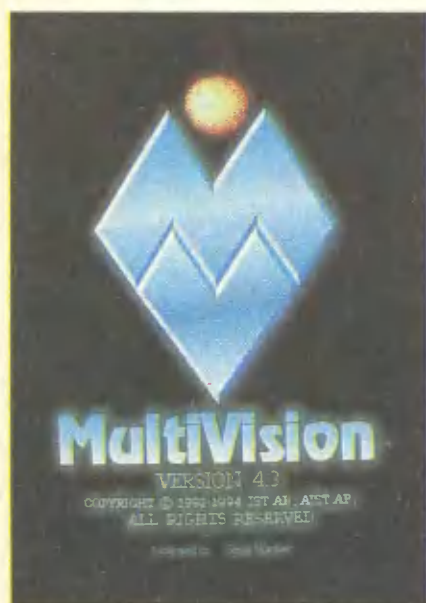


# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

3 1994



# ДАВАЙТЕ ОТПРАВИМСЯ В XXI ВЕК ВМЕСТЕ!



Мы неоднократно публиковали материалы, рассказывающие о профессиональном демонстрационном программно-аппаратном комплексе **MultiVision PRO.**

Специалисты НЦПСО и AIST в следующих номерах нашего журнала подробно расскажут об аппаратуре, входящей в состав комплекса. Будет также представлена новая модель компьютера, специально собранная и ориентированная на

**MultiVision PRO.**

На первой странице обложки – комплекс **MultiVision PRO** в полной комплектации.

	НЦПСО	AIST AB	НЦПСО-комплекс	AIST Inc.
Телефон	272 26 71	229 67 06	214 46 49	271 29 89
Факс	271 04 28	229 76 53	271 04 28	271 04 28

Научно-методический журнал  
Учрежден Министерством  
образования РФ  
и коллективом редакции

Издается с августа 1986 г.  
Выходит шесть раз в год

## СОДЕРЖАНИЕ

### ОБЩИЕ ВОПРОСЫ

- Уваров А. Ю. Новые информационные технологии  
и реформа образования 3

### МЕТОДИКА

- Штернберг Л. Ф. ЭВМ перебирает варианты,  
или об одной охотничьей истории 15
- Ерохина Е. А. От логики к программированию  
(Пролог в курсе информатики) 21
- Удалов С. Р. Обучение конструированию с применением  
компьютера на уроках черчения 35
- Маркова О. Н. Обработка массивов на языке БЕЙСИК 39

### ОЛИМПИАДА ПО ИНФОРМАТИКЕ

- Кирюхин В. М. VI Всероссийская олимпиада школьников  
по информатике 47
- Волченков С. Г. Задача «Паркет» 52
- Андреева Е. В. Задача «Буквоед» 55

### ИНФОРМАТИКА В МЛАДШИХ КЛАССАХ

- Горвиц Ю. М. Зачем нужны компьютеры  
в дошкольных учреждениях? 63
- Вершинин С. В. Компьютер для школьников 73
- Левченко И. В. Играем в «Телефон» 74

### НОВЫЕ ПРОЕКТЫ

- Паронджанов В. Д. Каким будет школьный  
алгоритмический язык XXI века? 77

Главный редактор  
академик  
БЕЛОЦЕРКОВСКИЙ О.М.

### РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Болотов В. А.  
Бредихин Г. А.  
Васильев Б. М.  
Зайдельман Я. Н.  
Зубченко А. А.  
Киселев Б. Г.  
Королев В. А.  
Кравцова А. Ю.  
Краснов А. Я.  
Кузякин А. П.  
Курнешова Л. Е.  
Лапчик М. П.  
Леонов А. Г.  
Пахомова Н. Ю.  
Савин А. Ю.  
Самовольнова Л. Е.  
Сапрыкин В. И.  
Смекалин Д. О.  
Уваров А. Ю.  
Угринович Н. Д.  
Урнов В. А.  
Фурсенко А. И.  
Хорошилов В. О.  
Христочевский С. А.  
Чуриков П. А.  
Щенников В. В.

## РЕДАКЦИЯ

*Первый зам. гл. ред.*

**Кравцова А. Ю.**

*Зам. гл. ред.*

**Васильев Б. М.**

*Ответственный секретарь*

**Иванова Т. В.**

*Редактор отдела*

**Усенков Д. Ю.**

*Компьютерная верстка*

**Павченко О. Н.,**

**Кириченко И. Б.**

*Технический редактор*

**Луговская Т. В.**

*Корректор*

**Антонова В. С.**

*Технический директор*

**Иванов В. А.**

*Экономический отдел*

**Бородаева З. В.**

*Отдел подписки и распространения*

**Мартынова И. С.**

**(208-70-02)**

*Информационное агентство ИА ИНФО*

**Васильева Н. А.**

**208-67-37**

## ТОЧКА ЗРЕНИЯ

**Байбаков В. О. ЭВМ в вузе**

93

## ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

**Свердлов С. З. Как нарисовать программу**

99

**Илларионов В. В. Графические технологии в Турбо-Паскале системы MSX**

104

**Демкина Н. Л. Конкурс «Пользователь ЭВМ-94»**

107

## КЛУБ «АГАТ»

**Костарева Н. А. Системы счисления**

109

**Петрова Н. В. Контроллер «CARD-93» для ПЭВМ «Агат»**

110

## КЛУБ «ЯМАХА»

**Попов В. А. «БРИДЖ-ИТ» на «Ямахе»**

111

## КЛУБ БК

**Саяпин А. А. Сетевая система SPRUT2. Краткое описание**

115

**Разбитной С. А. Геометрическое моделирование на БК**

116

## EX LIBRIS

118

## КЛУБ УКНЦ

**Брусенцев В. А. Новая жизнь УКНЦ**

121

## НАМ ПИШУТ

**БЕЙСИК КУВТ-86: работа с клавиатурой**

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

E-Mail: info@tit-bit.msk.su

При перепечатке материалов ссылка на журнал обязательна. Редакция не несет ответственности за содержание рекламы и используемые в ней товарные знаки.

Подписано в печать с оригинал-макета 25.08.94. Формат 70x100 1/16. Бумага офсетная. Печать офсетная. Усл. печ. л. 10,40. Усл. кр.-отг. 11,70. Уч.-изд. л. 13,5. Тираж 21 000 экз. Заказ 3126.

Цена по подписке:

для индивидуальных подписчиков 1200 руб. (индекс 70423);

для предприятий и организаций 2400 руб. (индекс 73176).

В розницу цена договорная.

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

**А. Ю. Уваров,**

*Научный совет по комплексной проблеме «Кибернетика» РАН*

## НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И РЕФОРМА ОБРАЗОВАНИЯ

Где наши знания, затерявшиеся в потоках информации?  
Где наша мудрость, утонувшая в море знаний?

*Т. Элиот. Голоса из камня*



Уваров Александр Юрьевич — член редколлегии журнала «Информатика и образование», канд. пед. наук, заведующий лабораторией «Телекоммуникация в образовании» Научного совета по комплексной проблеме «Кибернетика» РАН. Область научных интересов — реформа образования в условиях информатизации общества. Последние годы ведет работу по проблемам использования компьютерной коммуникации и других средств НИТ для изменения содержания, организационных форм и методов обучения в общеобразовательной школе.

Всякого, кто знакомится с новыми информационными технологиями, поражает диапазон возможностей, которые они открывают для совершенствования учебного процесса и системы образования в целом. Попытка осознать эти возможности заставляет обратиться к макроконтексту, из которого практически одновременно появились как сама задача (изменение сложившейся системы образования в соответствии с требованиями складывающегося информационного общества), так и средства ее решения (новые информационные технологии и в том числе компьютерная коммуникация).

### «Информационное общество»

Первоисточник сегодняшних разговоров об «информатизации общества» появился в недрах департамента коммерции (США). Статистические отчеты о деятельности американских предприятий в конце 50-х гг. показали, что число служащих («белых воротничков») впервые в истории

превысило число производственных рабочих («синих воротничков»). Вслед за этим экономист Фриц Машлуп показал на «рост индустрии знаний» [1], а гарвардский социолог Даниель Белл выпустил свою книгу [2], где объявил о наступлении «постиндустриального общества». Основываясь на этих данных, департамент коммерции подготовил девятитомный отчет «Информационная экономика» [3].

Однако потребовалось еще несколько лет, прежде чем после широко разошедшихся работ Алвина Тоффлера «Третья волна» [4] и Джона Найсбита «Мегатренды» [5] разработанные еще в 40-х гг. нашего века академиком В. И. Вернадским идеи «ноосферы» объединились с идеей «глобальной информационной экономики» и овладели вниманием общественности.

Появившееся затем исследование Шюшаны Зубовой «В век умных машин» [6] подтвердило: изменения в сфере производства, которые происходят в настоящее время, во многом сравнимы с изменениями,

которые происходили в промышленности на первых этапах индустриализации около двух столетий назад. Зубова провела наблюдения во многих американских компаниях за теми изменениями, которые приносит широкое использование информационной технологии. Она зафиксировала ключевое влияние процесса информатизации на жизнедеятельность отдельных промышленных предприятий, изучила реакцию работников на изменение условий их труда и новое перераспределение власти на производстве. Около десяти лет назад произошло осознание процессов «информатизации общества» и в нашей стране [7].

Итак, в основе ведущихся сегодня дебатов об изменении содержания образования «в ответ на изменение условий общественной жизни людей» лежит представление о становящемся «информационном обществе», которое идет на смену «индустриальному обществу», подобно тому как «индустриальное общество» пришло на смену «аграрному обществу».

Действительно, до начала XIX в. основная масса людей была занята в сфере сельскохозяйственного производства. В процессе индустриализации и массового переселения в города это соотношение постепенно изменялось. Уже в начале XX в. больше половины рабочей силы большинства европейских стран и США было занято в сфере промышленного производства. К середине XX в. «индустриальное общество» окончательно пришло на смену «аграрному». Связанные с этим глобальные изменения нашли самое широкое отражение в литературе и искусстве. Вместе с индустриализацией родился кинематограф, ставший новым искусством «индустриального общества». Были созданы средства массовой информации. Всеобщее среднее образование, развитие масштабов высшего образования и превращение науки в производительную силу привели к следующему этапу глобальных изменений, которые мы наблюдаем с вами сегодня. Можно считать, что в 80-е гг. XX в. на смену «индустриальному» пришло «информационное общество». Говоря о смене доминирующего вида деятельности людей в сфере общественного производства к середине 80-х гг. в США, Джон Найсбит [5] приводит такие данные:

- около 4% активной рабочей силы было непосредственно занято в производстве продуктов питания (сельское хозяйство);
- около 14% активной рабочей силы составляли собственно рабочие;
- более 60% активной рабочей силы составляли работники умственного труда

(пользуясь терминологией Найсбита, «информационные работники»).

Хорошее представление о сдвигах в структуре рабочих мест до конца этого тысячелетия на примере США дает приводимая ниже таблица, составленная по данным Хадсонского института [13] (см. стр. 3).

В условиях радикального усложнения жизни общества, его технической и социальной инфраструктуры решающим становится изменение отношения людей к информации, которая уже сейчас является таким же стратегическим ресурсом общества, как продукты питания в «аграрном», а традиционные материальные или энергетические ресурсы в «индустриальном» обществе.

«Информационная революция», как и предшествовавшие ей «аграрная» и «индустриальная» революции, сама по себе не решает стоящих перед обществом проблем. Снимаю одни проблемы, она порождает новые. «Несмотря на избыток информации, а может быть, отчасти и вследствие этого избытка,— пишет А. И. Солженицын [9],— Запад испытывает серьезные трудности в определении причин, порождающих современные события». По мнению Розак, избыток доступных многим современным американцам данных, которых больше, чем в состоянии переварить человеческое сознание, служит причиной снижения качества мышления, в том числе среди образованных членов современного общества. «Я встречал множество людей, которые потеряли способность мыслить, запутавшись в множестве доступных им фактов» [10]. Информационная перегрузка — это реальность. Ежегодно публикуется более 40 000 научных статей. Специалисты жалуются, что они не в состоянии прочесть всего, что относится к их предметной области. Огромные объемы информации, особенно собранные в виде статистических данных, являются полем для ошибочной и/или преднамеренно ложной интерпретации.

Новые средства обработки данных, прежде всего компьютеры, привели к появлению множества новых способов ведения пропагандистской работы среди людей. Типичным примером являются манипуляции с результатами опросов общественного мнения. Опасность, что немногочисленные «эксперты», контролирующие информационные потоки, могут эксплуатировать менее образованную часть населения, становится реальностью. Те, кто не имеет достаточного доступа к информационным системам, будут находиться в менее выгодном положении уже к концу этого века.

К печальным последствиям может привести злоумышленное использование мно-

## Изменения в структуре занятости населения США в 1984—2000 гг.

Вид занятий	Количество рабочих мест на рынке труда (тыс.)		
	в 1984 г.	новые к 2000г.	Рост (%)
Обслуживание	16 059	5 957	37
Менеджеры	10 893	4 280	39
Маркетинг и торговля	10 656	4 150	39
Канторские служащие	18 483	3 620	20
Техники	3 146	1 389	44
Врачи и медицинские работники	2 478	1 384	53
Учителя, библиотекари	4 437	1 381	31
Механики, монтажники и ремонтники	4 264	966	23
Транспортные рабочие	4 604	752	16
Инженеры, архитекторы	1 447	600	41
Строительные рабочие	3 127	595	19
Ученые в области естественно-математических наук	647	442	68
Артисты, писатели, художники, спортсмены	1 092	425	39
Другие представители свободных профессий	825	355	43
Юристы, судьи	457	326	71
Работники социального обеспечения, священники	759	235	31
Подсобные рабочие	4 168	205	5
Ученые в области гуманитарных наук	173	70	40
Рабочие предприятий прецизионной продукции	2 790	61	2
Рабочие по эксплуатации зданий и систем	275	36	13
Мастера и бригадиры	1 442	—6	0
Шахтеры	175	—28	—16
Рабочие на сборочно конвейере	2 604	—179	—7
Квалифицированные рабочие, операторы	5 527	—448	—8
Сельскохозяйственные рабочие, рыбаки, лесники	4 480	—538	—12
Всего	105 008	26 030	25

гочисленных досье, которые копятся на каждого человека в школе, в медицинских учреждениях, налоговой инспекции, в банках и десятках других подобных картотек. Вся эта информация о каждом из нас запоминается на лентах и дисках, и ее в любой момент можно вывести на дисплей компьютера и использовать для добрых и злых дел.

Различие в уровне информационного обеспечения сегодня становится одной из существенных причин в дисбалансе экономического развития передовых и слабо развитых стран, порождает нестабильность в отношениях между странами [11]. Информационная революция меняет глобальную экономику, преобразует национальную политику и предпринимательство, заставляет пересматривать внешнеполитические цели государств и методы их достижения. Подобно тому как распространение медицинских знаний подорвало всевластие знахарей, распространение информации о жизненных стандартах в других странах подорвало репутацию официальных идеологий. Неспособность старой ригидной политической структуры решить задачи информатизации общества явилась одной из практических причин экономического и политического кризиса, который переживаем сегодня мы с вами.

Знание всегда дает власть тем, кто им владеет и умеет пользоваться. Увеличение объема и распространение информации среди огромной массы людей являются предвестниками изменений существующих структур власти. «Информационная революция» ведет к изменению не только формального определения, но и самой природы национального суверенитета. Политический и экономический кризис, который переживает сегодня Россия, во многом связан с начавшимся переходом страны к жизни в «информационном обществе».

Будущее вторгается к нам уже сегодня. Глобальная информационная инфраструктура уже влияет на нашу жизнь:

- в Архангельском педагогическом институте разворачивают подготовку зарубежных учащихся по русскому языку с помощью компьютерной связи;
- педагоги и воспитанники нескольких десятков общеобразовательных школ регулярно связываются со своими коллегами в школах-партнерах как в России, так и за ее рубежами через учебную сеть PILOTNET, не выходя из своего компьютерного класса;
- программу CNN из Атланты (США) регулярно смотрят с помощью своих спутниковых антенн жители сотен городов и поселков по всей территории нашей страны;
- предприниматели 76 городов России

ежедневно просматривают биржевые сводки и другую коммерческую информацию по компьютерной сети RELCOM.

