поддержки, что приводит к следующим результатам:

- Прекращается износ дисководов и порча дискет. Исчезают сбои при чтении и записи, которые так часты при работе на НГМД.
- Загрузка РМУ идет в несколько раз быстрее, чем с НГМД, повышается скорость работы в сети и уменьшается количество конфликтов из-за одновременных обращений к РМП с нескольких РМУ.
- При установке ЭД на каждое РМУ ученики

получают быстрый и надежный персональный компьютер, не зависящий от сбоев и зависаний сети.

- После выключения питания несохраненные на НГМД данные исчезают, что избавляет преподавателя от необходимости ежедневно «расчищать мусор», накопленный учениками.
- Существуют модели ЭД, позволяющие после сбоя ЭВМ или зависания программы перезапустить операционную систему без использования НГМД (что особенно важно при установке их на РМУ).

### Накопитель на жестком диске (НЖД)

В настоящее время выпускаются несколько отечественных моделей НЖД объемом 5, 10 или 20 Мб. НЖД представляет собой жесткий диск в корпусе с блоком питания, подсоединенный кабелем к контроллеру, который вставляется в свободный слот УКНЦ. Использование НЖД дает вам следующие возможности:

- Появляется возможность одновременной работы на разных РМУ с существенно разными программными пакетами (которые просто не могли уместиться на одной дискете).
- Скорость и надежность работы класса существенно выше, чем при работе с НГМД (хотя и ниже скорости с ЭД).

### Модем

В какой-то момент перед каждым преподавателем встает проблема получения и обмена информацией. Одним из мощных средств является модем — устройство для связи компьютеров по телефону. В настоящее время выпускаются модемы для УКНЦ со скоростью передачи от 600 до 1200 Бод, что вполне достаточно для обмена информацией. На основе использования модемов организуются широкие сети обмена информацией по всей стране.

### Сетевой адаптер высокой производительности (САВП)

Поставляемое с классами УКНЦ сетевое оборудование отличается крайней ненадежностью и невысокой скоростью работы. Для устранения этих недостатков, а также для обес-

печения бесконфликтной работы учеников с накопителем на жестком диске можно применить САВП. САВП устанавливаются в свободный слот каждого рабочего места и соединяются одной парой проводов (без использования реле). На одном из рабочих мест (например, РМП) устанавливается САВП, совмещенный с контроллером гибкого диска. Под сервер сети выделяется отдельный компьютер с накопителем на жестком диске. Каждой станции в сети доступно общее «системное устройство» по чтению (на системном устройстве выделяются несколько файлов, доступных и по записи, например, LD.SYS, SWAP.SYS, SL.SYS и STARTS.COM, при этом физически эти файлы свои для каждого рабочего места). Кроме того, каждой станции доступно свое «рабочее устройство» по чтению и записи. И, наконец, с любой станции можно получить доступ по чтению к «рабочим устройствам» других станций. Загрузка любой станции производится выбором пункта «диск» из системного меню УКНЦ и может быть независимо произведена в любой момент времени.

Перечислим преимущества высокопроизводительной сети:

- Производительность САВП составляет около 250 КБод, что в пять раз выше производительности стандартного сетевого адаптера. Это ускоряет загрузку класса, реально позволяет использовать на каждом РМУ полноценную операционную систему, а не ее эмулятор, и программы, активно работающие с диском (компиляторы, графические редакторы и др.).
- Работать с файлами, не опасаясь конфликтов при их создании и удалении.

- Невозможно повреждение системных и пользовательских программ при несанкционированном доступе.
- Для установки сети в классе не требуется специальная квалификация. Можно использовать провода, проложенные для стандартной сети.

### Расширитель слотов (РС)

При таком обилии периферийных устройств возникает проблема их совместного использования — ведь в УКНЦ всего два слота (раз'ема для установки периферийных устройств). Решить данную проблему позволяет устройство «Расширитель слотов», которое представляет собой плату с микросхемами усиления шины компьютера и тремя раз'емами на задней панели. РС устанавливается в дальний слот и позволяет ставить в свои раз'емы до трех периферийных устройств одновременно.

### Конфигурация класса

На основе приведенного материала можно составить несколько конфигураций класса УКНЦ, руководствуясь, с одной стороны потребностями школы а, с другой — материальными возможностями.

Разумеется, выбор остается за преподавателями и методистами. Многие, начав с минимального варианта и убедившись в достоинствах электронных дисков, затем оснащают ими все РМУ, что не мешает в дальнейшем установить на РМП и жесткий диск. Автор надеется, что данная статья поможет читателю принять верное решение.



## АО «ЭЛЕКТРОННЫЕ РАБОТЫ»

предлагает

периферийные устройства для УКНЦ:

- |  |   |
|--|---|
| <input type="checkbox"/> Электронный диск 512 Кб и 1 Мб    | <input type="checkbox"/> Контроллер жесткого диска                  |
| <input type="checkbox"/> Накопитель на жестком диске 20 Мб | <input type="checkbox"/> Сетевой адаптер высокой производительности |
| <input type="checkbox"/> Модем 1200 Бод (F-bit Ltd.)       | <input type="checkbox"/> Расширитель слотов                         |

тел. (095) 264-73-40, 939-32-68



Ю.В. Котов, г. Москва

## ЭКОНОМНОЕ ПРОГРАММИРОВАНИЕ ДЛЯ БК-0010.01 И БК-0011М

Известен «главный» недостаток ПЭВМ БК-0010.01 — малый объем оперативной памяти, который значительно усугубляется недостатками системы обработки зашитого в них Бейсика с объемистым «объектным кодом». Существенная часть ОЗУ тратится также на таблицу адресов строк. Таблица поневоле получается длинной из-за недостатка данной реализации Бейсика, не позволяющей записывать в одной строке несколько операторов. В итоге в БК-0010.01 может работать примерно вдвое более короткая программа, нежели это было бы возможно при наличии интерпретатора (можно сравнить с допустимым объемом программ на Фокале).

Попытки работать с загружаемым с магнитной ленты или даже диска Бейсиком либо другим языком не дают существенного выигрыша (чаще приводят к проигрышу). Происходит это из-за необходимости занимать ОЗУ интерпретатором или компилятором, а также подпрограммами операций с нормализованными числами и вычисления функций. Надо сказать, что «штатный» режим работы с Бейсиком, когда в ПЗУ находится система обработки с подпрограммами арифметики и вычисления функций, а в ОЗУ одновременно исходный текст, объектный код (условное подобие машинной программы), таблицы и данные удобен лишь для отладки про-

грамм и в ситуациях, когда программа должна сработать всего один или несколько раз. Если же готовится программа для долгой жизни и многократного использования, этот режим неэффективен. После отладки программы становятся не нужны ни повторные трансляции, ни загрузка памяти лишними теперь таблицами и исходным текстом. Любопытно, что в имеющейся реализации Бейсика почти все сделано для того, чтобы объектный код мог работать без таблиц и исходного текста. Но «почти» не есть «да». И хотя иногда можно искусственно заставить работать объектный код самостоятельно — память уже распределена, сам код переместим и практического выигрыша просто так не получишь.

Конечно, разработчики давно должны были учесть критику и рациональные предложения и обновить систему, введя возможность записи в одной строке нескольких операторов, добавив команду «MERGE», исправив ошибку в обработке многомерных массивов и другие ошибки. Следовало бы также предусмотреть режимы записи на носитель и загрузки объектных кодов с настройкой адресов и интерпретацией этих кодов без исходных текстов. Но где ныне эти разработчики?