Вопрос о роли и месте системы образования на различных этапах построения «индустриального общества» (как в конце прошлого, так и в первой половине нынешнего века) еще ждет своих исследователей. Однако аналогия процессов трансформации образования при смене ведущей сферы общественного производства (от аграрной к индустриальной и от индустриальной к информационной) очевидна. «Индустриальная революция» привела к становлению системы всеобщего обязательного образования. Реформа этой системы, ее приспособление к нуждам информационного общества начинается сегодня.

### Обновление целей и содержания общего образования

До последнего времени влияние наступления «информационного века» на изменение целей и обновление содержания общего образования ощущается в школах довольно опосредованно. Приобретается вычислительная техника, учитель информатики рассказывает на своих уроках о том, что количество данных, собираемых и обрабатываемых в современном обществе, экспоненциально возрастает, говорят о необходимости использовать компьютеры в учебном процессе по другим предметам, и это почти все. Наша педагогическая общественность слабо осознает проблемы и опасности, которые несет широкое распространение информационных технологий, и с которыми придется столкнуться сегодняшним школьникам.

Задача подготовки учащихся к работе с информацией, к жизни полноценными гражданами в мире, который будет существенно отличаться от нашего прошлого и даже настоящего, пока в явном виде не сформулирована.

Можно считать общепризнанным, что образованный член «информационного общества» должен иметь возможность:

- доступа к базам данных и средствам информационного обслуживания;
- понимать различные формы и способы представления данных в вербальной, графической и числовой формах;
- знать о существовании общедоступных источников информации и уметь ими пользоваться;
- уметь оценивать и обрабатывать имеющиеся у него данные с различных точек зрения;
- уметь пользоваться техниками анализа статистической информации;



- уметь использовать имеющиеся данные при решении стоящих перед ним задач.

Учитель в «информационном обществе» перестает выступать перед своими учениками в качестве источника первичной информации. Он превращается в посредника, который облегчает ее получение. Вопрос о том, где взять необходимые сведения, заменяется вопросом о том, сколько данных и в каком виде в состоянии воспринять и усвоить учащиеся. Уже сегодня наши школьники должны изучать и понимать проблемы, которые возникают перед жителем «информационного общества», пронизанного новыми информационными технологиями (НИТ). Ясно, что существующие учебные программы неполны и фрагментарны. Предстоит решить, какие именно разделы требуют переработки.

НИТ существенно изменяет практику планирования (разработки планов), поиска необходимой информации, ее обработки и хранения, а также коммуникации (обмена результатами информационной работы). М. Вайт [12] считает естественным включить изучение соответствующих видов работы в учебный процесс. Она предлагает обучать школьников технике плановой работы как системе специальных умственных навыков. Электронные таблицы и системы поддержки построения сетевых графиков рассматриваются как естественные средства обеспечения этой работы. Школьники могут, например, рассчитывать ожидаемые изменения демографической ситуации в своем городе и обсуждать возможные способы решения проблем, которые возникнут через соответствующий период времени. Другой пример: они могут рассчитать изменение ситуации на рынке труда и обсудить свои возможные профессиональные карьеры в условиях прогнозируемого изменения конъюнктуры рынка рабочей силы.

Обучение поиску информации непосредственно связывается с умением пользоваться картотеками, справочниками, словарями и электронными базами данных. Учащиеся должны осваивать технику поиска данных из различных источников, представлять возможные способы организации данных и области поиска, уметь пользоваться ключевыми словами, кодами и различными видами дискрипторов. Работа с электронными базами данных требует достаточно развитых специализированных умственных навыков. Умение пользоваться реферативной информацией и проверять ее достоверность — первый шаг на пути к самостоятельной работе с информационными источниками, самостоятельному продуцированию значимой информации.

Умение организовывать хранение информации, анализировать ее и выбирать адекватные формы ее представления — еще одна важнейшая область специализированных умений и навыков, с которыми не знакомят учащихся в современной школе. Понять, что информационный ресурс представляет реальную ценность лишь в том случае, если доступ к нему соответствующим образом организован, — обязательная составляющая общего образования для жителей «информационного общества». Следует учить школьников доступным им способом эффективной организации данных на компьютере: создание каталогов поступающей в класс корреспонденции по электронной почте — хороший пример адекватной учебной задачи. Умение анализировать информацию, используя для этого соответствующие средства статистического анализа — еще один важный практический навык. Пакеты статистического анализа, электронные таблицы являются адекватными инструментами такой работы. Немаловажно умение сопоставлять между собой данные различных текстовых документов, уметь находить общее и различное в соответствующих материалах.

Умение наглядно представлять информацию в виде таблиц, графиков, гистограмм и различного вида диаграмм также становится обязательной частью современного общего образования. Использование программ-презентаторов делает эту задачу реальной. Речь идет не просто о том, чтобы научить школьников пользоваться соответствующими программными средствами. Учащиеся должны владеть этим инструментом в процессе собственно работы с данными, чтобы отвечать на вопросы, каковы эти данные, как они изменяются, как их можно оценивать.

Важнейшая задача образования в век информации — обучение коммуникационным навыкам. Текстовый редактор — один из мощнейших учебных инструментов, пришедших в школу вместе с компьютером. Графические редакторы позволяют по-новому рассмотреть вопрос об обучении техникам графического представления информации. Рисование и техническое черчение будут переживать такие же изменения в содержании и методах обучения, какие сегодня переживают родной язык и литература в связи с появлением текстовых редакторов со встроенными словарями синонимов и омонимов, средствами проверки правописания. Образное представление информации становится сегодня главным средством передачи данных. Прежде всего это видно на примере телевидения, коммерческой рекламы и т. п.

Предстоит учить школьников тому, как правильно читать эту информацию, а также формировать у них умение использовать эти средства для самовыражения. Текстовые и графические редакторы, программы-презентаторы доступны многим школам уже сегодня. В ближайшем будущем начнут распространяться простые и доступные средства работы с движущимися изображениями и звуком. Средства мультимедиа уже привлекли к себе пристальное внимание педагогов. Их освоение как инструмента формирования соответствующей информационной культуры школьников — дело ближайшего будущего.

Несмотря на всю значительность, перечисленные изменения целей образования, непосредственно связанные с изменениями в технике и средствах информационной работы, представляют собой лишь верхнюю часть айсберга тех изменений, которые идут в школу вместе с информационной и новой «индустриальной революцией».

По данным департамента труда США, опубликованным в 1991 г., умения читать, писать и считать уже недостаточно для современного работника. Столь же базовыми требованиями не только завтра, но уже и сегодня являются следующие пять требований:

- уметь организовать доступные ресурсы для обеспечения наибольшей эффективности производственного процесса;
- плодотворно сотрудничать с другими участниками производственного процесса;
- собирать, оценивать и использовать информацию;
- понимать и уметь использовать взаимосвязи между компонентами сложных производственных систем;
- уметь осваивать и использовать постоянно расширяющийся спектр производственных технологий.

Все эти требования являются реальными двигателями реформы школы, а использование новых информационных технологий в учебном процессе — катализатор соответствующих изменений.

### Реформа образования и новые информационные технологии

Компьютеры используются в наших школах уже более пяти лет. За это время у педагогов несколько раз менялись представления о том, для чего они нужны и как их использовать. В различных школах компьютеры используют по-разному, однако практически во всех случаях можно выделить несколько этапов, через которые проходит процесс компьютеризации школы.

На первом этапе в школе говорят, что

«ввели изучение информатики». И действительно, компьютеры используются исключительно как объект изучения. Вся имеющаяся в школе вычислительная техника установлена в кабинете информатики, и доступ к ней имеют только один-два учителя. Остальные педагоги рассматривают ЭВМ как дорогостоящее специальное оборудование, предназначенное для изучения специальной дисциплины и не думают об использовании компьютеров в своей работе.

На втором этапе обычно заявляют, что «компьютеры начали использовать для изучения других предметов». Учителя математики, физики, географии или начальной школы проводят отдельные занятия в кабинете информатики, где предлагают школьникам работу с обучающими и/или контролирующими программами. В основном это занятия по отработке учащимися тех или иных навыков и фронтального контроля знаний с помощью ЭВМ. Одновременно растет разочарование в так много обещавшей школе «компьютерной революции». Выясняется, что в рамках привычной организации учебного процесса возможности учебного применения ЭВМ довольно ограничены.

На третьем этапе компьютеры постепенно «расползаются» по школе. Становятся ясны ограничения в использовании «обучающих программ»: они используются только там, где дают заметный эффект. Учителя и учащиеся начинают использовать компьютеры главным образом как универсальный инструмент, приспособившая их для решения своих обычных задач. Главными средствами становятся универсальные программные средства массового применения: *текстовые процессоры, электронные таблицы, системы управления базами данных, электронная почта*. Регулярное использование компьютерной коммуникации, как правило свидетельствует о том, что третий этап освоения вычислительной техники в школе наступил. В этих школах, как правило, затрудняются ответить на вопрос о том, как используются компьютеры. Оптимизм и пессимизм первых этапов сменился привычным использованием новой технологии для решения повседневных задач, а попытка представить свою работу без компьютеров приводит в уныние: они уже стали частью жизни.

Большинство школ в нашей стране, где имеется вычислительная техника, находятся на первом или втором этапе ее освоения. И лишь единицы приближаются к следующему, четвертому этапу, связанному с освоением и использованием средств нового поколения: мультимедиа, глобальных и ло-

кальных (на CD ROM) баз данных, портативных компьютеров.

Несмотря на все трудности, переживаемые страной, процесс информатизации образования развивается. Он определяется двумя группами объективно действующих факторов [13]:

- внешними по отношению к образовательной системе, задающими условия, в которых функционирует школа,
- внутренними, определяющими готовность и способность общеобразовательных учреждений воспринять достижения научно-технического прогресса и эффективно использовать их для решения своих основных задач.

Внешние факторы характеризуются достигнутым уровнем развития информационной индустрии, широтой применения НИТ во всех сферах жизни общества. Внешние факторы определяют:

- доступность и качество средств вычислительной техники, технологически возможный на данный момент уровень информатизации образования;
- требуемую общеобразовательную подготовку и уровень информационной культуры выпускников.

Эти факторы, по существу, задают ограничения на темп и характер оснащения учебных заведений средствами вычислительной техники и программным обеспечением, практическую возможность широкого распространения отдельных нововведений.

Внутренние факторы в значительной степени контролируются самой системой образования и отражают текущий уровень развития педагогической науки и практики. Они определяют способность системы образования воспринять и освоить вновь появляющиеся средства работы с информацией и характеризуются:

- научно-методическим заделом (отработанные приемы эффективного использования НИТ в учебном процессе, программы для ЭВМ в комплекте с необходимыми учебными и методическими материалами), разработанностью нового содержания образования, методов и форм использования НИТ для повышения эффективности учебно-воспитательного процесса;
- достигнутым уровнем профессиональной подготовки педагогов, их знакомством с потенциальными возможностями НИТ и умением использовать эти возможности в своей практической работе;
- гибкостью системы управления образовательными учреждениями, ее готовностью к изменениям в содержании образования, способностью распозна-

вать и осваивать новое, перестраиваться, порождать и распространять прогрессивные организационные формы и методы учебной работы.

Внутренние факторы определяют, как на практике будет проходить использование средств НИТ, которые попали в сферу образования, насколько эффективными окажутся производимые капиталовложения.

Чтобы увидеть «движущие пружины» информатизации образования сегодня, достаточно посмотреть на динамику изменения этих факторов.

**ДОСТУПНОСТЬ И КАЧЕСТВО СРЕДСТВ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ.** Оснащение школ средствами вычислительной техники меняется достаточно быстро. Компьютеры, которые устанавливали в школах в 1985—1987 гг., сегодня выводятся из эксплуатации.

Около трех лет назад в школах начали устанавливать MS-DOS компьютеры. Вслед за тысячами учебных классов таких машин, развернутых в рамках проекта «Пилотные школы» [14], еще несколько тысяч школ будут располагать РС-совместимой техникой к середине 90-х гг. Новая тенденция — установка в школах компьютеров, объединенных в мощные профессиональные локальные сети с соответствующим сетевым программным обеспечением.

Можно утверждать, что сегодня MS-DOS компьютеры становятся фактическим стандартом в школе. В условиях промышленного спада и политической неустойчивости оснащение школ современными средствами вычислительной техники продолжается.

Другой феномен — компьютерная коммуникация. Первый эксперимент по подключению школ к глобальным компьютерным сетям [15] был встречен большинством специалистов с недоверием. Многим казалось, что до появления модемов в школах еще очень далеко. Сегодня сотни школ в различных регионах страны являются абонентами PILOTNET, GLASNET или RELCOM. В Самаре, Рязани, Петербурге и многих других городах органы образования поддерживают или разворачивают работу местных образовательных компьютерных сетей. Под руководством Министерства образования РФ начались работы по созданию единой Российской образовательной сети.

**УРОВЕНЬ ИНФОРМАЦИОННОЙ КУЛЬТУРЫ ВЫПУСКНИКОВ.** Компьютеры пришли в школу под лозунгом: «Программирование — вторая грамотность». Сейчас продолжается медленный процесс осознания того, что «алгоритмический стиль мышления» и «умение пользоваться

компьютером для решения своих задач» лишь верхушка айсберга нового содержания образования, привносимого в школу становящейся информационной культурой. Выработка навыков работы с клавиатурой, умение пользоваться текстовым редактором и электронными таблицами, построить и вести свою базу данных, подготовить иллюстрации к выступлению — эти очевидные сегодня требования к подготовке выпускников недостаточны. Перечисленные в предыдущем разделе новые требования начинают играть доминирующую роль при обсуждении целей и содержания общеобразовательной подготовки школьников.

Итак, мы видим, что направление действия внешних факторов изменяется. В середине 80-х они тормозили «волюнтаристски» начатый процесс информатизации школы, а сегодня стимулируют развитие этого процесса. Каково же сегодня действие внутренних факторов?

**НАУЧНО-МЕТОДИЧЕСКИЙ ЗАДЕЛ.** Первые представления об использовании ЭВМ в процессе обучения возникли около четверти века назад под флагом «технического перевооружения педагога», механизации его труда. Они развивались в рамках концепции «обучающей машины», имитирующей процесс индивидуальной работы учителя с обучаемым: компьютер предлагает ему порции учебного материала и устанавливает последовательность изложения на основе ответов обучаемого на контрольные вопросы. Сегодня уже разработаны и продолжают разрабатываться многие десятки подобных программ. Вместе с тем растет понимание ограниченности этой концепции. Все больше педагогов осознают, что основное влияние на методы и организационные формы обучения оказывают не обучающие программы. Решающая роль принадлежит НИТ как средству учебной деятельности, новому взгляду на цели и содержание образования, порождаемому процессом информатизации общества.

Стремление решить задачу развития познавательной самостоятельности школьников привело к появлению педагогической философии, впервые реализованной в рамках «Лого». ЭВМ, ее программное обеспечение («компьютерная среда» или «микромир»), дополнительно подключаемое периферийное оборудование (датчики и исполнительные механизмы), помещение, где происходят занятия, образуют специфическую среду, создающую условия для проявления познавательной активности учащихся. Ассортимент компьютерных сред сегодня быстро расширяется и охватывает большинство предметных областей. С их помощью осваивают письмо («Writing to

read»), ставят пьесы («Компьютерный театр на Макинтоше»), изучают физику («Physics Explorer»), биологию («Biology Explorer») и многое другое. Соответственно, меняется и роль учителя: направлять развитие личности учащихся, поддерживать творческий поиск и организовывать их коллективную работу.

Существенный вклад в изменение методов обучения вносит появление в учебном процессе автоматизированных экспериментальных установок («Лаборатория на базе ЭВМ»). С их помощью у учащихся появляется возможность проводить полноценные исследования: собирать и обрабатывать данные, анализировать получаемые результаты.

При изучении естественных дисциплин в школе в полной мере начинает реализовываться метод исследовательских проектов. Этому способствует также развитие общедоступных компьютерных сетей, позволяющих выполнять совместную работу учащимся различных учебных заведений, находящихся сколь угодно далеко друг от друга. Дешевые средства электронной коммуникации (электронная почта и телеконференции) позволяют развить невиданные ранее по своей эффективности методы коллективной работы у учащихся с представителями других культур, элементов полноценного планетарного сознания. Электронные базы данных предоставляют всем педагогам и учащимся простой и удобный доступ к практически неограниченным объемам информации.