Надо также отметить, что объектный код неэкономичен, объемист, он имитиру-

ет нуль-одноадресную «стековую» квази-машину, причем и код операции, и адрес занимают... 16 бит (где это видно?!). В итоге на одну операцию  $A\%=A\%+1\%$  без учета исходного текста уходит 20 байт.

Понятно, что автоматически генерировать эффективный промежуточный код — задача сложная, и тем не менее из-за отсутствия этой возможности практическая ценность машины существенно снижена.

А что вместо Бейсика? Работа на типовом ассемблере по замыслу и сложности не сравнима с работой на языке высокого уровня, да еще для многих приложений — математических, геометрических, физических и т. п. — неэффективна из-за примитивности процессора, который не имеет даже команд умножения-деления целых чисел, не говоря уже о нормализованных. Целесообразно было бы разработать специализированный «макроассемблер», совместимый с системой обработки Бейсика и использующий ее ресурсы для реализации нормализованной арифметики и вычисления функций, но, по имеющимся сведениям, такого нет.

Практическая работа над программами увеличенной сложности показала, что переход на программирование в машинных кодах с использованием ресурсов обрабатывающей системы Бейсика (фактически программирование в операциях объектного кода) позволяет разместить в ОЗУ вдвое более сложные (длинные) программы, чем на Бейсике. Дополнительная экономия еще примерно в полтора раза достигается для длинных программ реализацией подобия двухадресных команд «квази-машины» на уровне объектного кода.

Подобные машинные (или квазимашинные) программы могут работать совместно с Бейсиком как подпрограммы, вызываемые оператором `USR`. Разработана методика обращения к этим подпрограммам с передачей им фактических параметров списком через особым образом организуемые функции пользователя типа `FN`.

Используя специальную кодировку с дополнительными таблицами, при некоторых ограничениях и очень незначительном снижении быстродействия можно еще почти вдвое снизить затраты ОЗУ; в результате по сравнению со штатным программированием на Бейсике ожидается примерно пятикратный рост объема вмещаемых в ОЗУ программ.

Конечно непосредственное (ручное) программирование в условных псевдокодах и машинных командах — дело очень кропотливое и рядовому пользователю (начинающему программисту) малодоступное. Нужны специализированные трансляторы. Первоначально они могут быть функционально ограниченными, например, обеспечивать обработку только арифметических операций с целыми и нормализованными числами, вычисление функций, а также минимум операций управления (переходы, циклы, вызовы подпрограмм). Исходный же текст на особом языке программирования может где-то напоминать «автокод», не обязательно поднимаясь до уровня, подобного алгебраической записи формул.

*Редакция приглашает заинтересованных программистов заняться решением этой проблемы. Описания лучших разработок будут опубликованы.*

И. Л. Андронов, г. Одесса

## ОДА ФОРТУ

Впервые реализация Форта для БК-0010 появилась в 1986 г. трудами ленинградцев М. Ефимова и А. Цаплева. По тем временам для пользователей БК-0010 это был настоящий прорыв замкнутого круга под названием «Фокал». Однако первые радостные голоса быстро утихли, когда выяснилось, что Форт на БК-0010 оставляет пользователю всего лишь 3–3.5 Кб памяти, чего может хватить разве только для учебно-познавательных целей. Причина этого – ставший притчей во языцех мизерный ресурс ОЗУ БК-0010. Поэтому Форт для БК-0010 версии 1986 года не получил распространения.

Все другие причины непопулярности Форта, в том числе и его необычность, я считаю необоснованными. Утверждать обратное – значит подвергать сомнению «расторопность» пользователей БК-0010, которые одолели и не такие «необычности».

Еще чаще говорят о сложности освоения Форта, но об этом почти всегда можно услышать только от тех, кто и не пытался серьезно им овладеть. Вообще же, с учетом появившейся литературы о Форте и на основании личного опыта, смею утверждать, что Форт в освоении легче, чем Бейсик.

Что же касается «необычности», то этот термин говорит сам за себя: чего не знаешь, то и необычно. Если же разобраться по существу, то главная необычность Форта заключена в стереотипе мышления программирующего на императивных языках. По мере «вхождения» в Форт мышление быстро адаптируется.

Какими же «плюсами» отличается Форт от остальных языков?

- Широта и глубина мышления программирующего на Форте не ограничена жесткими формами «Кирпичики», с помощью которых строится программа, весьма разнообразны как по количеству, так и по диапазону сложности. Более того, лексикон этих «кирпичиков» зависит исключительно от программиста и им же обогащается, так как Форт по своей природе расширяемый язык.
- После создания каждого нового определения программист легко «прощается» с деталя-

ми, освобождая внимание для перехода на более высокий структурный уровень.

- Форт одинаково пригоден как для системного, так и для прикладного программирования.
- Форт для БК-0011 делает компьютер вполне «серьезной» и «самостоятельной» машиной. Не секрет, что работа на БК-0011 до сих пор шла в основном по пути не всегда эффективного использования богатого программного обеспечения от БК-0010. А вот поместив Форт на БК-0011, пользователь получает в свои руки все ресурсы машины — и «HARDWARE» и «SOFTWARE».
- Форт устроен так, что легко приспосабливается для своих нужд «чужое» программное обеспечение. Так, в описываемой версии (VER 2.0 JAN 92), имеется слово «BLOAD», позволяющее подгружать к Форту с магнитофона «чужие» программы в позиционно независимом коде (файлы, созданные «Меломаном», дезассемблер DISASM М. Цыпина и т.д.), а пакет арифметики с плавающей запятой напрямую использует подпрограммы из ПЗУ Бейсика.
- Пакет Форт-ассемблер создает интерактивную среду, более удобную, нежели при работе с MIRAGE, ОТЛ12 или, тем более, при работе в системах «MICRO». При этом Форт-ассемблер позволяет применять как локальные (в пределах одного определения), так и глобальные метки, где последними служат обычные Форт-слова (длиной до 31 символа).
- Форт имеет средства чтения EDASP-текстов.
- В Форте всегда присутствует свой редактор, по мощности сопоставимый с редактором EDASP.
- Если компилятор Бейсика долгое время был «закрытой зоной» для пользователей и лишь недавно «вскрыт» расторопными умельцами, то Форт сразу объявляет себя открытым для всех, даже для слабо подготовленных пользователей. Форт «насквозь просвечивается» такими мощными сред-

вами самодокументирования, как *REFOPH, DECOMPILER, DUMP.*

- Сколько «слез пролито» программирующими на Бейсике БК-0010.01 из-за отсутствия в системе того или иного оператора (*MERGE, SPRITE, MAT* и др.)! А суть программирования на Форте как раз и заключается в создании (определении) новых нужных в данный момент слов-операторов, так что в некотором смысле можно говорить о неисчерпаемости Форта.

А чем еще необычен Форт? Вот несколько примеров:

- В Форте возможно целочисленное умножение таких величин: *ОСТ 377 400 \* (В Бейсике при этом произошло бы переполнение).*
- Форт позволяет напечатать результат в нужном вам формате:

1) *ОСТ 377 400 \* .*  
*ОСТ 377 400 \* U.*

Ответы соответственно будут -400 и 177400.

2) Поместим два кода символов в одно слово:

*ОСТ ASCII Щ 400 \* ASCII Я + U. <БК>*  
*176761*

Теперь введите такую строку:  
*176761 SP @ 2 TYPE DROP*  
*ЯЩ*

('DROP' здесь нужно лишь для того, чтобы не «засорять» стек данных, так как 'TYPE' не предназначено для печати со стека и поэтому данные (176761) остаются на стеке после 'TYPE').