В этих условиях неизбежен пересмотр сложившихся сегодня организационных форм учебной работы: увеличение самостоятельной индивидуальной и групповой работы учащихся, отход от традиционного урока с преобладанием объяснительно-иллюстративного метода обучения, увеличение объема практических и лабораторных работ, внеклассных и внешкольных занятий, которые становятся обязательной составляющей учебного процесса.

**УРОВЕНЬ ПОДГОТОВКИ ПЕДАГОГОВ.** Новые информационные технологии бесполезны, если педагоги не имеют достаточной технической и методической подготовки, действенных стимулов для использования ЭВМ.

Педагогическое освоение НИТ включает:

- «компьютерную грамотность» и накопление личного опыта практического использования компьютеров;
- методическую подготовку по использованию НИТ в учебном процессе.

Первоначально в содержание «компьютерной грамотности» входило знакомство с программированием. Сегодня признано, что наибольшую практическую ценность

имеет приобретение учителями навыков использования ЭВМ для подготовки, редактирования и хранения текстовой и графической информации, обработки данных, ведения личных архивов.

Появились первые отечественные компьютерные курсы, позволяющие рядовому пользователю быстро освоить работу с программными средствами массового применения [16]. Процесс практического овладения работой на компьютерах широкой массой педагогов сегодня на подъеме.

Овладение компьютерной грамотностью — первый шаг на пути методической подготовки педагогов к использованию НИТ в учебном процессе. Средства НИТ предоставляют практически неограниченные возможности для самостоятельной и совместной творческой деятельности учащихся и учителей, но почти бесполезны при традиционном информационно-объяснительном подходе к обучению.

Из «носителя истины» учителя превращаются в соучастников продуктивной деятельности своих воспитанников. Намечается качественный сдвиг в существующих сегодня отношениях между педагогами и учащимися. Компьютеры — третий партнер в учебном процессе. Они предоставляют новые возможности другим его участникам и требуют изменения сложившихся отношений между ними. НИТ предъявляет более серьезные требования к качеству труда и уровню квалификации педагогов как по объему знаний и их системной организации, так и по педагогическому мастерству. Такова цена, которую должен заплатить преподаватель за большую самостоятельность своих учеников, за большую эффективность их познавательного труда и за свое личное удовлетворение от возросшего уровня общения со своими учениками.

Сегодня осознано, что существенную роль в освоении НИТ играют сами информационные технологии.

Краткосрочная интенсивная методическая переподготовка педагогов в сочетании с оперативной справочно-консультационной поддержкой и обменом опытом в ходе последующей практической работы через систему электронной почты дает возможность каждому педагогу контактировать со своими коллегами, придерживающимися сходной методической позиции, на всей территории страны. Сочетание средств «обучения на расстоянии» с периодическими личными контактами педагогов-единомышленников облегчит развитие творческой инициативы, проникновение ростков нового в каждое учебное заведение.

**ГИБКОСТЬ СИСТЕМЫ УПРАВЛЕНИЯ ШКОЛОЙ.** Способность системы управления образованием быстро перестраивать-

ся, обеспечивая возникновение и распространение новых прогрессивных организационных форм и методов учебно-воспитательной работы, — необходимая предпосылка развития процесса информатизации образования. Начавшийся процесс диверсификации образования позволяет учебным заведениям сегодня самим выбирать формы и темпы освоения НИТ, наиболее отвечающие их конкретным условиям.

В регионах с более развитой инфраструктурой, в учебных заведениях, которые лучше подготовлены к развертыванию этой работы, освоение НИТ происходит опережающими темпами. Складывается сеть «точек роста», где педагоги-новаторы ищут приемы повышения эффективности учебно-воспитательной работы с использованием НИТ. Сегодня этот процесс направляется местными органами управления образованием. Принимаемые ими программы направлены на повышение уровня работы конкретных учебных заведений:

- подготовку и переподготовку педагогических кадров;
- оснащение школ методическими разработками, техническими и программными средствами, обеспечение их эксплуатации.

Расслоение школ по уровням использования НИТ продолжится. Складывается сравнительно немногочисленная группа учебных заведений, лидирующих в области использования НИТ.

Технологическая революция в образовании происходит на наших глазах. Новые информационные технологии в школе уже начали влиять на результаты подготовки учащихся. Однако дело не только в том, какие знания, умения и навыки приобретают школьники. Нельзя не согласиться с профессором С. Керром [17], что в долгосрочной перспективе еще важнее 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.

Все эти процессы уже начались, их результат далеко не ясен. Однако вернуться назад уже невозможно.

### **Компьютерная коммуникация как катализатор изменений**

Интерес к использованию компьютерной коммуникации в учебном процессе быстро растет. Вместе с другими новыми информационными технологиями компьютерная коммуникация обещает существенно

помочь в становлении нового содержания образования, изменения организационных форм и методов обучения. Появление компьютерной коммуникации в школе предполагает:

- интенсивное использование ЭВМ и безбумажной технологии как инструмента повседневной учебной работы учащихся и педагогов;
- изменение содержания обучения в традиционных дисциплинах и их комплексирование при выполнении учебных проектов;
- разработку методов самостоятельной поисковой и исследовательской работы учащихся в ходе выполнения учебных телекоммуникационных проектов;
- обучение учащихся методам коллективного решения проблем;
- подготовку учителей к работе с новым содержанием, новыми методами и организационными формами обучения.

Глобальная компьютерная связь открывает реальные возможности повседневного сотрудничества педагогов и школ, как в своей стране, так и с другими странами мира, по-новому ставит вопрос о возможностях формирования у школьников глобального сознания. Она стимулирует введение в практику методов проектной работы учащихся, способствует освоению учащимися навыков продуктивной совместной работы по достижению общей цели, создает предпосылки интеграции изучения средств информатики, языка (родного и иностранного) с освоением предметного содержания других общеобразовательных дисциплин.

Компьютерная коммуникация позволяет получить доступ к практически неограниченному массивам информации, хранящейся в централизованных банках данных. Это дает возможность учителям при организации учебного процесса опираться на весь запас знаний, доступных жителю «информационного общества».

Сегодня большинство экспертов согласны в том, что обращение педагогов к компьютерной коммуникации способствует радикальным изменениям в школе. Она служит основой для создания новых учебных предметов. Вместе с тем их создание весьма трудоемкое дело, так как обычно связано с изменением природы традиционной организации учебного процесса.

Компьютерные сети открывают школе доступ к ранее недоступным информационным ресурсам:

- университетские, правительственные, общественные и коммерческие собрания данных;
- работающие в компьютерной сети специалисты по различным областям зна-

ний, коллеги, живущие и работающие в существенно иной культурной среде;

- электронные конференции, где обсуждаются новейшие работы в области науки и техники, а также практически ведутся эти работы;
- электронные архивы программного обеспечения для ПЭВМ.

Работая в сети, педагоги могут поддерживать регулярные рабочие контакты со своими коллегами в других школах, узнавать об особенностях преподавания знакомых им учебных предметов в других учебных заведениях, знакомиться с новыми и альтернативными учебными программами и учебниками, следить за новыми теоретическими и прикладными разработками в области теории обучения, методики, педагогической психологии, дефектологии.

Однако главное достоинство компьютерной коммуникации, как новой для школы информационной технологии, состоит в том, что она замыкает электронную информационную среду в школе, позволяя школьникам и учителям работать с компьютером как универсальным средством обработки информации. Не существует учебной компьютерной коммуникации (в отличие от учебных редакторов, электронных таблиц и баз данных). Использование компьютерной коммуникации естественно сочетается с интенсивным использованием профессиональных текстовых и графических редакторов, электронных таблиц, баз данных. Использование глобальной компьютерной связи естественно сочетается с созданием внутришкольной локальной компьютерной сети. Без внутришкольной сети широкое использование глобальной компьютерной связи в учебном процессе невозможно.

Компьютерная коммуникация замыкает также и создание соответствующей учебной среды жителя «информационного общества». Поэтому можно сказать, что появление компьютерной коммуникации, дополняя уже существующие средства НИТ, способствует изменению:

- форм и способов взаимодействия между учителями и учащимися;
- средств работы с информацией, с помощью которых учащиеся и учителя решают стоящие перед ними задачи;
- представление о том, где и когда действительно происходит обучение школьников;
- природу учебных задач, которые должны решать школьники.

В результате учащиеся, работающие в такой среде и имеющие соответствующую методическую поддержку:

- эффективно используют НИТ для ре-

шения рутинных задач (подготовка текстов, таблиц, сбор, обработка и представление экспериментальной информации, поиск необходимых данных и коммуникация с другими членами «электронного сообщества»);

- способны самостоятельно осваивать и использовать необходимые для них средства (готовы и в состоянии самообучаться);
- осознают, что именно они знают и умеют, и готовы в любой момент поделиться этими знаниями со своими коллегами;
- умеют сотрудничать со своими коллегами;
- лучше понимают проблемы, возникающие в ходе коллективной работы;
- могут хорошо объяснить другому сущность и устройство достаточно сложных процессов и систем;
- собирают и представляют информацию в различных формах (таблицы графики, отчеты и даже гипертекст и видеоряды);
- с оптимизмом смотрят в будущее, настроены на продолжение образования и на работу в наиболее сложных и престижных областях общественного производства, политики, управления, торговли.

Однако это не происходит само собой. Информационный век стимулирует нас перепрыгнуть через границы отдельных дисциплин и развивать более широкие представления, обучая школьников на основе междисциплинарного подхода. По мере новых открытий в науке и новшеств в технологии предметное содержание быстро устаревает. Сегодня документ об окончании школы или вуза по прошествии нескольких лет ничего не значит, если человек не берет на себя ответственность за самообразование, пополняя и обновляя свои знания, умения и навыки.

Однако существующая структура и организация школы, как и привычные учебные предметы, особенно в старших классах средней школы, обычно не ведут к провалу попыток двигаться в этом направлении.

Работа с готовыми учебными проектами, представляемыми через компьютерную сеть различными учебными центрами, редко согласуется с текущим расписанием школьных занятий. Эти проекты обычно плохо соответствуют действующим в школе учебным планам и программам, и в любом случае они требуют дополнительных временных затрат со стороны учащихся и педагогов.

Большинство курсов по подготовке и переподготовке педагогов, связанных с освоением НИТ, и в том числе компьютерной коммуникации, концентрируют внимание

на технических аспектах компьютерной связи, а не на сопряженных с ее использованием изменениях содержания, методов и организационных форм обучения. Но и после того как учителя начинают практически работать в сети, им, особенно первое время, требуется помощь в распознавании доступных информационных ресурсов, овладении наиболее эффективными приемами использования технических и программных средств.

Еще одна типичная трудность связана с недостаточно свободным доступом школьников к ресурсам компьютерной сети. В школе используется лишь несколько модемов (обычно один). Доступное время работы в сети обычно также ограничено. В этих условиях лишь немногим из сотен школьников удается получить (сильно ограниченный) доступ к ресурсам сети. Компьютерная коммуникация не может стать инструментом учебной работы без того, чтобы учащиеся получили возможность регулярно пользоваться сетью. Единственное практическое решение состоит в том, чтобы развернуть общедоступную локальную внутришкольную сеть с соответствующими мостами, позволяющими пользоваться услугами глобальной компьютерной связи.

Когда эти проблемы в той или иной степени преодолены, в школе можно начать более или менее успешное экспериментирование с использованием проектного метода, групповой работой, интеграцией учебных дисциплин, с изменением учебной среды, за счет включения в нее средств новой информационной технологии.

Как показывает опыт, результаты оказываются наиболее впечатляющими в тех случаях, когда введение компьютерной коммуникации объединено с использованием других средств НИТ и поддерживает широкий диапазон методических и организационно-педагогических нововведений (работа в группах сотрудничества, проектная работа, альтернативные методы оценки достижений школьников и т. п.). В отличие от тех школ, где предпочитают опробовать нововведения по одному, в школах использующих множественные нововведения, педагоги первоначально чувствуют себя весьма неуверенно. Однако, если в первом случае «одиночные» нововведения мало что меняют в практике работы школы, «множественные» нововведения оказывают существенно более заметный эффект. После того как в педагогическом коллективе произойдет соответствующее перераспределение ролей, а соответствующие технические средства и педагогические техники станут достаточно привычны, в школах, пошедших по пути множественных нововведений, учебная среда действительно из-

меняется, что делает для большинства педагогов практически неприемлемым возвращение к традиционным методам и организационным формам работы. Школа как целостный организм начинает ощущать себя живущей в условиях «информационного общества».

Успех, как правило, связывается не столько с использованием той или иной информационной технологии или педагогической техники, сколько в том, что учителя и учащиеся вместе с родителями фактически сами заново создают свою школу. И роль компьютерной коммуникации как интегрирующего средства, обеспечивающего практическую реализацию этого процесса, оценивается участниками достаточно высоко. Ясно, что конкретный ход во многом зависит от специфических условий конкретной школы и ее окружения, и появление единой «модели» или переносимого образца для таких изменений практически невозможно.

Если этот переход удастся, изменения учебной среды можно констатировать по тому, что профессиональная позиция педагогов заметно меняется. Из носителя готовых знаний и способов работы учитель превращается в руководителя, посредника и помощника учащихся в процессе их совместной творческой работы. Он становится «архитектором учебной среды», того учебного контекста, в рамках которого школьники тратят разумное время на обсуждение и решение стоящих перед ними

творческих задач, воодушевляемые и поощряемые своими друзьями и наставниками. Достигается необходимый баланс между передачей готовых знаний и процессом самостоятельного добывания знаний учащимися. Учебный класс превращается в место, где учащиеся являются активными субъектами деятельности, занятыми решением значимых для них и посильных задач.

Происходит перенос акцента с изучения фактов на изучение способов действия. Конечно, никто не оспаривает важность и необходимость знания конкретных фактов. Однако упор начинает делаться на установление связей между ними. Помимо фактов школьники осознанно овладевают системой ориентировок и операций, которые могут быть сформированы только самим обучаемым в ходе интенсивной практической работы и являющихся его личным достоянием. Программные средства массового применения, обучающие программы, компьютерные среды, компьютерная коммуникация начинают выступать как взаимозависимые средства для построения учебного процесса, а компьютер превращается в обычный рабочий инструмент школьника, каким сегодня является книга, тетрадка или карандаш.

Работа по практическому воплощению в жизнь подобных изменений в школе становится сегодня в нашей стране одним из ключевых условий успеха начавшихся преобразований.

## Литература

1. *Machlup F.* The Production and Distribution of Knowledge in the United States. Princeton, New Jersey: Princeton University Press, 1962.
2. *Bell D.* The Coming Postindustrial Society. New York: Basic Books, 1973.
3. *Porat M.* The Information Economy. Washington, D.C.: U.S. Department of Commerce, 1977.
4. *Toffler A.* The Third Wave. New York: Harper and Row, 1978.
5. *Naisbit J.* Megatrends. New York: Warent Books, 1982.
6. *Zuboff S.* In the Age of the Smart Machine. New York: Basic Books, 1988.
7. *Громов Г.* Национальные информационные ресурсы: проблемы промышленной эксплуатации. М.: Наука, 1984.
8. *Workforce 2000: Work and Workers for the Twenty-first Century.* Hudson Institute, Indianapolis, Indiana, HI-3789-RR, June 1987.
9. *Solzhenitsyn A.* A World Split Apart. New York: Harper and Row, 1978.
10. *Roszak T.* The Cult of Information. New York: Pantheon Book, 1986.
11. *Blumenthal W.* The World Economy and Technological Change // Foreign Affairs. 1987/88. Vol. 66. № 3. P. 529—550.
12. *White M. A Curriculum for the Information Age* // Warger C. (ed.). Technology in

Today's Schools. ASCD Stock № 611—90085. 1990. P. 5—13.

13. *Уваров А.* Перестройка образования и информатизация общества. // Прогнозное социальное проектирование: Методологические и методические проблемы. М.: Наука, 1989. С. 222—236.

14. *Шульман К.* Проект «Пилотные школы»: год спустя. Информатика и образование. 1993. № 1. С. 51—54.

15. *Uvarov A., Prussakova A.* New York State/Moscow School Telecommunication Project: First Year of Experiences // A. McDougl and C. Dowling (ed.). Computers in Education. Amsterdam: Elsevier Science Publisher, 1990. P. 1027—1030.

16. *Уваров А., Никитин Н.* Компьютерные курсы для обучения работе с профессиональным текстовым редактором, электронными таблицами, презентаторами, СУБД и операционной системой // Применение новых компьютерных технологий в образовании: Материалы IV международной конференции, 24—26 июня 1993 г. Троицк: изд. Фонд «Байтик», 1993. С. 116—117.

17. *Керр С.* Новые информационные технологии и реформа школы // Информатика и образование. 1993. № 5. С. 117—123.



Л. Ф. Штернберг

## ЭВМ ПЕРЕБИРАЕТ ВАРИАНТЫ, ИЛИ ОБ ОДНОЙ ОХОТНИЧЬЕЙ ИСТОРИИ

«В течение нескольких секунд мощная ЭВМ проанализировала огромное количество возможных вариантов и выдала наилучший» — подобную фразу нередко можно увидеть в газете или услышать по радио, когда идет рассказ о широких возможностях компьютера. Но программист понимает, что для того, чтобы перебирать варианты, ЭВМ должна иметь соответствующую программу.

Если нужно просмотреть таблицу и выбрать наибольший (наименьший, наилучший и т. д.) элемент, то это делается простым циклом, и такую программу напишет любой школьник. Если надо перебрать все возможные сочетания пар элементов, один из которых берется из одной таблицы, а другой — из второй, то придется сделать вложенный цикл:

```
... таб A[1:M] ... таб B[1:N]
нц для i от 1 до M
  нц для j от 1 до N
  обработка пары (A[i],B[j])
кц;
кц;
```

А если требуется просмотреть все сочетания из 10 элементов — 10 вложенных циклов? А если из 100? А если число элементов заранее неизвестно и вообще различно в разных вариантах — что делать тогда? Пусть, например, надо найти «плечо локомотивной бригады», т. е. определить пункты, до которых бригада машинистов может довести состав и успеть вернуться назад за одну рабочую смену. Как организовать циклы перебора возможных станций? И сколько их должно быть? Простым вложением циклов здесь не поможешь, явно нужна какая-то другая идея.

Итак, перед нами элемент высшего программистского пилотажа — задача перебора. И оказывается, что для ее решения достаточно всего лишь двух циклов и одного магазина. Только магазин нам потребуется не тот, в котором что-то покупают, а тот, который имеется у автоматического оружия, точнее, его программный аналог. Особенность такого магазина в том, что патрон в него можно положить только с одного конца (через прорезь сверху), при этом уже лежащие там патроны проталкиваются вглубь. И достать патрон можно только через эту прорезь, после чего в нее подается новый патрон из глубины магазина. Нетрудно видеть, что доставать патроны можно только в порядке, обратном тому, в котором их клали, — это и есть нужная особенность. Программисты называют такой магазин стеком (от английского stack — «пачка»).

Программный аналог магазина — это массив и переменная-индекс, показывающая, до каких пор этот массив заполнен:

.... таб МАГ[1:k], цел  $im$

Значение  $im$  может показывать первую свободную или последнюю занятую ячейку магазина. Запись значения  $A$  в магазин производится так (см. рис. 1):

$МАГ[im]:=A; im:=im+1$  .

а извлечение — так:

$im:=im-1; A:=МАГ[im];$  .

«Вооружившись» магазином, попробуем разобрать простейшую переборную задачу. Предположим, что егерю Васе Гулякину поставлена задача просмотреть все  $N$ -дневные (возьмем  $N=3$ ) охотничьи маршруты, начинающиеся от лесничества (рис. 2): проверить проходимость троп, оборудовать засады, т. е., попросту говоря, пройти по очереди все маршруты. Проходить их можно в любом порядке, и Вася, не мудрствуя лукаво, всегда начинает с левой

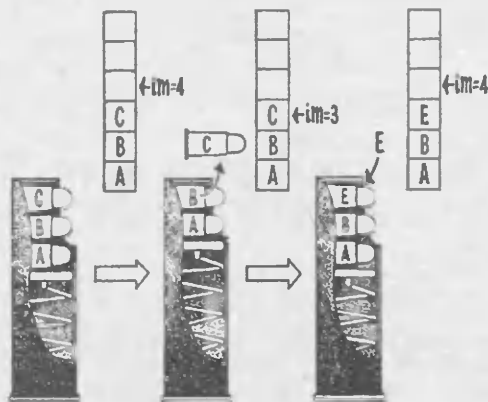


Рис. 1

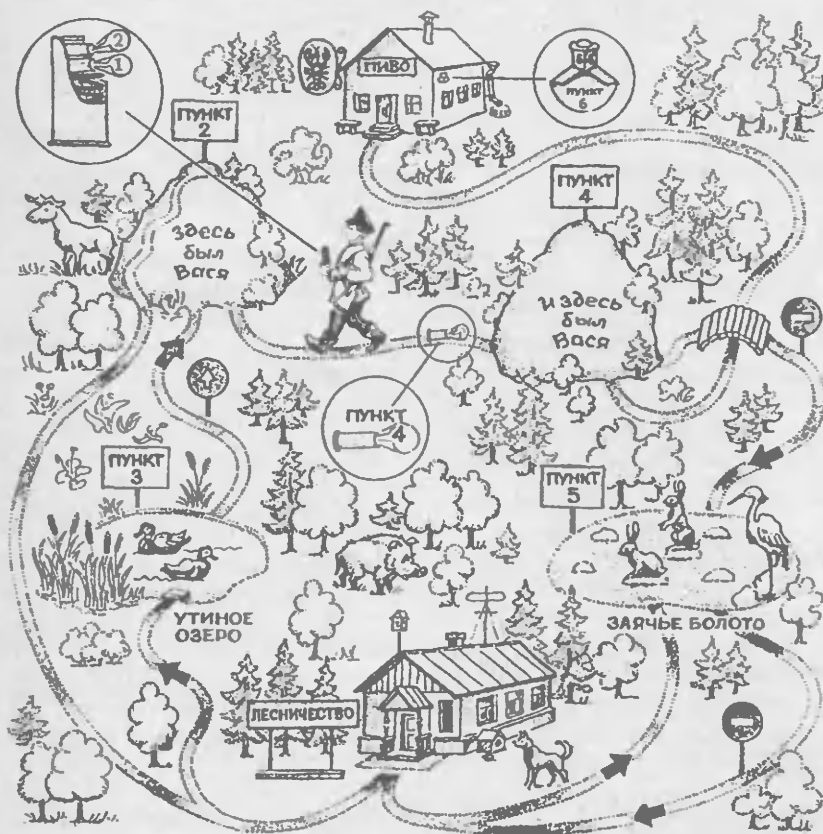


Рис. 2

дороги. Первый маршрут очевиден: пункты 1—2—4—6. Но на дороге были развилки. Значит, надо вернуться и пройти по развилкам. Из пункта 2 можно уйти еще и в 5, а откуда мы приходили в 5: из 1, 2 или 3? И сколько дней в пути: если шли через 1—2—5, то 2 дня, а если 1—5, то один... Все, Вася запутался окончательно (и мы тоже). Конечно, маршруты записаны, но в совершенном беспорядке. Найти уже пройденный и определить следующий довольно сложно.

И тут Вася вспомнил, что у него есть запасной магазин, и решил,

- 1 июня *В путь.*  
 2 июня *Залез на гору 2.*  
 3 июня *Залез на гору 4. Ну и гора!*  
 4 июня *Ура! 1-й маршрут пройден: 1-2-4-5.*  
 5 июня *Возвращаюсь в 4. Завтра идти в 6.*  
 6 июня *Ничего не помню...*  
 7 июня *Проснулся поздно, голова трещит. Можно записать: 1-2-4-6.*  
 8 июня *Выкинул гильзу и вернулся в 2.*  
 9 июня *Пришел вчера из 4. Сегодня иду в 5.*  
 10 июня *В 1 можно не идти. Но запишем кольцевой маршрут: 1-2-5-1.*  
 11 июня *В 4 не иду, а просто записываю: 1-2-5-4.*  
 ...  
 27 июня *Прошел маршрут 1-5-1-5. Все же, прежде чем ходить, не вредно лишний раз посмотреть в магазин.*  
 ...  
*Маршрут 1-5-4-6.*  
*Возвращаюсь.*  
*Возвращаюсь!*  
*Возвращаюсь!!*  
*Возв... Больше некуда! Ура!! Какое сегодня число?*

Рис. 3. Дневник Васи

что, уходя из какого-либо пункта, он запишет его номер на бумажке, бумажку положит в гильзу, а гильзу — в магазин. Тогда в нем будет накапливаться маршрут, а количество гильз будет определять длину этого маршрута. Для удобства (чтобы не забыть, где родное лесничество) на дно магазина была положена красная гильза с записью «1». На третий день Вася добрался до пункта 5 и записал в дневник первый маршрут: 1—2—4—5 (рис. 3). Вспоминать утром, откуда он пришел в 5, не потребовалось: в прорези магазина была гильза с запиской «пункт 4», — значит, надо возвращаться в пункт 4. Обработав маршрут 1—2—4—6 и опять вернувшись в 4, Вася обнаружил, что оттуда больше идти некуда. Надо возвращаться еще назад, но куда? Тогда он выбросил ненужную больше гильзу из магазина (вот она на рис. 2 валяется у пункта 4) и в прорези магазина увидел следующую гильзу с запиской: «пункт 2». Вернувшись в пункт 2, он пошел от развилки правее — в пункт 5 и изучил маршруты: 1—2—5—1, 1—2—5—6. Обойдя все маршруты вида 1—2—..., он выбросил из магазина и гильзу с запиской «пункт 2», вернулся в пункт 1, прошел по тропинке, идущей правее: 1—3—2—4, 1—3—2—5, 1—3—5—1 и т. д., по самой правой: 1—5—..., и наконец, исчерпал все варианты выхода из пункта 1. Теперь вынимаем красную гильзу: магазин пуст, а значит, все маршруты исчерпаны. И перебраны они оказались в лексико-графическом порядке (т. е. упорядочены по возрастанию цифр).

Запишем алгоритм, которым пользовался Вася:

предстартовая подготовка

отметить 1-й вариант (левую дорогу) в стартовой точке

нц пока магазин не пуст

отметить очередной вариант пути (дорогу, по которой еще не шли)

если дороги нет

то выбросить гильзу из магазина;

вернуться в пункт, записанный в прорези магазина

иначе если маршрут завершен

то обработать маршрут (записать его)

иначе положить в магазин гильзу с номером пункта,  
из которого ушли; пройти до нового пункта;  
отметить в этой точке 1-ю (левую) дорогу

все

все

кц

Но это алгоритм, понятный Васе, а как его записать для ЭВМ? Сеть дорог с точки зрения математики представляет собой граф, т. е. множество из  $M$  точек, соединенных дугами (или ребрами), который задается матрицей смежности размером  $M \times M$ . Элемент матрицы  $C_{ij}$  равен 1, если из точки  $i$  в  $j$  есть путь (т. е. дуга), и 0, если пути нет. Граф может быть ориентированным (когда по ребру можно идти только в одном направлении) и неориентированным (ребро проходимо в обоих направлениях). В последнем случае матрица получается симметричной, т. е.  $C_{ij}=C_{ji}$  (это все равно что соединить вершины парой направленных дуг: одна туда, другая обратно). Если считать, что по Васиным тропам по каким-то причинам можно идти только в направлениях, показанных стрелками, то эта сеть задается графом:

$$C = \begin{vmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix}$$

Предстартовая подготовка заключается в том, чтобы положить в магазин красную гильзу. Строго говоря, ее можно и не класть (стартовую точку и так несложно запомнить), но с ней удобнее программировать, ибо иначе возврат в стартовую точку придется выполнять не так, как в другие. Поэтому в магазине предусмотрим «нулевую» ячейку для красной гильзы. С выбором дорог будем поступать так: поскольку, какая из них левее, по матрице не определишь, то просто будем перебирать пункты по возрастанию их номеров (если, конечно, между ними есть дорога), т. е. более левой будем считать дорогу в пункт с меньшим номером. Переменная  $Np$  будет определять номер пункта, в который мы хотим идти. В качестве отметки первого варианта будем  $Np$  приравнивать нулю, т. е. покажем, что еще не проверяли пути ни в один пункт. Проверка на завершение маршрута проста: если в магазине есть  $N$  элементов, то маршрут пройден. В итоге получаем программу:

```

МАГ[0]:=номер стартовой точки; im:=1;
Np:=0;           {еще ни один пункт не пройден}
нц пока im>0    {im=0 <- магазин пуст}
  нц пока Np<M и C[МАГ[im],Np]=0  {поиск нового пути}
    Np:=Np+1
  кц;
если Np>M      {больше путей из этой точки нет}
  то im:=im-1; Np:=МАГ[im]      {вернулись назад}
  иначе если im=N      {маршрут завершен}
    то печать (МАГ[0],...,МАГ[im-1],Np)
    иначе МАГ[im]:=Np; im:=im+1 {положили в магазин
      гильзу с номером пункта, из которого уходим}
      Np:=0; {отсюда еще ничего не проходили}
все
все
кц

```

Отметим, что внутренний цикл пока завершает свою работу в двух случаях:

а)  $Np > M$ , т. е. просмотрели все дороги и ничего не нашли;

б)  $C[МАГ[im], Нп]=1$ , т. е. из пункта, который мы записали в магазин как покидаемый, в пункт  $N:=Нп$  есть путь.

Во втором случае мы идем в этот пункт и тут же записываем  $Нп$  в магазин, так как теперь мы собрались покинуть его. Только последний пункт маршрута нет смысла записывать в магазин, так как его оттуда придется тут же извлечь. Таким образом, при печати всех ранних пунктов маршрута их надо брать из магазина, а последний записан в  $Нп$ . При возврате в какой-либо пункт из магазина выбрасывается гильза, а ее содержимое попадает в  $Нп$ , и можно проверять пути в пункты с большими номерами.

Теперь попробуем усложнить задачу. Давайте запретим кольцевые и самопересекающиеся маршруты, т. е. маршрут не должен дважды проходить через один и тот же пункт. Что изменится в алгоритме? Почти ничего. Усложнится только внутренний цикл: если раньше он искал любой пункт, в который есть дорога из текущей точки, то теперь нужно проверить, не было ли ее уже на нашем пути, т. е. просмотреть содержимое магазина.

**Задача для самостоятельного решения.** Напишите программу, дающую все маршруты длины  $N$  без самопересечений, исходящие из данной точки.

Изменим условия: пусть нам нужны маршруты не заданной длины, а все приводящие из пункта  $A$  в пункт  $B$ . Что меняется в алгоритме? Только проверка на конец маршрута, она теперь выглядит так:

если  $Нп=B$  то ...

Но не всегда нам нужны все маршруты. Пусть мы хотим найти кратчайший путь из  $A$  в  $B$ . Что меняется? Опять немного: только блок обработки. Теперь надо первый маршрут запомнить в отдельной таблице, и если очередной путь оказался короче, то перезапомнить его. (Здесь, правда, потребуются дополнительные присваивания в предстартовой подготовке.)

**Задача для самостоятельного решения.** Напишите полностью программу поиска кратчайшего маршрута.

До сих пор мы работали с графом, который напоминает скорее не дорожную, а телеграфную сеть: переход из пункта в пункт происходит всегда за одно и то же время. Для задания дорожной сети придется несколько изменить матрицу  $C$ : теперь ее элементы будут принимать любые положительные значения, задающие длину или время пути. Длина пути должна подсчитываться суммированием весов дуг (т. е. числовых значений  $C_{ij}$ ) при проходе в очередной пункт и вычитанием — при возврате.

**Задача для самостоятельного решения.** Дан взвешенный граф (на дугах которого заданы числа-веса), описывающий дорожную сеть. Веса — время прохождения дуги. В пункте  $A$  находится локомотивное депо. Определить пункты, в которых надо построить гостиницы для машинистов, если бригада локомотива должна быть в пути не более  $T$  часов.