Необычностей в Форте много, и среди них есть такие экзотические, что не отпугивают, а скорее завораживают своими чудесными и ценными свойствами. Что вы, например, скажете о возможности в Форте присваивать переменным не только величины, но и действия, т. е. переменная в некотором роде может стать оператором,

Пример 1.

```
: TABLE ."СТОЛ";
: СТОЛ ."TABLE";
```

и т. д. — вот у вас уже и готов англо-русский и русско-английский словарь.

и таким, каким вам заблаговолится!? Вот пример программы:

```
ОКТ VAR A VAR B VAR C ' NOOP C !
: ANSWER A @ B @ C @ EXECUTE ;
```

```
: SUM ['] + C ! ANSWER ;
: DIFFER ['] - C ! ANSWER ;
: MUL ['] * C ! ANSWER ;
: DIV ['] / C ! ANSWER ;
```

```
:QUADR_SUM SUM DUP * ;
:QUADP_DIFFER DIFFER DUP * ;
```

```
10 A ! 4 B !
```

(В зависимости от вводимого вами слова переменной 'C' присваивается соответствующий оператор, который исполняется словом 'EXECUTE').

Поговорим теперь о сложности освоения Форта. Да, Форт действительно достаточно сложен. Когда я впервые прочитал, что Форт изобретен Чарльзом Муром, то термин «изобретен» представился мне несколько высокомерным. Теперь же, когда Форт стал моим «вероисповеданием», я преклоняюсь перед гением создателя. Уровень погружения в Форт безмерен. Занимаясь Фортом уже более 3 лет, я все еще ощущаю себя юнцом, постоянно встречаясь все с новыми и новыми открытиями, а количество непонятных вещей, пожалуй, даже растет.

На следующих примерах я хочу показать, что самый консервативный учитель математики, усвоивший Бейсик «на всю оставшуюся жизнь», или самый нерадивый ученик 4–5 класса, на примере четырех-пяти игр научившийся осмысленно стучать по клавиатуре, за 15–20 минут «инструктажа» или даже самостоятельно вполне могут усвоить на приведенных примерах простейший синтаксис и метод программирования на Форте (заодно попытайтесь-ка решить эти же задачи средствами Бейсика, и сравните результат).

Форт позволяет «открытым текстом», в том числе и на русском языке, записывать (а в этом языке программирования правильно записать – значит уже решить) условия задачи, оформляя их как «определения через двоеточие». Запустив на исполнение последнее слово '??', вы получите на своем экране строку с ответом:

```

: частей_алюминия      19 ;
: частей_магния        2;
: вес_разности         34;
: разность_частей     частей_алюминия частей_магния - ;
: вес_одной_части     вес_разности разность_частей / ;
: вес_алюминия        частей_алюминия вес_одной_части * ;
: вес_магния          частей_магния вес_одной_части * ;
: ?? ."
ответ: вес сплава = "
вес_алюминия вес_магния + . ." кг" ;
    
```

```

??
ответ: вес сплава = 42 кг
    
```

Обратите внимание: мы не использовали ни одной переменной. В решении этой задачи Форт показал нам еще одну свою грань: Форт – это язык спецификаций. (В данном случае для решения задачи мы обошлись целочисленной арифметикой. Синтаксис при использовании арифметики с плавающей запятой тот же).

И в заключение хочу отметить следующее. Еще одно «препятствие», по мнению многих оппонентов (постфиксная запись формул) также является, на мой взгляд,

надуманным. Во-первых Форт – стековая система, поэтому когда вы научитесь думать по-фортовски (а для этого надо немедленно начать программировать на Форте), постфиксная запись представится вам вполне естественной (кроме оправданных исключений, этот принцип прослеживается во всем словаре Форта). А во-вторых, весьма просто переопределить способ ввода на любой (но нужно ли это?). Вот пример для инфиксной записи:

```
:INFIX BL WORD NUMBER DROP ;
```

Теперь можно переопределить любые операторы:

```

: + INFIX + ;
: - INFIX - ;
: AND INFIX AND ;
    
```

и т.д.

И теперь можно вводить строки формул «обычным» способом (без скобок!).

Вообще говоря, возможности Форта, как тема для разговора, не имеют види-

мых границ, и вы наверняка откроете для себя много нового в чудесном мире Форта.

**КОЛЛЕДЖ**

### **Фирма "КОЛЛЕДЖ":**

*Мы помогаем Вам в решении проблем компьютеризации учебных заведений*



### *К Вашим услугам:*

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

*Доставка техники и мебели осуществляется по Москве и Московской области.*

### *Последние программные и методические разработки:*

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

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

Телефоны: 262-62-65

267-70-58

Телефакс: 265-62-65

*Каталог программного обеспечения высылается бесплатно*

# ЗАРУБЕЖНЫЙ ОПЫТ

Стефан Керр,

*профессор педагогического колледжа Вашингтонского университета (США)*

## НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И РЕФОРМА ШКОЛЫ

Когда обсуждают «педагогические технологии» речь обычно идет о «технических средствах» и «процессах».

«Аудио-визуальные средства», «обучающие машины», «автоматизированные обучающие системы» — вот далеко не полный перечень широко известных названий и образов, которые возникают у большинства учителей, когда речь заходит об использовании информационных технологий в образовании.

Последние два десятилетия в среде специалистов представление о предметной области «педагогической технологии» расширялось и уточнялось. В результате, сегодня внимание концентрируется не на технических средствах, а на способах их использования. Типичный пример — процедуры разработки учебных материалов и построения учебного процесса, известные как «IDD» (Instructional Design and Development). Процедурный подход выявил новое сущностное основание для определения «педагогической технологии». (Подробно об этом см. в [1], [2]).

В статье я пытаюсь показать, что и этот взгляд на роль технологий в образовании слишком узок. Он не позволяет увидеть изменений, которые технологии фактически привносят в сегодняшнюю общеобразовательную школу. Я хочу показать, что «педагогическая технология» связана в первую очередь не с «техническими средствами» и не с «процессами», а с изменением фундаментальных основ, которые собственно и порождают в обществе потребность в формальном образовании.

Подтверждением этому служит, во-первых, история развития технологии. Имеется множество примеров, когда развитие технологии приводило к таким изменениям в нашей жизни, которые далеко выходили за рамки проблем, на

решение которых эти технологии были первоначально направлены.

Кроме того, эффективное использование новых информационных технологий ведет к глубинным изменениям в содержании, организационных формах и методах работы школы.

Распространение технических средств и процессов порождает такие последствия, которые выходят за рамки ожиданий, инициировавших их разработку.

### Технология, как катализатор изменений

Книгопечатание и рождение современности. Обычно считают, что книгопечатание — это изобретение, направленное на снижение стоимости, увеличение точности и повышение эффективности производства отдельных копий письменных материалов. Это конечно верно. Однако изобретение книгопечатания привело к таким последствиям, о которых и не помышляли Гутенберг, Альдус и другие первопечатники. Правительства, решая общенациональные задачи, стали собирать и применять для своей работы огромные объемы информации, что было бы невозможно без использования типографий. Для огромного большинства людей стала доступна грамотность, что было бы невозможно без резкого удешевления массово доступных текстов. Книгопечатание изменило и образование. Абсолютный контроль учителя над знаниями учеников стал невозможен. Учитель перестал быть единственным источником информации для школьников. Зависимость человека от собственной памяти уменьшилась, а целый класс монахов и писцов, которые переписывали и интерпретировали тексты, потерял влияние в обществе.