В переборных задачах магазин иногда состоит из нескольких массивов, а часто нужны и дополнительные массивы для разных целей. Использование дополнительных массивов мы видели в задаче поиска минимального пути. Пример же двухмассивного магазина дает такая задача: пусть взвешенный граф задает трубопроводную сеть, где числа на дугах определяют толщину труб (пропускную способность). «Толщина маршрута» определяется минимальной толщиной входящих в него дуг. Пусть надо найти самый «толстый» маршрут из  $A$  в  $B$ . Здесь удобно взять магазин из двух массивов —  $МАГ$  и  $ТОЛ$ , где элемент массива  $ТОЛ$  показывает текущую «толщину маршрута». Запись в такой магазин выглядит так:

$МАГ[im]:=Нп; ТОЛ[im]:=\min(ТОЛ[im], C[МАГ[im-1], Нп]); im:=im+1;$

(если дуга из пункта  $C[МАГ[im-1]$  в пункт  $Нп$  более тонкая, то она уменьшает текущую «толщину маршрута»), а процедура выбрасывания не меняется. «Прорезь» магазина теперь состоит из двух значений, где  $ТОЛ[im-1]$  показывает текущую «толщину маршрута».

**Задача для самостоятельного решения.** Написать программу поиска самого толстого пути.

## Уважаемые читатели!

Если вы хотите и дальше получать и читать наши журналы, напоминаем, что с 1 сентября началась подписка на периодические издания на первое полугодие 1995 года.

Редакция, как и в прошлом году, гарантирует нашим подписчикам выпуск всех номеров журнала «Информатика и образование» и его приложений.

Обращаем ваше внимание, что

### ЦЕНЫ НА ВСЕ НАШИ ИЗДАНИЯ ОСТАЛИСЬ НА ПРЕЖНЕМ УРОВНЕ!

Напоминаем сведения, необходимые для подписки:

Название журнала	Подписной индекс	Периодичность	Стоимость подписки за полугодие по каталогу «Роспечати»
«Информатика и образование»	70423 для индивидуальных подписчиков	1 раз в 2 мес.	18 000 руб.
	73176 для предприятий и организаций		45 000 руб.
Библиотека журнала «Информатика и образование»			
«Персональный компьютер БК-0010 - БК-0011М»	73177 для индивидуальных подписчиков	1 раз в 2 мес.	12 000 руб.
	73092 для предприятий и организаций		24 000 руб.
«Персональный компьютер УКНЦ»	73179 для индивидуальных подписчиков	1 раз в 3 мес.	10 000 руб.
	73093 для предприятий и организаций		20 000 руб.



Приглашаем наших читателей на выставку «Пресса-95», которая будет проходить с 26 по 30 октября 1994 года в Москве в 4-м павильоне Всероссийского выставочного центра (бывшей ВДНХ). Организаторы выставки – Агентство «Роспечать», ИТАР-ТАСС, общество «Информатизация России». На выставке вы сможете пообщаться с сотрудниками редакции, высказать свои пожелания по содержанию и оформлению наших изданий, приобрести некоторые из них. Ждем вас на стенде 3.5.

Е. А. Ерохина,  
Москва

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

### 16. Рекурсия, степень, факториал и НОД на Прологе

Рекуррентное определение факториала

$$\begin{cases} 0! = 1 \\ n! = (n-1)! \cdot n, n - \text{натуральное} \end{cases} \xrightarrow{n} \boxed{\phantom{0! = 1}} \xrightarrow{k=n!}$$

легко переводится на Пролог. Назовем правило «факториал» и расположим параметры так:  
**факториал(n,k)**

Сначала записывается факт «0!=1»:

**факториал(0,1);**

Затем записываем правило для определения факториала любого другого числа:

**факториал(n,k) <- БОЛЬШЕ(n,0), ВЫЧИТАНИЕ(n,1,t),  
факториал(t,s), УМНОЖИТЬ(s,n,k);**

Получилось какое-то необычное правило: факториал числа определяется через факториал меньшего числа. Интересно, а как же машина применяет это правило? Если мы зададим вопрос:

**?факториал(0,x);**

то для нахождения ответа машине достаточно найти факт:

**факториал(0,1);**

идентифицировать x и число 1 и дать ответ:

**x=1**

Если же соответствующего факта нет, то машина применяет правило. Например, зададим машине вопрос:

**?факториал(3,y);**

Процесс поиска ответа будет таким:

$n=3, n>0$  — «истина»,  $t = n-1 = 2$ , и машине надо найти  $s=t!$ .

Здесь машина не может применять это правило дальше, так как неизвестно, чему равно  $s$ . Тогда машина приостанавливает выполнение поиска и создает новую (вторую) «копию» правила «факториал» для вычисления  $2!$ .

В этой копии  $n=2, n>0$  — «истина»,  $t = n-1 = 1$ , и машине надо найти  $s=t!$  уже для новой копии. Затем машине понадобится еще одна копия, в которой  $n=1, n>0$  — «истина»,  $t = n-1 = 0$ , и машине вновь надо найти  $s=t!$ .

Но для нахождения  $0!$  в базе есть факт:  $0!=1$ . Итак, в последней копии  $s=0!=1$  и  $k=s \cdot n=1$ .

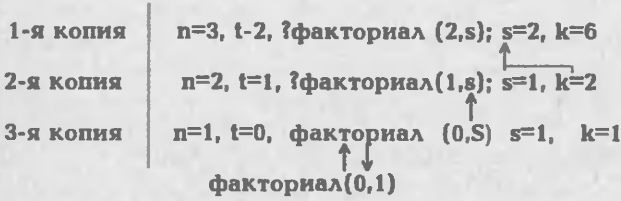
Теперь начинается процесс, называемый «обратным ходом»: машина начинает подставлять полученные ответы в предыдущие копии. Для второй копии  $s=1$ , и, следовательно,  $k=2$ . И наконец, в первой копии  $s=2, k=6$ , и машина даст ответ:

**y=6.**

\* Продолжение. Начало см.: Информатика и образование. 1993. № 5.

Все проведенные выше рассуждения можно представить на схеме, в которой приведены значения переменных во всех копиях правила.

**!факториал(3,y);**



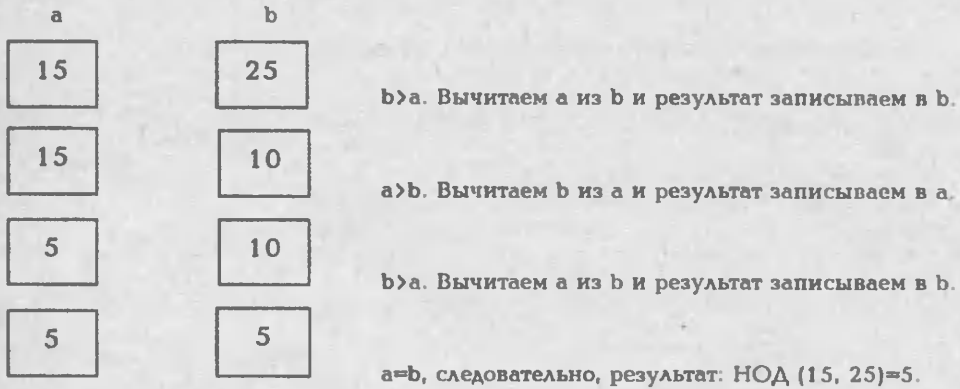
Стрелками показаны подстановки вычисленных значений при обратном ходе.

Теперь рассмотрим широко известную задачу вычисления наибольшего общего делителя двух натуральных чисел. Для этого используется самый древний из известных человечеству алгоритмов — алгоритм Евклида. Напомним его суть.

**АЛГОРИТМ ЕВКЛИДА**

Пусть нам даны числа a и b (натуральные, a ≠ b). Мы ищем наибольшее из этих чисел и вычитаем из него второе число, a результат записываем на место большего числа. Затем сравниваем эти числа и, если они не равны, снова вычитаем меньшее из большего. Процесс завершается тогда, когда числа станут равны. Полученное число и есть НОД.

*Пример:* Пусть b=25, a=15. Изобразим все значения, которые будут принимать переменные:



Каждый шаг процесса рекурсивного вычисления НОД можно описать так:

$$\text{НОД}(a,b) = \begin{cases} \text{НОД}(a-b,b), & \text{при } a > b \\ \text{НОД}(a,b-a), & \text{при } a < b \\ a, & \text{при } a = b \end{cases}$$

$\begin{matrix} \xrightarrow{a} \\ \xrightarrow{b} \end{matrix}$  $\xrightarrow{c}$

Справа изображен «черный ящик» для программы вычисления НОД.

Сама же программа будет такой:  
**НОД(a,b,c) <- БОЛЬШЕ(a,b), ВЫЧИТАНИЕ(a,b,t), НОД(t,b,c);**  
**НОД(a,b,c) <- МЕНЬШЕ(a,b), ВЫЧИТАНИЕ(b,a,t), НОД(a,t,c);**  
**НОД(a,a,a);**

Теперь можно задавать вопросы, например:



По-русски:	На Прологе:	Ответ ЭВМ:
Каков НОД 15 и 25 ?	?НОД(15,25,a);	a=5
Каков НОД 5 и 5 ?	?НОД(5,5,a);	a=5
Каков НОД 13 и 5 ?	?НОД(13,5,a);	a=1

Но если задать машине вопрос: ?НОД(1,5000);, то ответ получен не будет, так как машине просто не хватит памяти для того, чтобы сохранить все копии этого правила. Но мы можем усовершенствовать алгоритм, если вместо разности будем использовать остаток от деления (т. е. сведем процесс многократного вычитания к делению с остатком). Тогда наш алгоритм приобретет следующий вид:

$$\text{НОД}(a,b) = \begin{cases} \text{НОД}(\text{ост}(a/b), b), & \text{при } a > b \\ \text{НОД}(a, \text{ост}(b/a)), & \text{при } a < b \\ a & , \text{при } a = b \\ b & , \text{при } a = 0, b < > 0 \\ a & , \text{при } a < > 0, b = 0 \end{cases}$$

a	b
15	25
15	10
5	10
0	5

Ответ: НОД(15,25)=5

Программа на Прологе, использующая «усовершенствованный» алгоритм, будет выглядеть так:

```
НОД(a,b,c)<-БОЛЬШЕ(a,b),БОЛЬШЕ(b,0),остаток(a,b,t),НОД(t,b,c);
НОД(a,b,c)<-МЕНЬШЕ(a,b),БОЛЬШЕ(a,0),остаток(b,a,t),НОД(a,t,c);
НОД(a,a,a);
НОД(0,a,a);
НОД(a,0,a);
```

Эта программа может справиться с любым вопросом, в том числе и с ?НОД(1,5000);

и при этом она находит ответ гораздо быстрее предыдущей.

### ЗАДАЧИ И УПРАЖНЕНИЯ

1. Написать программу на Прологе для вычисления натуральной степени числа (правило будет иметь вид *степень(основание, показатель, степень)*;) и схематически изобразить поиск ответа на вопросы:

а) ?степень(0,1,a); б) ?степень(3,2,b); в) ?степень(2,3,c).

2. Написать правило «остаток», которое используется во втором варианте программы НОД.

## 17. Пусть машина посчитает...

Далее все программы мы будем писать так, чтобы машина «подсказывала» нам, какие данные вводить, и что получено в качестве результата. Для этого мы будем использовать правило «работа» для ввода данных и вывода результатов. Например, программа поиска НОД с правилом «работа» будет выглядеть так:

```
%нод;
вычитание(a,b,c)<-СЛОЖЕНИЕ(c,b,a);
нод(a,b,c)<-БОЛЬШЕ(a,b),вычитание(a,b,t),ВЫВОД(1),нод(t,b,c);
```

```

нод(a,b,c)<-МЕНЬШЕ(a,b),вычитание(b,a,t),ВЫВОД(2),нод(a,t,c);
нод(a,a,a);
работа<-ВЫВОД("Введите число А:"),ВВОДЦЕЛ(a),ПС,
    ВЫВОД("Теперь введите число В:"),ВВОДЦЕЛ(b),
    нод(a,b,c),ПС,ВЫВОД("Наибольший общий делитель равен ",c);
Для того чтобы работать с этой программой, нужно задать вопрос
?работа;

```

В этом разделе мы рассмотрим решение традиционных для курса программирования вычислительных задач.

**Задача 1.** Подсчитать сумму квадратов четных чисел от 2 до  $p$ .

*Решение:*

```

вычитание(a,b,c)<-СЛОЖЕНИЕ(c,b,a);
сум2(2,4);
чет(a)<-УМНОЖЕНИЕ(2,b,0,a);
сум2(p,s)<-БОЛЬШЕ(p,2),чет(p),вычитание(p,2,t),сум2(t,c),
    УМНОЖЕНИЕ(p,p,0,p2'),СЛОЖЕНИЕ(c,p2',s);
сум2(p,s)<-БОЛЬШЕ(p,2),НЕ(чет(p)),вычитание(p,1,k),сум2(k,s);
работа<-ВЫВОД("Введите P,большее двух:"),ВВОДЦЕЛ(p),ПС,сум2(p,s),
    ВЫВОД("Сумма квадратов четных чисел от 2 до P равна ",s);

```

В этой программе рассматривается два случая:

1.  $P$  — четное число. В этом случае (по первому правилу) рекурсивно считается сумма четных чисел, меньших этого числа, и к ней прибавляется само это число.
2.  $P$  — нечетное. В этом случае (по второму правилу) из этого числа вычитается единица (разность будет четной) и считается сумма четных чисел до разности.

**Задача 2.** Подсчитать сумму цифр введенного числа.

*Решение:*

```

суммацифр(x,x)<-МЕНЬШЕ(x,10);
суммацифр(x,s)<-БОЛЬШЕ(x,9),УМНОЖЕНИЕ(10,y,e,x),МЕНЬШЕ(e,10),
    суммацифр(y,c),СЛОЖЕНИЕ(c,e,s);
работа<-ВЫВОД("Введите число X:"),ВВОДЦЕЛ(x),ПС,суммацифр(x,s),
    ВЫВОД("Сумма цифр вашего числа равна "),ВЫВОД(s);

```

Если число меньше 9, то сумма цифр равна этому числу, а в противном случае это число делится на 10 (с отбрасыванием остатка), рекурсивно вычисляется сумма цифр частного, а затем к ней прибавляется число единиц.

**Задача 3.** Подсчет суммы числового ряда. Все задачи этого типа решаются по одному и тому же алгоритму. Например, рассмотрим подсчет суммы ряда:

$$\sum_{i=1}^n a^i + 2 * i^2$$

Сначала описывается одно слагаемое (с номером  $i$ ):

% $i$ -е слагаемое;

```

формула(a,i,f)<-степень(a,i,s),УМНОЖЕНИЕ(i,i,0,i2'),
    УМНОЖЕНИЕ(2,i2',s,f);

```

%степень;

```

вычитание(a,b,c)<-СЛОЖЕНИЕ(b,c,a);

```

```

степень(a,0,1);

```

```

степень(a,n,s)<-БОЛЬШЕ(n,0),вычитание(n,1,k),степень(a,k,p),
    УМНОЖЕНИЕ(p,a,0,s);

```

Затем записывается правило для рекурсивного суммирования:

```

сум(a,1,s)<-формула(a,1,s);

```

```

сум(a,n,s)<-БОЛЬШЕ(n,1),вычитание(n,1,m),сум(a,m,c),формула(a,n,k),

```

## СЛОЖЕНИЕ(k,c,s);

И наконец, записывается правило «работа»:

```
работа<-ВЫВОД("Введите a:"),ВВОДЦЕЛ(a),ПС,ВЫВОД("Введите b:"),
ВВОДЦЕЛ(b),ПС,сум(a,n,s),ВЫВОД("Сумма равна "),ВЫВОД(s);
```

**Задача 4.** Написать программу на Прологе, переводящую заданное число из десятичной системы счисления в двоичную.

*Решение:*

Так как диапазон чисел в Прологе ограничен, для записи результата будем использовать строку.

```
%перевод в двоичную систему;
деление(a,b,c,ost')<-УМНОЖЕНИЕ(b,c,ost',a),МЕНЬШЕ(ost',b);
перевод(0,"0");
перевод(1,"1");
перевод(x,y)<-БОЛЬШЕ(x,1),деление(x,2,c,ost'),перевод(c,z),
СТРЦЕЛ(ost2',ost'),СЦЕП(z,ost2',y);
работа<-ВЫВОД("Введите число X:"),ВВОДЦЕЛ(x),ПС,перевод(x,y),
ВЫВОД("Ваше число в двоичной системе равно "),ВЫВОД(y);
```

## ЗАДАЧИ

1. Написать программу, проверяющую, четные ли все цифры данного числа.

2. Подсчитать  $N$ -й член ряда Фибоначчи.

Числа Фибоначчи вычисляются по формуле:

$$\begin{cases} F(1) = F(2) = 1; \\ F(n) = F(n-1) + F(n-2), n > 2 \end{cases}$$

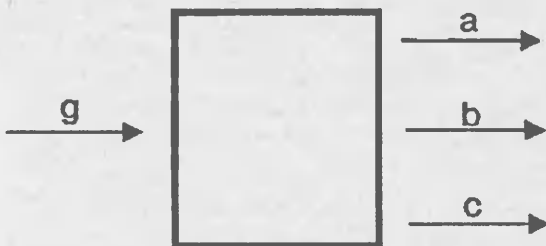
3. Подсчитать сумму  $N$  членов ряда Фибоначчи и установить, почему при больших  $N$  машина дает неправильный ответ.

4. Написать программу на Прологе для перевода заданного числа из десятичной системы счисления в систему счисления с заданным основанием (меньшим десяти).

5. Написать программу на Прологе, проверяющую, верно ли, что квадрат числа равен кубу суммы его цифр.

## 18. «Не прячьте ваши денежки...» или обмен денежных сумм

Пусть в наличии имеются купюры по 1, 3 и 5 рублям. Требуется выдать указанную сумму денег (самыми крупными купюрами). Изобразим «черный ящик».



где  $g$  — выдаваемая сумма,

$a$  — количество выдаваемых купюр по 1 рублю,

$b$  — количество выдаваемых купюр по 3 рубля,

$c$  — количество выдаваемых купюр по 5 рублей,

```
%размен денег;
```

```
% если денег нет, то и менять нечего
```

```
размен(0,0,0,0);
```

```
% если сумма денег равна достоинству некоторой купюры, то ее и выдаем
```

```
размен(1,0,0,1)<-ВЫВОД("1 по 1 рублю");
```

```
размен(3,0,1,0)<-ВЫВОД("1 по 3 рубля");
```

```
размен(5,1,0,0)<-ВЫВОД("1 по 5 рублей");
```

% эти правила вычисляют количество купюр определенного достоинства,  
% а затем рекурсивно разменивают остаток

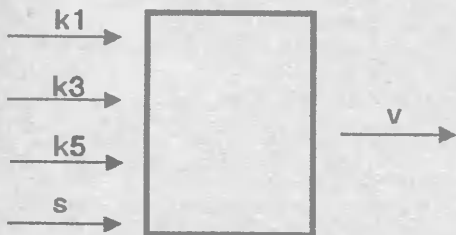
```
размен(g,a,b,c)<-БОЛЬШЕ(g,5),УМНОЖЕНИЕ(y,5,ost',g),МЕНЬШЕ(ost',5),
    ВЫВОД(y," по 5 рублей"),размен(ost',0,b,c);
размен(g,a,b,c)<-БОЛЬШЕ(g,3),УМНОЖЕНИЕ(y,3,ost',g),МЕНЬШЕ(ost',3),
    ВЫВОД(y," по 3 рубля"),размен(ost',0,0,c);
размен(g,a,b,c)<-БОЛЬШЕ(g,0),ВЫВОД(g," по 1 рублю");
```

```
работа<-ВЫВОД("Введите сумму:"),ВВОДЦЕЛ(g),ПС,размен(g,a,b,c),ПС;
```

ВОПРОС: Почему сумма выдается самыми крупными из имеющихся купюр?

ОТВЕТ: Машина применяет правила для размена в том порядке, в котором они записаны.

Чаще всего при размене денег количество купюр разного достоинства не бесконечно (ограничено). Поэтому требуется найти способ размена, при котором хватит купюр, имеющихся в наличии. Нарисуем «черный ящик» для этого случая.



s — денежная сумма,  
k1 — количество наличных купюр по 1 рублю,  
k3 — количество наличных купюр по 3 рубля,  
k5 — количество наличных купюр по 5 рублей,  
v — строка, в которой записывается порядок выдачи купюр.

При выдаче очередной купюры происходит следующее:

1. К списку выдаваемых денег добавляется очередная купюра.
2. Из количества соответствующих наличных купюр вычитается единица.
3. Из выдаваемой суммы вычитается достоинство выданной купюры.
4. Размен продолжается до тех пор, пока сумма не станет равной нулю или не выяснится, что дальнейший размен невозможен.

%размен дефицитных денег;

% если вся сумма выдана, размен кончается (базис)  
размен(.,.,.,0," ");

вычитание(a,b,c)<-СЛОЖЕНИЕ(b,c,a);

% выдача рубля

```
размен(k1',k3',k5',s,v)<-БОЛЬШЕ(s,0),БОЛЬШЕ(k1',0),вычитание(k1',1,k),
    вычитание(s,1,s1'),размен(k,k3',k5',s1',v1'),
    СЦЕП("1 руб +",v,v1');
```

% выдача трех рублей

```
размен(k1',k3',k5',s,v)<-БОЛЬШЕ(s,2),БОЛЬШЕ(k3',0),вычитание(k3',1,k),
    вычитание(s,3,s1'),размен(k1',k,k5',s1',v1'),
    СЦЕП("3 руб +",v,v1');
```

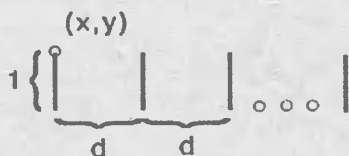
% выдача пяти рублей

```
размен(k1',k3',k5',s,v)<-БОЛЬШЕ(s,4),БОЛЬШЕ(k5',0),вычитание(k5',1,k),
    вычитание(s,5,s1'),размен(k1',k3',k,s1',v1'),
    СЦЕП("5 руб +",v,v1');
```

работа <- ВВЫВОД("Введите сумму:"), ВВОДЦЕЛ(s), ПС,  
 ВВЫВОД("Введите количество рублей:"), ВВОДЦЕЛ(k1'), ПС,  
 ВВЫВОД("Введите количество трешек:"), ВВОДЦЕЛ(k3'), ПС,  
 ВВЫВОД("Введите количество пятерок:"), ВВОДЦЕЛ(k5'), ПС,  
 размен(k1', k3', k5', s, v), ВВЫВОД("Получите", v);

## 19. Рекурсивные графические построения

В предыдущем параграфе мы решили некоторые математические задачи с использованием рекурсии. Но самое красивое применение рекурсии — рекурсивные графические построения. На Прологе удобно строить картинку с повторяющимися элементами. Например, нарисуем «забор»:



Всего в заборе  $n$  «палок» длины  $l$ .

Сначала напишем правило для рисования одной палки длиной  $l$  из точки с координатами  $(x, y)$  вниз ( $c$  — цвет палки):

палка( $x, y, l, c$ ) <- СЛОЖЕНИЕ( $y, l, z$ ), ЛИНИЯ( $x, y, x, z, c$ );

А теперь напишем общее правило для рисования забора произвольной длины. Вид забора зависит от числа палок, координат начала забора, длины палки и выбранного цвета; правило будет иметь вид:

забор( $n, x, y, l, c$ ).

Сначала нарисуем забор из одной-единственной палки:

забор( $1, x, y, l, c$ ) <- палка( $x, y, l, c$ );

Для того чтобы получить забор из большего числа палок, используем рекурсию: если число палок  $n > 1$ , то рисуем забор из  $n-1$  палки, а затем пририсовываем к ним еще одну. Координаты начала очередной палки будут высчитываться по формуле:

$$y_1 = y, \quad x_1 = x + d * (n-1)$$

(Это не что иное, как формула арифметической прогрессии!). Правило для «длинного» забора (из  $n$  палок) будет такое:

забор( $n, x, y, l, c$ ) <- БОЛЬШЕ( $n, 1$ ), ВЫЧИТАНИЕ( $n, 1, p$ ), забор( $p, x, y, l, c$ ),  
 УМНОЖИТЬ( $d, p, k$ ), СЛОЖЕНИЕ( $x, k, b$ ), палка( $b, y, l, c$ );

Аналогичным образом можно составить программу для рисования мишени из  $n$  белых концентрических окружностей. Базисом будет являться точка в центре окружности (а центр окружности расположен в центре экрана):

мишень(0) <- ТОЧКА(123, 86, 15);

Чтобы получить мишень, надо вывести формулу для радиуса окружности с номером  $i$ :

$$R_i = 10 * i$$

И тогда можно легко написать рекурсивное правило для мишени из  $n$  окружностей.

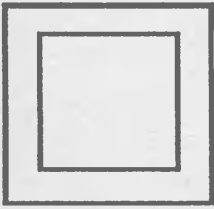
мишень( $n$ ) <- ВЫЧИТАНИЕ( $n, 1, i$ ), мишень( $i$ ), УМНОЖИТЬ(10,  $i, r$ ),  
 ОКРУЖНОСТЬ(123, 86,  $r, 15$ );

Примечание: вопросы к этим фактам и правилам должны содержать открытие графического режима (ЗАПИСЬ\_В("gr:")).

### ЗАДАЧИ

1. Нарисовать аллею из  $n$  деревьев. Расстояние между соседними деревьями равно 10, высота ствола — 15, радиус кроны — 10. Первое дерево имеет координаты «корня»  $x, y$ .

2. Нарисовать картинку:



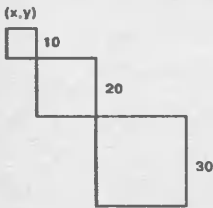
Длина стороны первого квадрата равна 10, а сторона каждого следующего на 20 больше.

Всего квадратов  $n$  штук.

Координаты левого верхнего угла самого маленького квадрата —  $x, y$ .

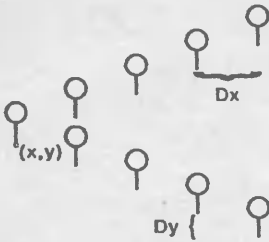
3. Нарисовать аллею «с перспективой»: первое дерево имеет координаты «корня»  $x, y$ , высоту ствола и радиус кроны 10. Каждое следующее дерево отстоит от предыдущего на  $15 \cdot n$  точек и имеет радиус кроны и высоту ствола  $10 \cdot n$ .

4. Нарисовать картинку. Квадратов должно быть  $n$  штук.

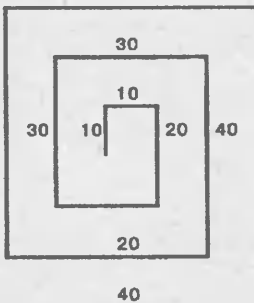


5. Нарисовать картинку. Радиус первой окружности — 10, а каждой следующей — на 5 больше.

6. Нарисовать «расходящуюся» аллею.



7. Нарисовать картинку.  
Спираль состоит из  $n$  витков.



## ОТВЕТЫ И РЕШЕНИЯ\*

## К разделу 11

1. остроуг(имя')<-кандидат(имя',a,b,c),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),  
МЕНЬШЕ(p,l);

2. Это следует из того, что для теоремы Пифагора обратная теорема также верна, т. е. если для трех чисел верно, что  $a^2 + b^2 = c^2$ , то они могут быть сторонами прямоугольного треугольника. Для тупоугольного треугольника из того, что  $c^2 > a^2 + b^2$ , не следует, что  $c < a+b$ .

3. Здесь надо доказать, что  $c^2 < a^2 + b^2$ , то  $c < a+b$ . Действительно, так как стороны треугольника — положительные числа, из того, что  $c \leq a+b$ , следует, что  $c^2 < (a+b)^2$ ,  $c^2 < a^2 + 2ab + b^2$ ,  $ab > 0$ , следовательно,  $a^2 + 2ab + b^2 > a^2 + b^2$ , т. е.  $c < a+b$  — условие более сильное, чем  $c^2 < a^2 + b^2$ .

4. равнобедренный(имя')<-кандидат(имя',a,b,c),РАВНО(a,b);  
равнобедренный(имя')<-кандидат(имя',a,b,c),РАВНО(a,c);  
равнобедренный(имя')<-кандидат(имя',a,b,c),РАВНО(b,c);  
правильный(имя') <-кандидат(имя',a,b,c),РАВНО(a,b),РАВНО(b,c);

5. Правило «кв\_стороны» не только рассматривает стороны треугольника в нужном сочетании, но и перебирает все возможные сочетания длин сторон. Например, если взят кандидат ABC со сторонами 3, 4 и 5, то это правило считает, что его стороны не только (3,4,5), но и (3,3,3), (3,4,4), (3,5,5) и т. д.

## К разделу 13

Рассмотрим программу рисования домика:

?ФОН(7,1);

вычитание(a,b,c)<-СЛОЖЕНИЕ(b,c,a);

%Домик с координатами нижнего левого угла x,y

дом(x,y,a,b,c)<-вычитание(y,50,y1'),СЛОЖЕНИЕ(x,100,x1'),

вычитание(y,130,yk'),СЛОЖЕНИЕ(x,50,xk'),

СЛОЖЕНИЕ(x,a,xo'),вычитание(y,b,yo'),

СЛОЖЕНИЕ(xo',40,xo2'),вычитание(yo',20,yo2'),

ЛИНИЯ(x,y,x,y1',c),ЛИНИЯ(x,y,x1',y,c),

ЛИНИЯ(x,y1',x1',y1',c),ЛИНИЯ(x1',y1',x,y1',c),

ЛИНИЯ(x1',y1',x1',y,c),ЛИНИЯ(x,y1',xk',yk',c),

ЛИНИЯ(xk',yk',x1',y1',c),ЛИНИЯ(xo',yo',xo2',yo',c),

ЛИНИЯ(xo',yo',xo',yo2',c),ЛИНИЯ(xo',yo2',xo2',yo2',c),

ЛИНИЯ(xo2',yo2',xo2',yo',c);

%?ЗАПИСЬ\_В ("grp:"),дом(60,100,10,10,15);

Дерево рисуется аналогично.

\* Разделы 11—15 опубликованы: Информатика и образование. 1993. № 5.

## К разделу 14

Ошибка в доказательстве «с кошками» состоит в том, что в нем неверно проведен индуктивный переход: взяв  $d$  в  $u$  кошек, мы не сможем доказать, что у них один и тот же цвет.

## ДОКАЗАТЕЛЬСТВА:

1. Б А З И С: При  $n=1$  утверждение верно:  $1+3+5+3=12$  — кратно 3.

И Н Д У К Т И В Н Ы Й П Е Р Е Х О Д:

ДАНО:  $k^3 + 3k^2 + 5k + 3$  кратно 3 при некотором натуральном  $k$ .

ДОКАЗАТЬ:  $(k+1)^3 + 3(k+1)^2 + 5(k+1) + 3$  кратно 3.

Доказательство:

$$\begin{aligned} (k+1)^3 + 3(k+1)^2 + 5(k+1) + 3 &= k^3 + 3k^2 + 3k + 1 + 3(k^2 + 2k + 1) + 5k + 5 + 3 = \\ &= k^3 + 6k^2 + 14k + 12 = (k^3 + 3k^2 + 5k + 3) + (3k^2 + 9k + 9) \end{aligned}$$

Сумма слагаемых в первых скобках кратна трем (дано), а кратность трем суммы во вторых скобках очевидна.

2. Б А З И С: При  $n=1$  утверждение верно:  $1=1^2$ .

И Н Д У К Т И В Н Ы Й П Е Р Е Х О Д:

ДАНО:  $1+3+\dots+(2k-1)=k^2$ .

ДОКАЗАТЬ:  $1+3+\dots+(2k-1)+(2k+1)=(k+1)^2$ .

Доказательство:

$$1+3+\dots+(2k-1)+(2k+1) = k^2 + 2k + 1 = (k+1)^2.$$

Остальные соотношения попробуйте доказать сами.