Книгопечатание — пример технологической инновации, оказавшей существенное влияние на развитие общества. Вызванные им изменения происходили медленно и были по большей части неожиданны для большинства населения. Однако эти изменения оказались настолько мощными, что позволили современному обществу сменить феодализм, привели к новому общественному и политическому устройству, к новым ментальным структурам, которыми мы пользуемся сегодня. (О развитии и последствиях книгопечатания подробнее см. [3, 4, 5]).

Телефон и новое представление о рабочем месте. Телефонная связь — сравнительно недавнее техническое новшество. Она является еще более ярким примером, возможно потому, что соответствующие изменения нам очень хорошо знакомы.

Если вы думаете, что телефон был изобретен для того, чтобы облегчить и ускорить общение между людьми, вы ошибаетесь. Сегодня это выглядит аксиомой, но мы легко забываем, что изобретатель телефона Александр Белл вовсе не думал о тех услугах связи, которыми мы пользуемся сегодня. Его идея состояла в том, чтобы обеспечить постоянную связь одного абонента с другим заранее определенным абонентом: управляющий прокладывал телефонную линию от дома до фабрики, а если ему нужно разговаривать с центральным агентством компании, он прокладывал другую линию. Идея телефонной сети родилась значительно позже.

Многие энтузиасты телефонной связи на ранних этапах ее развития прогнозировали использование вновь появившейся технологии для выполнения и других функций (распространение новостей, передача музыкальных программ или публичных лекций). Они ошибались: это стало возможным лишь с появлением радио. Однако телефон сделал другое: он изменил наше представление о том, каким может и должно быть современное рабочее место.

До изобретения телефона работники (прежде всего работники информационной сферы тех времен: газет, банков и т. п.) стремились жить и работать как можно ближе друг к другу: пересылать сообщения на расстояние было долго, трудно и дорого. Расчеты, выполненные к началу распространения телефона, показывали, что без телефона конторы в двадцатом столетии должны необычайно вырасти, чтобы вместить всех бесконечно спящих по лифтам и лестницам курьеров, переносящих почту между помещениями. Телефонная связь выработала новые модели де-

лового общения, ориентированные на личные связи, менее замкнутые и формальные [6].

Телевидение. Еще одним примером может служить телевидение. Мы обычно представляем его как средство распространения новостей и развлечений, расширяющее традиционные возможности радио и кино. Телевидение оказалось необычайно успешным нововведением: сегодня телевизоры установлены более чем в 90% домов в США. Это произошло всего за несколько десятилетий после появления коммерческого телевидения.

Однако существуют и другие последствия распространения телевидения, которые связаны с его воздействием на зрителей. Маршал Маклан [7] утверждает, что под воздействием современных коммуникационных средств (и прежде всего телевидения) большинство жителей утратили возможность неторопливого «линейного» восприятия информации. Сегодня господствует телеграфный афористичный стиль, формируемый телевидением. Хотя эти изменения накапливаются медленно, Мероуиц [8] предполагает, что телевидение приводит к новому восприятию мира, которое разрушит традиционные социальные структуры и умственные барьеры, поместит полтиков в другое, отличное от традиционного общественное пространство, изменяет взаимоотношения между родителями и детьми, между мужчинами и женщинами.

По крайней мере в одной области — политической жизни страны — телевидение произвело переворот (причем негативный). Телевидение изменило содержание и глубину обсуждения как самих кандидатов, так и их политических программ. В дотелевизионную эру обсуждение концентрировалось вокруг политических взглядов и жизненного опыта кандидатов, вокруг их политических программ. Телевидение сместило обсуждение с содержательных вопросов избирательной компании к результатам опросов общественного мнения с бесконечными комментариями на тему «Кто первый?». Вместо характерного в прошлом обсуждения долговременных национальных целей, телевидение касается только таких тем, которые можно уложить в семисекундную реплику, и которые являются зрелищными.

Сборочный конвейер: технологический процесс, который изменил мир. Все рассмотренные примеры так или иначе касались технических средств. Ведет ли к таким же последствиям появление новых технологических процессов? Хорошим примером может служить конвейерное производство — центральное событие промышленной революции девятнадцатого



столетия. Введение конвейерного производства было призвано увеличить объем выпуска продукции, снизить ее стоимость, повысить качество и, в конечном итоге, увеличить получаемые доходы. Главной особенностью любого конвейера, будь то производство ружей, мясных консервов или автомобилей, остается рационализация и сверхспециализация труда.

Повышая прибыльность предприятия, введение конвейерного производства порождало и свои проблемы. Отчуждение рабочего фабрики от своего труда стало очевидным уже в середине девятнадцатого века. Пороки, порождаемые этой системой, стали основным фактором возникновения социалистических движений и воинствующих профсоюзов. В то же время, массовое производство позволило большинству населения достичь сравнительно высокого уровня жизни. Общедоступность широкого ассортимента товаров и услуг для подавляющего большинства населения позволила, по мнению некоторых [9], построить более демократическое общество.

**Взгляд через кривое зеркало.** Когда происходит большой технологический сдвиг, представление о роли и месте этой технологии остается крайне ограниченным. Как заметил М. Маклухан [7], мы видим настоящее через кривое зеркало. Непосредственные «цели» технологии, которые определяют в терминах соответствующей специальной области, как правило, мало связаны с тем, что лежит за ее рамками.

Во времена Гутенберга монахи не сомневались, что будут по-прежнему переписывать книги уже потому, что «занятие это нравственно само по себе» [10].

Когда Уильяму Ортону, президенту Западной Объединенной Телеграфной Компании предложили приобрести патент на телефон он удивился: «Зачем нам нужна эта электронная игрушка?» [11].

Лишь немногие из тех, кто способствовал широкому распространению конвейерного производства, осознавали важность привносимых им изменений. Но никто не предполагал, что конвейерное производство несет в себе угрозу изменения сложившихся общественным отношениям [12, 13].

По мере того, как основанная на использовании компьютеров новая информационная технология осваивается и становится частью повседневной жизни, все более ощущается ее воздействие на происходящие в обществе изменения. В США сейчас идут противоречивые, неявные споры о действительном значении технологии: одни видят в ней источник всех зол, другие — превозносят ее вклад в развитие общества [14, 15, 16, 17]. Вне зависимости от итогов этой дис-

куссии можно отметить, что технология занимает значительное место в умах американцев. Поэтому так понятен энтузиазм многих педагогов и их видение технологии, как средства решения вечных проблем учения и обучения.

Влиянию информационной технологии на различные области нашей жизни посвящено множество работ. Однако ее влияние на процессы, происходящие в сфере образования, все еще игнорируются.

## Новые информационные технологии в образовании

Изложенное выше позволяет начать описание социальных последствий, к которым могут привести технологические нововведения в образовании. Однако это не входит в наши намерения. Мы рассмотрим лишь несколько областей внутри самой образовательной системы, где распространение технологии уже привело к существенным изменениям.

Мы будем говорить исключительно о новых информационных технологиях (НИТ). И это не потому, что другие технические средства или процессы менее важны. (Существуют убедительные аргументы в пользу того, что в центре внимания педагогической технологии стоит учитель [18]). Дело в том, что проникновение компьютеров и информационных технологий в образование значительно способствует тем изменениям, которые происходят здесь сегодня.