## К разделу 15

$$\begin{cases} 0! = 1 \\ n! = (n-1)! * n, \text{ при натуральных } n \end{cases}$$

## К разделу 16

1. Сначала надо вспомнить определение степени и изобразить «черный ящик»:

$$\begin{cases} 0 \\ a = 1 \\ n \quad n-1 \\ a = a \end{cases} * a, \text{ при натуральных } n \quad \begin{array}{c} \xrightarrow{a} \\ \xrightarrow{n} \end{array} \boxed{\phantom{a^n}} \xrightarrow{k=a^n}$$

Программа на Прологе получится такая:

```
степень(a,0,1);
степень(a,n,k) ← БОЛЬШЕ(n,0), ВЫЧИТАНИЕ(n,1,p), степень(a,p,t),
                УМНОЖЕНИЕ(t,a,k);
```

а) При ответе на этот вопрос машина просто применяет факт:

степень(a,0,1); следовательно,  $a = 1$ .

б) Поиск ответа изображен на схеме:





5.  
 %сумма цифр;  
 суммацифр(x,x)<-МЕНЬШЕ(x,10);  
 суммацифр(x,s)<-БОЛЬШЕ(x,9),УМНОЖЕНИЕ(10,y,e,x),МЕНЬШЕ(e,10),  
 суммацифр(y,c),СЛОЖЕНИЕ(c,e,s);

квадрат(x,y)<-УМНОЖЕНИЕ(x,x,0,y);  
 кубсч(a,b)<-суммацифр(a,s),УМНОЖЕНИЕ(s,s,0,f),УМНОЖЕНИЕ(s,f,0,b);

работа<-ВЫВОД("Введите число X:"),ВВОДЦЕЛ(x),ПС,суммацифр(x,s),  
 квадрат(x,y),кубсч(x,b),ВЫВОД("Сумма цифр вашего числа равна "),  
 ВЫВОД(s),ПС,ВЫВОД("Квадрат числа равен "),ВЫВОД(y),ПС,  
 ВЫВОД("Куб суммы цифр числа равен "),ВЫВОД(b),РАВНО(y,b),  
 ПС,ВЫВОД("Квадрат числа равен кубу суммы его цифр");

### К разделу 18

1. Сначала напишем правило для изображения одного дерева:

вычитание(a,b,c)<-СЛОЖЕНИЕ(b,c,a);  
 дерево(x,y)<-вычитание(y,10,z),ЛИНИЯ(x,y,x,z,6),вычитание(z,15,h),  
 ОКРУЖНОСТЬ(x,h,10);

Затем запишем базис (факт для аллеи из одного дерева) и рекурсивное правило для изображения n деревьев. Они аналогичны тем, что рассматривались в задаче изображения забора из n палочек.

аллея(x,y,1)<-дерево(x,y);  
 аллея(x,y,n)<-БОЛЬШЕ(n,1),ВЫЧИТАНИЕ(n,1,p),аллея(x,y,p),  
 УМНОЖЕНИЕ(10,p,0,k),СЛОЖЕНИЕ(x,k,b),дерево(b,y);

Для выполнения программы надо задать вопрос:

?ЗАПИСЬ\_В("гр:"),аллея(10,100,9);

2. вычитание(a,b,c)<-СЛОЖЕНИЕ(b,c,a);  
 квадрат(x,y,l)<-СЛОЖЕНИЕ(x,l,d),СЛОЖЕНИЕ(y,l,e),ЛИНИЯ(x,y,d,y,5),  
 ЛИНИЯ(d,y,d,e,5),ЛИНИЯ(d,e,x,e,5),ЛИНИЯ(x,e,x,y,5);  
 картинка(x,y,1)<-квадрат(x,y,10);  
 картинка(x,y,n)<-БОЛЬШЕ(n,1),ВЫЧИТАНИЕ(n,1,k),картинка(x,y,k),  
 УМНОЖЕНИЕ(10,k,0,z),вычитание(x,z,x1'),  
 вычитание(y,z,y1'),УМНОЖЕНИЕ(20,k,10,m),  
 квадрат(x1',y1',m);

Квадрат с номером n имеет следующие координаты левого верхнего угла и длину стороны:

$$\begin{cases} X_n = x - 10(n-1) \\ Y_n = y - 10(n-1) \\ L_n = 10 + 20(n-1) \end{cases}$$

Для того чтобы программа заработала, задайте вопрос:

?ЗАПИСЬ\_В("гр:"),картинка(100,100,3);

3. Сначала напишем правило для изображения одного дерева:

вычитание(a,b,c)<-СЛОЖЕНИЕ(b,c,a);  
 дерево(x,y,r)<-вычитание(y,r,z),ЛИНИЯ(x,y,x,z,6),вычитание(z,r,h),  
 ОКРУЖНОСТЬ(x,h,r);

Затем запишем базис (факт для аллеи из одного дерева) и рекурсивное правило для изображения  $n$  деревьев. Они аналогичны тем, что рассматривались в задаче изображения забора из  $n$  палочек.

аллея( $x, y, 1$ )<-дерево( $x, y$ );  
 аллея( $x, y, n$ )<-БОЛЬШЕ( $n, 1$ ), ВЫЧИТАНИЕ( $n, 1, p$ ), аллея( $x, y, p$ ),  
 УМНОЖЕНИЕ( $10, p, 0, k$ ), СЛОЖЕНИЕ( $x, k, b$ ), дерево( $b, y$ );  
 Если задать вопрос

?ЗАПИСЬ\_В("гр:"), аллея( $10, 100, 9$ );

на экране «вырастет» аллея.

4. Сначала рассчитаем длину стороны  $i$ -го квадрата  $L_i$ . Координаты его верхнего левого угла —  $X_i, Y_i$ . Через  $D_i$  обозначим сумму длин сторон всех предыдущих квадратов:

$$\begin{cases} L_i = 10 * i \\ X_i = x + D_i \\ Y_i = y + D_i \\ D_i = \frac{10 + 10i}{2} * i = (5 + 5i) i \end{cases}$$

В программе  $L_i$  и  $D_i$  обозначены  $l$  и  $d$ .

Сначала запишем правило для изображения квадрата ( $c$  — его цвет):

Базис будет таким:

квадрат( $x, y, l, c$ )<-СЛОЖЕНИЕ( $x, l, d$ ), СЛОЖЕНИЕ( $y, l, e$ ), ЛИНИЯ( $x, y, d, y, c$ ),  
 ЛИНИЯ( $d, y, d, e, c$ ), ЛИНИЯ( $d, e, x, e, c$ ), ЛИНИЯ( $x, e, x, y, c$ );  
 лестница( $x, y, 1$ )<-квадрат( $x, y, 10$ );

А рекурсивное правило выглядит так:

вычитание( $a, b, c$ )<-СЛОЖЕНИЕ( $b, c, a$ );  
 лестница( $x, y, n, c$ )<-БОЛЬШЕ( $n, 1$ ), вычитание( $n, 1, k$ ), лестница( $x, y, k, c$ ),  
 УМНОЖЕНИЕ( $10, l, 0, l$ ), УМНОЖЕНИЕ( $5, l, 5, q$ ), УМНОЖЕНИЕ( $q, i, 0, d$ ),  
 СЛОЖЕНИЕ( $x, d, x1'$ ), СЛОЖЕНИЕ( $y, d, y1'$ ), квадрат( $x1', y1', l, c$ );

Не забудьте установить графический режим.

Рассчитайте максимальное  $n$ , при котором картинка поместится на экране.

5.

%окружности-неваляшки;

неваляшка( $x, y, l, c$ )<-ОКРУЖНОСТЬ( $x, y, 10, c$ );

неваляшка( $x, y, n, c$ )<-БОЛЬШЕ( $n, 1$ ), СЛОЖЕНИЕ( $k, l, n$ ), неваляшка( $x, y, k, c$ ),  
 УМНОЖЕНИЕ( $n, n, 0, n2'$ ), УМНОЖЕНИЕ( $10, n, 0, p$ ), СЛОЖЕНИЕ( $y1', 10, y$ ),  
 УМНОЖЕНИЕ( $10, n2', y1', y2'$ ), ОКРУЖНОСТЬ( $x, y2', p, c$ );

?ЗАПИСЬ\_В("гр:"), неваляшка( $123, 12, 4, 15$ );

6. Расстояние между двумя соседними деревьями: по горизонтали равно  $Dx$ , а по вертикали —  $Dy$ . Всего должно быть  $n$  пар деревьев. Обозначим координаты корня «верхнего» дерева в паре  $X_{i1}$  и  $Y_{i1}$ , а нижнего —  $X_{i2}$  и  $Y_{i2}$ . Они вычисляются по следующим формулам:

$$\begin{cases} X_{i1} = X_{i2} = x + Dx * (i - 1) \\ Y_{i1} = y - Dy * (i - 1) \\ Y_{i2} = y + Dy * (i - 1) \end{cases}$$

вычитание( $a, b, c$ )<-СЛОЖЕНИЕ( $b, c, a$ );

дерево(x,y,r,l,c)<-вычитание(y,l,z),ЛИНИЯ(x,y,x,z,c),вычитание(z,r,h),  
 ОКРУЖНОСТЬ(x,h,c);  
 аллея(x,y,dx',dy',1)<-дерево(x,y,10,10,3);  
 аллея(x,y,dx',dy',n)<-БОЛЬШЕ(n,1),вычитание(n,1,k),аллея(x,y,dx',dy',k),  
 УМНОЖЕНИЕ(dx',k,x,x1'),УМНОЖЕНИЕ(dy',k,0,yp'),СЛОЖЕНИЕ(yp',y,y2'),  
 вычитание(y,yp',y1'),дерево(x1',y1',10,10,3),  
 дерево(x1',y2',10,10,3);

7. вычитание(a,b,c)<-СЛОЖЕНИЕ(c,b,a);  
 спираль(x,y,0,c)<-ТОЧКА(x,y,c);  
 спираль(x,y,n,c)<-БОЛЬШЕ(n,0),УМНОЖЕНИЕ(n,10,0,nf'),вычитание(x,nf',xn'),  
 вычитание(y,nf',yn'),УМНОЖЕНИЕ(2,n,0,n2'),УМНОЖЕНИЕ(n2',10,0,l),  
 СЛОЖЕНИЕ(yn',l,yn1'),СЛОЖЕНИЕ(xn',l,xn1'),вычитание(l,10,l1'),  
 СЛОЖЕНИЕ(yn',10,yn2'),СЛОЖЕНИЕ(xn',10,xn2'),  
 ЛИНИЯ(xn',yn',xn',yn1',c),ЛИНИЯ(xn',yn1',xn1',yn1',c),  
 ЛИНИЯ(xn1',yn1',xn1',yn2',c),ЛИНИЯ(xn1',yn2',xn2',yn2',c),  
 вычитание(n,1,k),спираль(x,y,k,c);

Для получения картинки задайте вопрос:

??ЗАПИСЬ\_В("gr:").спираль(100,100,4,15);

*(Продолжение следует)*

\* \* \*

### Читатель и редакция

Уважаемая редакция, к сожалению, в некоторых программных листингах, опубликованных в журнале «Информатика и образование», № 1 за 1994 г., оказались потеряны знаки «<» и «>». Исправленные тексты программ должны быть следующими.

1. Статья А. А. Самбиева, стр. 89, третий по счету листинг:

```
IF X<0 GOTO 100
IF X=0 GOTO 200
IF X>0 GOTO 300
```

2. Статья В. Рейникова, стр. 100, первый листинг:

```
пока I%>0 - 10 FOR I%=1 TO 0 STEP-1:INPUT I%:NEXT
пока I%<0 - 10 FOR I%=1 TO 0:INPUT I%:NEXT
пока I%=0 - 10 FOR I%=0 TO 1:INPUT I%:I%=ABS(I%):NEXT
пока I%<>0 - 10 FOR I%=1 TO 0 STEP 0:INPUT I%:NEXT
```

3. Там же, второй фрагмент:

```
пока I<5 : 10 FOR I%=6 TO 5 STEP-1:INPUT I%:NEXT
```

4. Примечание редактора, стр. 101:

```
IF (I+Step) <= Кон_I THEN повтор цикла
```

```
IF (I-Step) => Кон_I THEN повтор цикла
```

В. В. Михеев,  
г. Самара

От редакции. Редакция приносит извинения авторам А. А. Самбиеву и В. Рейникову за допущенные ошибки.

С. Р. Удалов,

г. Омск

## ОБУЧЕНИЕ КОНСТРУИРОВАНИЮ С ПРИМЕНЕНИЕМ КОМПЬЮТЕРА НА УРОКАХ ЧЕРЧЕНИЯ

Удалов Сергей Робертович — кандидат педагогических наук, ст. преподаватель кафедры новых информационных технологий в образовании Омского государственного педагогического университета, учитель черчения гимназии № 19.

Работает на художественно-графическом факультете педагогического университета. Специализация — информатика и новые информационные технологии в подготовке учителей черчения и изобразительного искусства.

Компьютер — устройство для обработки данных. Данные могут быть числовыми и текстовыми, звуковыми и графическими. И возможности компьютера оперировать данными нашли широкое применение в сфере, связанной с черчением. Инженеров-конструкторов компьютер заинтересовал прежде всего как устройство, позволяющее автоматизировать проектирование. Современная вычислительная техника значительно повышает эффективность работы конструкторов. По оценке специалистов, ход работы ускоряется в 5—6 раз. При этом достигается и другой важный эффект — возможность перебора и сравнения гораздо большего числа вариантов конструкций позволяет оптимизировать различные их характеристики и параметры. В этой области накоплен значительный опыт, есть очевидные положительные результаты, что открывает путь применению компьютера в учебной деятельности, связанной с черчением.

Для успешного применения компьютеров в черчении необходим комплекс программных и аппаратных средств. Часть таких средств можно подобрать среди существующих, другие нужно будет создавать.

Необходимы специальные программы, поддерживающие разнообразные типы учебных занятий по черчению. Большую перспективу имеют специальные программы типа AutoCAD или Компас. Однако использование их приведет к перестройке содержания школьного курса черчения. На уроках могут применяться и программные средства общего назначения — учебные базы данных, электронные таблицы, графические редакторы. Но для этого требуется разработка методики работы с ними. И конечно, особое внимание необходимо уделить специальным педагогическим программным средствам. Среди них особо выделим три.

*Информирующая система*, где ПЭВМ выступает только в роли источника информации и эффективно «загружает» память учащегося. Такая программа необязательно включает в себя все теоретические сведения, которые предусматривает школьная программа. Целесообразнее ввести в информирующую систе-

му только основные, базовые сведения, так как тонкости учебного материала учитель должен давать сам — машина не сможет разъяснить их. Излагать учебный материал при помощи ЭВМ в традиционном порядке, т. е. по схеме «учебник—ученик», не рационально. В этом случае информация идет в одном направлении — от источника к приемнику; при этом практически не контролируется степень обработки этой информации, необходимость дополнительных пояснений. Ведь одно из достоинств компьютера — возможность поддерживать обратную связь с учеником. Следовательно, компьютер таким образом регулирует процесс обучения, причем не периодически, а постоянно. Осуществляется это с помощью мини-теста после каждой порции информации. По результатам этого теста учащийся или переходит к следующему этапу, или остается на прежнем этапе для более детального изучения плохо усвоенного материала.

*Система-тренажер*, где ПЭВМ нацелена на формирование автоматических физических и умственных действий учащихся. Если говорить о содержании такой системы, то она может формировать навыки поиска информации, расчета соединений, создания графических изображений. Но кроме автоматизма ученику требуется и осмысление своих действий, поэтому основой этой системы служит не большое количество однотипных упражнений, а система разнообразных упражнений, требующих от ученика творческого подхода, инициативы, самостоятельности.

Если в материал программы входят достаточно сложные упражнения, то целесообразно на первоначальном этапе разбить их на более простые. Компоновать упражнения необходимо по принципу возрастающей сложности. Кроме того, неплохо предусмотреть при подборе дидактического наполнения конкретных упражнений различные уровни сложности. Возможен такой порядок расположения упражнений в программе, использующей конструктор из геометрических тел и тренирующей мыслительные навыки учеников: анализ геометрического состава изображений (перечисление геометрических тел, входящих в состав детали); замена одних геометрических тел в детали другими (переконструирование); создание детали с определенными параметрами из набора геометрических тел.