Из областей, где эти изменения наиболее значительны, можно выделить:

- организацию жизни в классе,
- роль педагога,
- способы учебной работы учащихся,
- взаимосвязь школы с социально-экономическими процессами, происходящими как на местном, так и на общегосударственном уровнях.

**Организация жизни в классе: от унификации к разнообразию.** Проникновение ЭВМ в образование уже привело к существенным сдвигам в организации учебной работы в классе. До появления ЭВМ даже те учителя, которые действительно стремились дать своим воспитанникам различные виды учебного материала и обеспечить многообразие подходов к его изучению, с трудом могли это себе позволить. Обстоятельства вынуждали их делать то, что можно, вместо того, что нужно. У нас нет практически приемлемых способов (методик) для организации индивидуальной работы школьников в группе, состоящей из 25—30 человек. Преподаватели педаго-

гических колледжей постоянно испытывали трудности, пытаясь побудить учителей, проходящих курсы переподготовки, подумать о возможных формах индивидуализации работы с учащимися [19].

Использование ЭВМ позволяет (хотя это и требует большой подготовительной работы) существенно высвободить время учителей, что казалось невыполнимым всего несколько лет назад. Но это возможно только тогда, когда выполняются следующие условия:

- учителя находят достаточно времени для ознакомления со средствами ИИТ и понимают как можно было бы их использовать в учебном процессе, а далее в течение нескольких лет на практике осваивают представленные возможности;
- администрация школы и школьного округа оказывает действенную поддержку проводимым начинаниям;
- в школе имеется достаточное количество пригодных для работы компьютеров приемлемого качества (не обязательно по компьютеру на каждого ученика, но как минимум по несколько компьютеров в каждом классе);
- внутришкольная организация позволяет каждому педагогу выработать свой собственный стиль работы, ощутить себя создателем лично значимых оригинальных способов учебной работы (что и является отличительной чертой использования средств ИИТ в учебном процессе).

Многочисленные примеры, подтверждающие сказанное, можно найти в работах [20–23].

Сегодня все мы являемся свидетелями начинающейся радикальной перестройки учебной работы в классе, и это не просто очередная смена методов обучения: сочетание новых организационных форм и методов обучения ведет к самым значительным за прошедшие сто лет изменениям педагогической практики. Отличительными чертами новой педагогики являются:

- увеличение удельного веса проектной работы учащихся и возрастание объема работы, выполняемой в группах;
- снижение роли учителя как «источника знаний» и возрастание его роли как «воспитателя и наставника»;
- возрастающая готовность учащихся нести ответственность за результаты своего обучения.

Конечно, все это не дается даром. Многие педагоги уже убедились, что работать с самоуправляемыми группами школьников, которые имеют самостоятельный доступ к источникам информации и сами оценивают результативность своей работы, совсем непросто. Введение ЭВМ

и соответствующих организационных форм обучения неизбежно ведет к кардинальным изменениям взаимоотношений, которые традиционно складывались в школьном коллективе.

**Педагоги: превращение из астрологов – в потребителей и производителей информации.** Другой важнейший результат использования ИИТ состоит в том, что педагоги (учителя, администраторы и другие специалисты) изменяют способы сбора и использования необходимой им информации. Педагогику давно и справедливо критикуют за то, что она является «неточной» дисциплиной. Однако превратить ее в «точную» непросто так, как:

- статистическое описание результатов оценки успеваемости школьников довольно трудно правильно интерпретировать, и даже обучение учителей использованию этой информации – непростая задача;
- компилятивная информация по проблемам педагогики сужена до минимума, но даже эти данные редко попадают туда, где их можно использовать по назначению;
- педагоги, как правило, не учат тому, как отбирать и накапливать полезную информацию, использовать ее в своей повседневной работе.

Доступность компьютеров и компьютерных сетей кардинально меняет практику работы с педагогической информацией. Использование широко распространенных пакетов обработки и визуализации статистической информации (электронные таблицы, базы данных) позволяет существенно ускорить обучение учителей методам статистики и процедурам, принятым в исследовательской работе. Компьютерные сети упрощают распространение информации между учителями различных школ, районов и даже стран. Вместе с возрастающим вниманием к совместным исследовательским проектам, растет и готовность педагогов обмениваться имеющейся у них информацией.

Политический климат в учебных заведениях во многом определяется тем, как в них распространяется (или не распространяется) информация. Сегодня, по мере того, как педагогическая информация становится доступной и понятной все большему числу рядовых педагогов, принятие ответственных решений переносится из кабинетов районной и школьной администрации в педагогические и ученические коллективы.

**Учащиеся: противоречие между активностью и пассивностью.** Использование технологии предоставляет учащимся широчайший

диапазон возможностей для учебной работы в школе.

С одной стороны, это хорошо. Происходящие изменения (повышение гибкости и разнообразия учебной среды, увеличение возможностей для индивидуальной работы и большее внимание со стороны педагогов) создают благоприятные условия для интенсификации обучения. Появление компьютерных коммуникаций порождает весьма привлекательную для школьников возможность участвовать в совместных исследовательских проектах с партнерами в других странах.

С другой стороны, распространение технологии несет с собой и отрицательные последствия. Фрагментация внимания учащихся при просмотре телевизионных программ, где длительность кадра ограничивается 15-секундными интервалами, превратное представление о телевидении, как о средстве развлечения, порождает проблемы, которые еще не до конца осознали педагоги [24 – 26]. Мы все говорим, что школа – это место, где у детей воспитывается активное мышление. Тем более удивительно, что некоторые учителя и родители ратуют за введение в школах технических средств, которые способствуют увеличению пассивности. Если внедрение стандартных видеоклипов в школьные программы будет продолжаться, возросшая социальная пассивность широких масс населения приведет к далеко идущим отрицательным последствиям для всех сторон жизни американцев.

**Школа и ее окружение: от «самодостаточности» к «открытости».** Распространение НИТ в образовании связано не столько с самими школами, сколько с изменением их статуса в окружающей общине. В частности, это особенно заметно в отношениях между школами и предприятиями, которые принимают на работу выпускников. Хорошо известно, что многие годы предприниматели весьма скептически относились к школам. Это не удивительно: цели, да и весь склад жизни этих учреждений были существенно различны. Интересно наблюдать, как распространение НИТ стирает эти различия.

С точки зрения предпринимателей интенсивность труда участников учебно-воспитательного процесса в школе крайне недостаточна. Для бизнеса школа – это предприятие где расточительно расходуются огромные человеческие ресурсы. Энтузиазм многих компаний, поддержавших введение НИТ в образовании, отражает тенденцию производителей менять старую технологию (учителей) на новую (компьютеры).

По мере развития кооперации между школами и производственными предприятиями постепенно вырисовались реальные проблемы и ограничения, с которыми сталкиваются школы. (Стремление к сотрудничеству было вызвано потребностью предприятий выжить в новом технологизированном обществе и удовлетворить свои потребности в квалифицированной рабочей силе, которые не удовлетворяет существующая школа.) С другой стороны, классы со средствами НИТ и обученными педагогами постепенно становятся похожи на «современное рабочее место».

Конечно, промышленные предприятия – это не все общество. Однако озабоченность американцев конкурентоспособностью выпускников американских школ и американской промышленности на мировом рынке возрастает. По мере того, как НИТ улучшает отношения между школами и производственными предприятиями, позиция школ в обществе усиливается.

### Заключение

Говоря о внедрении НИТ в школах, мы выяснили, в чем состоят задачи школы, кто и как их решает, как это затрагивает интересы общества, в котором живет и которому служит школа.

Непосредственное влияние НИТ на результаты обучения в долгосрочной перспективе менее важно, чем влияние компьютеров на изменение представления педагогов о том:

- как должна выглядеть классная комната,
- как осознают себя участники учебно-воспитательного процесса,
- как меняется их взаимодействие в условиях использования НИТ.

Выработка у школьников тех или иных навыков, усвоение ими отдельных фактов с использованием компьютеров в конечном итоге не так значимы, как влияние НИТ на превращение школьников в активных или пассивных участников учебного процесса, активных или пассивных участников жизни демократического общества. НИТ меняет практику распространения информации в школе, что, в свою очередь, приводит к «перераспределению власти» в школе и принципиально меняет ее работу. Наконец, распространение НИТ ведет к сближению школы и ее ближайшего окружения.

Все эти процессы уже начались, хотя их результат еще не вполне ясен. Новые исследования могут указать на другие, более существенные факторы, чем те, о которых говорится в этой статье. Однако ясно, что социальная составляющая вклада технических устройств и

процессов, привносимых НИТ в школу, более существенна, чем те технические и технологические проблемы, для решения которых они привлекаются. Нам следует обращать больше внимания на социальные изменения (часто загадочные), которые мы можем различать, и меньше обращать внимания на знакомые (хотя и второстепенные) проблемы педагогической эффективности средств НИТ, которые мы зачастую видим в кривом зеркале.

## Литература

- [1]. The Definition of Educational Technology. Association for Educational Communication and Technology, Washington, DC, 1977.
- [2]. *Heinich, R.* Technology and the Management of Instruction, Monograph No.4, Association for Educational Communication and Technology, Washington, DC, 1971.
- [3]. *Burstin, D.J.* The Discoverers. Random House, N.Y., 1983.
- [4]. *Eisestein, E.* The Printing Press as an Agent of Change. Cambridge, N.Y., 1979.
- [5]. *Luke, C.* Pedagogy, Printing and Protestantism: The Discourse on Childhood, SUNY Press, Albany, N.Y., 1989.
- [6]. *de Sola Pool, I.* (ed.) The social impact of telephone. MIT Press, Cambridge, 1977.
- [7]. *McLuhan, H. & Fiore, Q.* The Medium is the Massage. Bantam N.Y., 1967.
- [8]. *Meyrowitz, J.* No sense of Place: The Impact of electronic media on social behavior. Oxford, N.Y., 1985.
- [9]. *Boorstin, D.* The Americans: The Democratic Experience. Random House, N.Y., 1973.
- [10]. *Brann, N.* A Monastic Dilemma Posed by the Invention of printing: The Context of De Laude Scriptorum Manualium by Abbot Johan Thritheimus (1462—1516). Visible Language, Vol. 13(2), 1979, pp. 150—167.
- [11]. *Aronson, S.* Bell's electrical toy: What's the use? The sociology of early telephone usage. In: The social impact of the telephone, MIT Press, Cambridge, MA 1977, pp. 15—39.
- [12]. *Jannings, H.* Pandemonium: The Coming of the Machine as Seen by Contemporary Observers, Free Press, N.Y., 1985.
- [13]. *Marvin, C.* When Old Technologies are New: Thinking about electric communication in the late nineteenth century. Oxford, N.Y., 1988.
- [14]. *Florman, S.* Blaming Technology: The Irrational Search for Scapegoats. St. Martin's, N.Y., 1984.
- [15]. *Pagels, H.* The Dreams of reason: The computer and the rise of sciences of complexity. Simon&Schuster, N.Y., 1988.
- [16]. *Segal, H.* Technological Utopianism in American Culture. University of Chicago Press, Chicago, 1985.
- [17]. Winner, L. «Do artifacts have politics?», in: The Whale and Reactor: A search for limits in an age of high technology, Chapter 2, University of Chicago Press, Chicago, 1986.
- [18]. *Kerr, S.* Technology, Teachers and the Search for School Reform. Educational Technology Research and Development, Vol. 37(4), pp. 5—11.
- [19]. *Joyce, B. & Weil, M.* Models of Teaching. Englewood Cliffs, NJ: Prentice Hall, 1986.
- [20]. *Dwyer, D., Ringstaff, C., Sandholz, J.* The evolution of teachers' instructional beliefs and practices in high access—technology classrooms. Paper presented at the Annual Meeting AERA, Boston, MA, 1990.
- [21]. *Olson, J.* Schoolworlds/Microworlds: Computers and the Culture of the Classroom. New York, Oxford, 1985.
- [22]. *Sheigold, K. & Hadley, M.* Survey of Teachers' Uses of Computers. New York: Bank Street College of Education, Center on Educational Technology, 1990.
- [23]. *Wisk, M., Zoghates, P.* How Technology Affects Teaching. ETC Publication No. TR 87—10, Cambridge, MA: Harvard University, 1988.
- [24]. *Greenfield, P.* Mind and Media: The Effects of Television, Video Games and Computers. Cambridge: Harvard University, 1971.
- [25]. *Kubey, R. & Csikszentmihalyi, M.* Television and the Quality of the Life: How Viewing Shapes Everyday experience. Hillsdale, NJ: Lawrence Erlbaum Associates, 1990.
- [26]. *Solomon G.* TV is «Easy» and Print is «Tough»: The Differential Investment of Mental Effort in Learning as a Function of Perceptions and Attributions. Journal of Educational Psychology, Vol. 76, pp. 647—658.

Автором статьи «Универсальная тест-программа для проверки знаний в школе», опубликованной в ИНФО № 4-93, является Карпов М. В. Редакция приносит автору извинения за опечатку.

В статье Д. Ю. Усенкова «Реализация многоцветной закрашки на Бейсике» (ИНФО, №4, 1993 г.) в кодовом листинге 16 PATH. BIN допущена опечатка.

Вторая строка кодов должна выглядеть следующим образом:

000074 000411 021627 126310 001771 021627 126320 001033

Ассемблерный листинг ошибок не содержит.

## О РАЗМЕРАХ ДОПЛАТ УЧИТЕЛЯМ ИНФОРМАТИКИ

Мы получили следующие разъяснения у главного специалиста отдела труда и заработной платы Министерства образования РФ Соколовой В. Д.

Постановлением Минтруда РФ №48 от 4 марта 1993 г. «Об утверждении разъяснения «О порядке установления доплат и надбавок работникам учреждений, организаций и предприятий, находящихся на бюджетном финансировании», отменено постановление Министерства труда и занятости населения РФ от 4 июня 1992 г. №16 «О размерах надбавок и доплат работников образования».

Однако указанное постановление Минтруда России не может служить основанием для отмены права работника на получение оплаты за дополнительную работу, не входящую в круг его основных обязанностей, за счет средств, которые должны выделяться на эти цели образовательному учреждению. Этим постановлением отменяется лишь ранее действовавший порядок установления доплат, их перечень и размеры.

Письмо Министерства образования РФ №67-М от 09.04.93 г. «О порядке установления доплат и надбавок работникам учреждений образования» объясняет вышеуказанное постановление Минтруда РФ и доводит до сведения следующее:

– «Размеры доплат и надбавок стимулирующего характера (за работу, не входящую в круг основных обязанностей работника) в пределах средств, направляемых на оплату труда, определяются учреждениями, организациями и предприятиями, находящимися на бюджет-

ном финансировании, самостоятельно. Размеры доплат и надбавок работников максимальными размерами не ограничиваются и определяются в зависимости от дополнительного объема работ, выполняемого ими».

В целях реализации прав, предусмотренных постановлением Минтруда РФ №48 от 4 марта 1993 г., Министерство образования РФ рекомендовало образовательным учреждениям:

– разработать и утвердить положения о порядке установления доплат и надбавок в данном учреждении за счет бюджетных средств и средств от внебюджетной деятельности.

– руководствоваться рекомендациями, изложенными в инструктивно-методическом письме Министерства образования РФ №112-М от 23.06.93 «Об организации обучения информатике, аттестации учителей и оплате их труда (ИНФО №3-93 г.)».

В зависимости от объемов дополнительной работы и ее качества размеры доплат могут быть стабильными или различными в разное время года. Однако это условие должно быть оговорено образовательным учреждением в указанном положении.

Ответ на вопрос о доплате за работу в неблагоприятных условиях труда был опубликован в №2 за 1993 г. нашего журнала на стр.128. Письмо Министерства образования Российской Федерации от 12.01.1993 г. №10/32-Т, в котором говорится о размере существующих доплат за вредные условия труда – действует и в настоящее время.

## ТИПОВОЕ ПОЛОЖЕНИЕ

*об аттестации педагогических и руководящих работников  
государственных, муниципальных учреждений и организаций образования  
Российской Федерации*

### I. ОБЩИЕ ПОЛОЖЕНИЯ

11. Настоящее Типовое положение в соответствии со статьей 28 Закона Российской Федерации «Об образовании» регламентирует порядок проведения аттестации педагогических и руководящих работников государственных, муниципальных учреждений и организаций образования и является единой федеральной основой для разработки положений об аттестации в республиках в составе Российской Федерации, краях, областях, автономных образованиях, городах Москве и Санкт-Петербурге, министерствах и ведомствах Российской Федерации.
12. Целью аттестации является стимулирование роста квалификации, профессионализма, продуктивности педагогического и управленческого труда, развитие творческой инициативы, обеспечение социальной защищенности работников отрасли в условиях рыночных отношений путем дифференциации оплаты их труда.

Задачами аттестации являются:

- целенаправленное, непрерывное повышение профессионального уровня педагогических и руководящих работников;
- установление соответствия между качеством и оплатой труда;
- управление качеством образования для создания оптимальных условий развития личности.

Аттестация — средство решения поставленных задач путем установления соответствующих квалификационных категорий и разрядов оплаты труда на основе Единой тарифной сетки.

Аттестация осуществляется на основе комплексной оценки уровня квалификации, педагогического и управленческого профессионализма и продуктивности деятельности работников учреждений образования.

14. Нормативной основой для проведения аттестации является данное Типовое положение.

... Квалификационные категории педагогическим и руководящим работникам присваиваются на срок, определяемый территориальными и ведомственными положениями, но не более, чем на пять лет.

17. Педагогическим и руководящим работникам учреждений образования, не изъя-

вившим желания проходить аттестацию на присвоение квалификационной категории, разряды оплаты труда по ЕТС устанавливаются:

- педагогическим работникам — в зависимости от образования и стажа педагогической работы (стажа работы по специальности);
- руководящим работникам — по начальному разряду в диапазоне разрядов, соответствующем группе по оплате труда руководящих работников, к которой отнесено учреждение образования

19. Контроль за соблюдением порядка проведения аттестации, правовой и социальной защищенности педагогических и руководящих работников учреждений образования Российской Федерации осуществляют Министерство образования Российской Федерации, государственные органы управления образованием.

### II. Аттестационная комиссия, ее состав

25. Результаты аттестации определяются открытым (тайным) голосованием на итоговом заседании комиссии. По результатам проведенной аттестации комиссия выносит по каждому работнику одно из решений:

- а) о соответствии заявленной квалификационной категории;
- б) о несоответствии заявленной квалификационной категории;
- в) о соответствии занимаемой должности;
- г) о несоответствии занимаемой должности.

Решение считается действительным, если в голосовании участвовало не менее 2/3 членов утвержденного состава комиссии и оно было принято большинством голосов.

В случае признания работника не соответствующим заявленной квалификационной категории, повторная аттестация на ту же квалификационную категорию по заявлению работника может производиться не ранее, чем через год.

### III. Порядок проведения аттестации

32. Основанием для включения в списки аттестуемых является личное заявление ра-

ботника с указанием квалификационной категории, на которую он претендует, с обоснованием.

Работник, не имеющий необходимого стажа работы и уровня образования, заложенных в квалификационных характеристиках, может претендовать на любую квалификационную категорию и получить ее при успешном прохождении аттестации.

**3.3.** Аттестация проводится по двум направлениям.

**3.3.1.** Первое – обобщение итогов деятельности педагогических и руководящих работников.

Может осуществляться в форме собеседования, творческого отчета, защиты научно-методической или опытно-экспериментальной разработки и других формах.

При этом учитывается знание преподаваемых предметов, методик преподавания и методик воспитательной работы, теории педагогики и педагогической психологии, уровень практических навыков и умений, а также оценивается творческая деятельность педагогического работника.

**3.3.2.** Второе – экспертная оценка практической деятельности педагогических и руководящих работников.

Может осуществляться путем использования различных форм психолого-педагогической диагностики и изучения результативности профессиональной деятельности.

**3.4.** Решение о соответствии педагогических и руководящих работников заявленной квалификационной категории выносятся:

**высшей категории** – Главная аттестационная комиссия;

**первой категории** – Районная (окружная, муниципальная), городская аттестационная комиссия;

**второй категории** – Аттестационная комиссия учреждения образования.

**3.6.** За педагогическими и руководящими работниками, вышедшими на аттестацию по собственному желанию до истечения срока установленной ранее квалификационной категории и не получившими положительного решения аттестационной комиссии, сохраняется имеющаяся квалификационная категория до истечения срока ее действия.

**3.8.** При переходе на новое место работы, в том числе в связи с переездом в другой регион, за педагогическими и руководящими работниками сохраняется присвоенная по прежнему месту работы квалификационная категория.

**3.9.** В случае истечения срока действия квалификационной категории педагогических и руководящих работников во время:

- длительной нетрудоспособности;
- отпуска по уходу за ребенком;
- длительной командировки на работу по специальности в российские образовательные учреждения за рубежом;

– отпуска до одного года в соответствии с пунктом 5 статьи 55 Закона Российской Федерации «Об образовании»

по выходе на работу срок действия имеющейся у них квалификационной категории продлевается в индивидуальном порядке не более, чем на один год.

Педагогическим и руководящим работникам, прекратившим педагогическую деятельность в связи с ликвидацией учреждения образования или уходом на пенсию, независимо от ее вида, в случае возобновления ими педагогической деятельности сохраняется имевшаяся квалификационная категория до окончания срока ее действия или (в случае истечения срока действия) продлевается не более, чем на один год.

В случае истечения срока действия квалификационной категории у педагогических и руководящих работников, которым до пенсии по старости (по возрасту) осталось менее трех лет, имеющаяся у них квалификационная категория сохраняется до наступления пенсионного возраста.

**3.10.** Сроки действия квалификационных категорий, порядок и условия, дополняющие, конкретизирующие настоящее Типовое положение, определяются территориальными и ведомственными положениями.

#### IV. Реализация решений аттестационной комиссии.

##### Рассмотрение трудовых споров, связанных с аттестацией.

**4.1.** Руководитель органа управления образованием, учреждения образования на основании решения соответствующей аттестационной комиссии в течение двух недель издает приказ о присвоении (снятии) работникам квалификационных категорий и установлении уровня оплаты труда по соответствующим им разрядам; увольнении (переводе с согласия работника на другую работу) педагогических и руководящих работников в случае несоответствия их занимаемой должности.

Оплата труда работника в соответствии с присвоенной квалификационной категорией осуществляется с момента вынесения аттестационной комиссией соответствующего решения.

**4.2.** Педагогическим и руководящим работникам, которым по результатам аттестации присвоены квалификационные категории, делается соответствующая запись в трудовой книжке.

**4.3.** Трудовые споры, связанные с аттестацией, рассматриваются в соответствии с действующим законодательством о порядке рассмотрения трудовых споров.

## ПРИНТЕРЫ ПРОИЗВОДСТВЕННОГО ОБЪЕДИНЕНИЯ «РАДИЙ»

Где приобрести недорогой и надежный принтер?

Разрешить эту проблему вам поможет производственное объединение «Радий», продукция которого заинтересует каждого, кто работает с компьютером.

С конца 1989 года ПО «Радий» начало серийный выпуск устройства ввода-вывода информации печатающее (УВИП) «Электроника МС-6312» — малогабаритного безударного чернильно-струйного принтера, предназначенного для вывода алфавитно-цифровой и графической информации, поступающей из ЭВМ.

УВИП позволяет регистрировать информацию на стандартной (формата А4), рулонной и перфорированной бумаге. «Электроника МС-6312» печатает текст как обычным шрифтом (80 знаков в строке, скорость 150 зн/с), так и уплотненным (120 знаков в строке, скорость 150 зн/с), расширенным (40 знаков в строке, скорость 75 зн/с), расширенно-сжатым шрифтами, шрифтом с подчеркиванием. Принтер данной модификации имеет возможность воспроизводить подстрочный и надстрочный символы, а также различные комбинации шрифтов. Он реализует стандартный набор из 162 печатных символов как в кодировке КОИ-7 по

ГОСТ 27463-87, так и в кодировке, принятой в ПЭВМ типа IBM PC. Подключение его возможно к любому компьютеру, имеющему интерфейс «Centronics» или ИРПР.

В принтере используется термоструйная печатающая головка «Электроника МС-6902», которая обеспечивает печатание не менее 600 страниц формата А4 текста Шеннона.

По набору команд УВИП «Электроника МС-6902» совместим с принтерами FX-80 фирмы Epson.

ПО «Радий» постоянно ведет работу по усовершенствованию выпускаемой продукции.

В 1992 году был запущен в серию новый тип принтера «Электроника MS-6312». Вместе с теми достоинствами, которые отличают предыдущую модель, этот УВИП имеет ряд существенных преимуществ. В «Электронике MS-6312» реализована полная таблица кодировки 256 печатаемых символов, как алфавитно-цифровых, так и псевдографических; изменена форма некоторых строчных букв и увеличена их высота; улучшен алгоритм движения каретки; изменен алгоритм печатающей головки, что позволило осуществить более четкую пропечатку



вертикальных линий при одновременной работе всех разрядов.

В настоящее время основной моделью, выпускаемой объединением, является УВИП «MS-6312M» («РАДИЙ»).

Увеличение объема памяти ПЗУ, значительная доработка и модернизация печатающей головки позволили существенно расширить возможности работы принтера данной модели.

УВИП «MS-6312M» («РАДИЙ») позволяет реализовать большее количество шрифтов, в том числе:

- шрифт двойной высоты (дубль);
- качественный шрифт (NLQ);
- шрифт «ПАЙКА» (65 символов в строке);

а также их комбинации.

Значительно увеличено быстродействие прибора, скорость печати составляет не менее 225 зн/с.

Модель «РАДИЙ» позволяет использовать при эксплуатации печатающие головки фирмы «Hewlett Packard», обеспечивает работу в графическом режиме.

К несомненным достоинствам наших принтеров относится и то, что работают они бесшумно, низка их потребляемая мощность (меньше 14 Вт). Не менее привлекательны и массогабаритные характеристики: прибор, размеры корпуса которого всего 275x165x55 мм, займет минимум места в офисе, классе, квартире. Масса принтера с автономным блоком питания (который входит в комплект поставки) 2,2 кг.

Стоимость наших приборов значительно ниже других российских аналогов.

С нашими принтерами легко и приятно работать!

Они надежны и создают массу удобств и широчайших возможностей для пользователей.

### СПЕШИТЕ!

При галопирующей инфляции мы вынуждены постоянно повышать цену на свою продукцию.

ПО «Радий» заключает договора на поставку принтеров, осуществляет их продажу оптом и в розницу, проводит гарантийное и постгарантийное обслуживание.

**ПО ВСЕМ ВОПРОСАМ ОБРАЩАТЬСЯ ПО АДРЕСУ:**

125057, г. Москва, ул. Часовая, 28.  
Тел.: (095) 151-47-31  
ПО «Радий»

Сервисно-Информационный центр  
ПО «Радий»:  
г. Москва, ул. Усиевича, д. 31 а.

---

Распространением журнала «Информатика и образование»  
в зарубежных странах занимается Акционерное общество  
«Международная книга»

через своих конгр-агентов в соответствующих странах.

Адреса фирм-агентов А/О «Международная книга»

Вы можете узнать

в редакции журнала или в А/О «Международная книга»:

117040, Россия, Москва, улица Большая Якиманка, 39.

<b>Факс:</b>	<b>(095) 238-46-34</b>
<b>Телефон:</b>	<b>(095) 238-49-67</b>
<b>Телекс:</b>	<b>411160</b>
<b>Индекс журнала:</b>	<b>70423</b>
<b>Периодичность:</b>	<b>6 номеров в год</b>

---

*Our journal is exported by Joint-Stock Company  
«Mezhdunarodnaya Kniga» through their agents around the world.*

*Address of the company «Mezhdunarodnaya Kniga»:*

*117040, Russia, Moscow, Bolshaya Yakimanka, 39.*

<b>Telefax:</b>	<b>(095) 238-46-34</b>
<b>Phone:</b>	<b>(095) 238-49-67</b>
<b>Telex:</b>	<b>411160</b>
<b>Index:</b>	<b>70423</b>
<b>Issues per year:</b>	<b>6</b>
<b>Price for year</b>	
<b>Included air delivery:</b>	<b>39 US Dollars</b>

---



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.



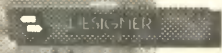
PhotoMaster служит для обработки сканированных изображений в формате TIFF.



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 делает 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 сцену 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.

Работа с MultiVision требует немного, но дает намного больше.

**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/я 80 AIST AB Москва Россия Телефон 095-229 67 06

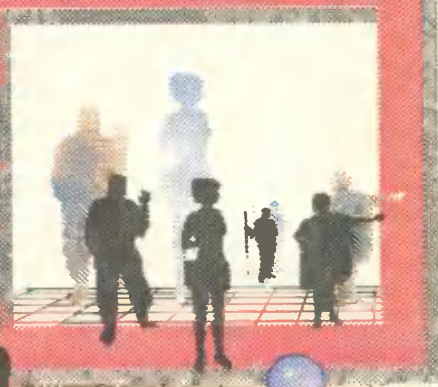
# MultiVision



MultiVision -  
это удобный  
Конструктор  
Презентаций

45-132

MultiVision - это пакет программ  
для деловых людей, преподавателей  
и специалистов в любой области.



**AIST** Advanced Instructional  
Software Trading AB