*Развивающая система*, предназначенная для совершенствования способностей обучающегося, таких, например, как способность к комбинированию образов, логическому мышлению. В таких системах учащемуся предлагается самостоятельно получить знания, решая ряд задач. Естественно, что самостоятель-

ность ученика относительна: его надо направлять, указывать на неверные выводы. Этот метод не нов, но в реальных условиях мало применим, потому что учитель не в состоянии проработать материал таким образом с каждым школьником, а работа со всем классом теряет смысл — поисками истины займутся единицы, остальные будут пассивно наблюдать за событиями. Компьютер здесь может в какой-то степени заменить учителя и поработать индивидуально с каждым учащимся.

Развивающие системы, на наш взгляд, наиболее подходят для решения школьниками задач с элементами конструирования. Решение таких задач не только формирует у учеников профессиональные навыки, но и развивает их мышление. Применение обучающих систем на базе персонального компьютера повышает эффективность этого процесса.

Поисковая форма освоения знаний активизирует интерес учащегося к самому процессу обучения. Подобные программы могут выглядеть так: учащемуся предлагается конкретный фактический материал, который следует проанализировать. На основе анализа решаются задачи и делаются определенные выводы. Именно на таких принципах построен пакет программ «Конструирование сборочных соединений», разработанный в Омском государственном педагогическом университете.

Этот пакет представляет собой развивающую систему, имеющую следующие характеристики:

- педагогическое назначение ее состоит в обучении конструированию при изучении темы «Сборочные чертежи» в VIII классе;
- содержанием данной развивающей системы являются автоматизированные конструкторы различных сборочных соединений (шпоночного, штифтового, болтового и т. д.);
- ввод данных учеником производится с клавиатуры, вывод результатов — на дисплей. Работа ученика с данной системой имитирует работу конструктора с системой автоматизированного проектирования. Ученику надо сконструировать изделие, соединяя различные детали. Но детали изначально не готовы к сборке. Их надо предварительно подготовить (проточить шпоночные пазы, нарезать резьбу и т. п.).

Учащийся должен выбрать параметры этих элементов, ввести их в компьютер, а машина сама дополнит чертежи деталей этими элементами. Когда все детали изделия доработаны таким образом, можно приступить к сборке. Для этого вводится название детали и ее изображение появляется на экране. Но для того чтобы сборка была выполнена правильно, необходимо соблюдать ее логику, т. е. соблюдать последовательность соединения деталей. Программы из этого пакета делятся на две группы.

Первая группа — программы на констру-

ирование простых изделий с одним видом соединений (например, только шпоночное). Успешно справившись с этой работой, учащиеся могут приступить к работе с программами второй группы.

Программы второй группы позволяют конструировать более сложные изделия с соединениями разных видов (например, штифтовое и шпоночное). Перенос знаний и навыков, полученных при работе с программами первой группы, на решение задач второй группы позволяет развивать логическое мышление школьников. Этот пакет программ можно успешно применять при изучении соединений деталей.

Внедрение его в учебный процесс позволяет ученикам в дальнейшем более успешно решать задачи на детализирование сборочных чертежей, так как процесс составления сборочных чертежей реализуется с помощью программ пакета «Конструирование сборочных соединений».

В пакет входят шесть программ, предназначенных для обучения конструированию с применением шпоночного соединения, с применением штифтового соединения, с применением шпоночного и штифтового соединений, с применением резьбового соединения, с применением болтовых соединений, с применением шпилечного и болтового соединений.

Приведем пример работы программы на конструирование шпоночного соединения.

Задание для учащихся: сконструировать соединение, позволяющее передавать вращательный момент от вала. Выполните чертежи деталей, входящих в сборку, а затем сборочный чертеж, учитывая логику сборки.

После запуска программы на экране появляется чертеж вала и меню (рис. 1), в котором можно выбрать чертеж рычага (рис. 2), справочник по шпонкам или перейти к выполнению сборочного чертежа. Для разработки чертежей деталей на них надо выполнить изготовления пазов. Поэтому сначала, исходя из диаметра вала и ширины ступицы рычага, необходимо подобрать размеры шпонки и узнать параметры пазов. Это делается при выборе режима «Шпонка» (рис. 3). Зная нужные параметры, можно выполнить чертежи деталей, вводя значения размеров на запросы компьютера (рис. 4, 5). Когда эта часть работы закончена, можно переходить к режиму сборки. Там, называя по порядку вхождения в сборку наименования деталей, можно создать сборочный чертеж шпоночного соединения (рис. 6).

Таким образом, используя этот пакет программ, другие программы по черчению и специальные профессиональные программные системы, можно перестроить процесс обучения черчению, ликвидировав существующее в настоящее время несоответствие между потребностью сферы производства и конструкторской деятельности в применении компьютеров и новых информационных технологий и отсутствием подготовки учащихся в школе к применению таких средств и технологий.

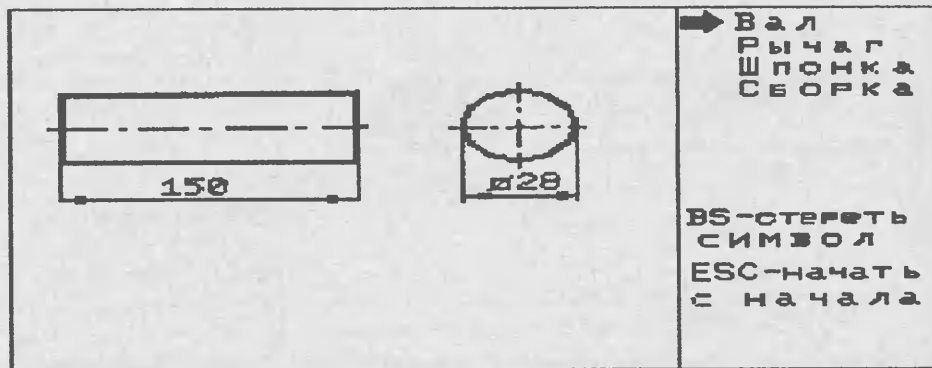


Рис. 1

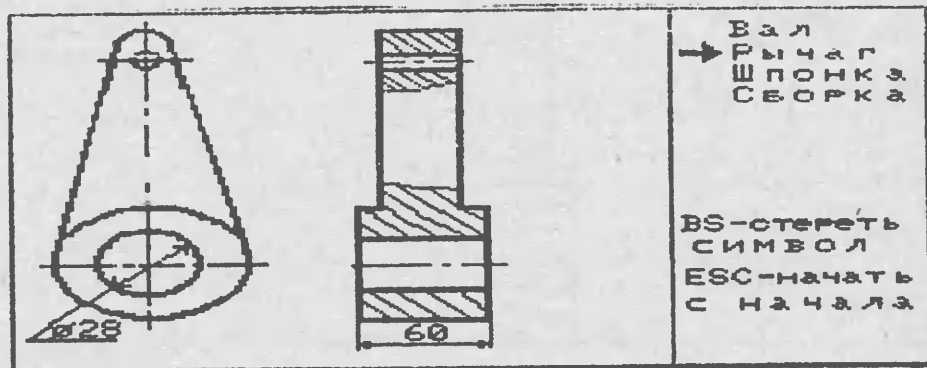


Рис. 2

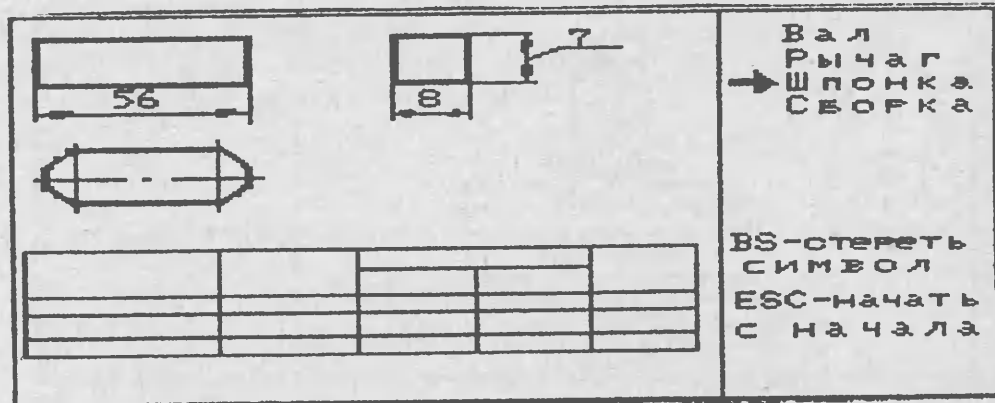


Рис. 3

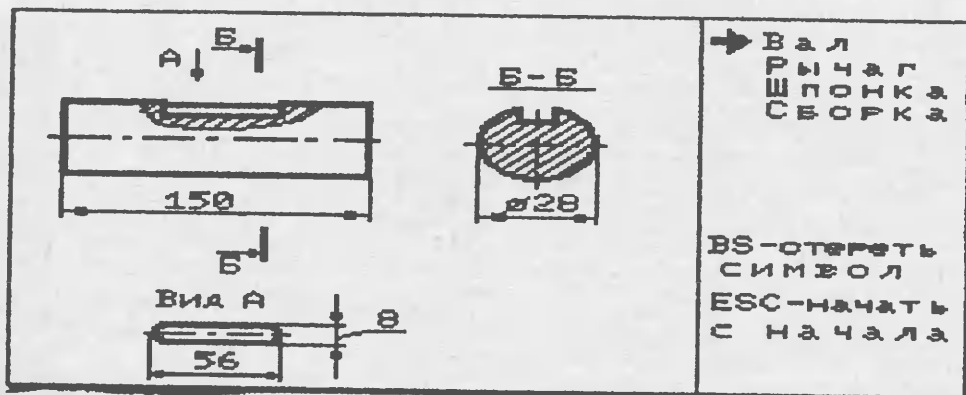


Рис. 4

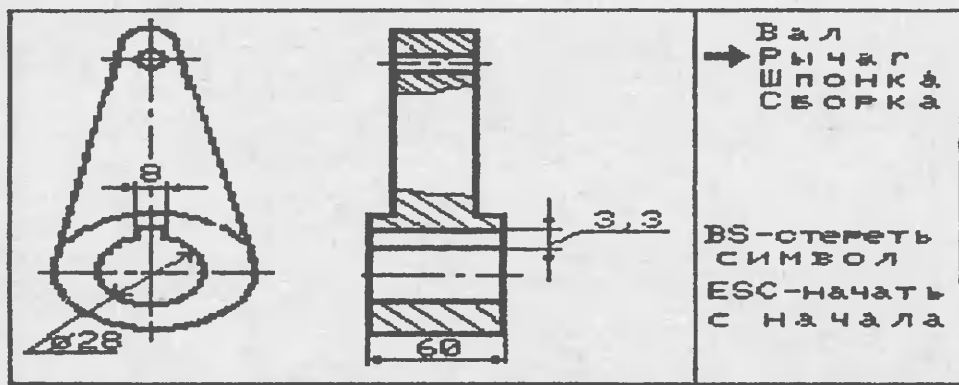


Рис. 5

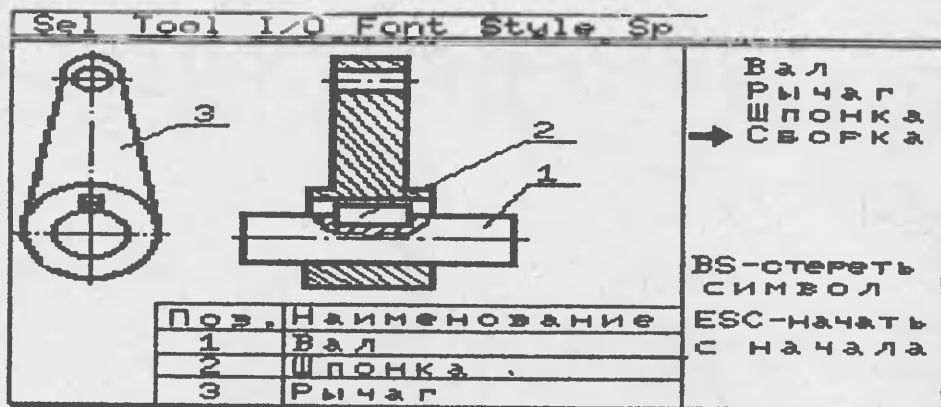


Рис. 6



Современные учебники не предоставляют школьникам теории массивов, хотя роль их велика: они используются для упрощения процесса составления программ, создания более лаконичных листингов. В школах ЭВМ часто применяют в качестве амперметра, вольтметра, весов (на лабораторных работах). При этом в основе программ лежит использование массивов (DIM). Затем данные массивов применяются для графического представления динамики физических явлений.

Изучение темы «Массивы» в календарном плане, составленном автором, занимает целое полугодие XI класса. Все методические разработки теории по этой теме автор составляет из следующих пунктов:

1. Табличный способ организации данных.
2. Организация данных на языке БЕЙСИК.
3. Данные в массиве.
4. Способы заполнения массивов.
5. Задачи на обработку числового массива.
6. Задачи на обработку символьного массива.
7. Задачи на нахождение элементов, отвечающих заданному условию.
8. Задачи на сортировку массива.

О. Н. Маркова,

Удмуртия

## ОБРАБОТКА МАССИВОВ НА ЯЗЫКЕ БЕЙСИК

В программах на БЕЙСИКе нередко используются массивы данных — одномерные и двумерные (в обычном понимании это соответствует списку и прямоугольной таблице), реже трех- и более «мерные». Прежде чем использовать их в программе, надо дать машине указание, чтобы она заранее отвела в памяти место. Для этого служит команда DIM (по-английски «dimension» — «размер»). В этой команде указывается имя массива и количество элементов в нем (для двумерного — число строк и столбцов):

```
DIM A(10)
DIM X(3,4)
```

Определяются одномерный массив А и двумерный X. (В версии БЕЙСИКа, реализованной на БК, а также на многих других компьютерах, отсчет номеров, или индексов, элементов ведется от нуля. Поэтому, вообще говоря, DIM B(3) отводит место для четырех элементов, а не трех: B(0), B(1), B(2) и B(3). Однако нулевой элемент часто не используется. — Прим. ред.)

Следует помнить, что при резервировании памяти для массива оператор DIM должен выполняться до первого обращения к элементам массива. Лучше всего записывать этот оператор в первой строке программы — это улучшит ее читаемость.

Действие «ввести массив» программируется следующим образом:

```
10 DIM A(6)
20 FOR I=1 TO 6
30 INPUT A(I)
40 NEXT I
```

Выполняя этот фрагмент, БК-0010.01 последовательно запрашивает числа и помещает их в соответствующие ячейки массива.

Для двумерного массива используется более сложная конструкция:

```
10 DIM A(3,4)
20 FOR I=1 TO 3
30 FOR J=1 TO 4
40 INPUT A(I,J)
50 NEXT J
60 NEXT I
```

В цикле по I (строки 20 и 60) производится последовательный перебор номеров строк двумерного массива. Внутренний цикл и оператор INPUT (30—50) обеспечивают ввод элементов очередной строки.

### Данные в массиве

Данные в БЕЙСИКе могут принадлежать к одному из типов: вещественные, целочисленные, символьные. Если в программе тип не указан явно, то данные считаются вещественными.

Чтобы указать, что переменные и массивы содержат целочисленные значения, нужно приписать к их именам знак «%». Например: A%, X% — имена целочисленных переменных.

Пример программы:

```
10 DIM A%(3,4)
20 FOR I=1 TO 3
30 FOR J=1 TO 4
40 INPUT A%(I,J)
50 NEXT J
60 NEXT I
```

Второй способ получения целочисленных значений — с помощью функции INT, которая округляет вещественное число до целого, отбрасывая дробную часть:

```
10 DIM A(4)
20 FOR I=1 TO 4
30 A(I)=INT(RND(1)*10)
40 NEXT I
```

Элементы массива A получают случайные значения от 0 до 9.

Символьный тип переменных и массивов определяется знаком\$. Примеры: V\$, Y\$ — символьные переменные.

```
10 DIM A $(4)
20 FOR I=1 TO 4
30 INPUT A $(I)
40 NEXT I
50 FOR I=1 TO 4
60 PRINT A $(I);
70 NEXT I
```

Строки 20—40 запрашивают, а 50—70 выводят на экран четыре текстовые строки. Другой способ получения символьных значений — оператор STR, который преобразует числовые данные в символьные:

```
10 A=123
20 X=STR $(A)
30 PRINT MID $(X,2,LEN(X)-2)
```

Функция MID\$ в строке 30 отсекает лишние пробелы, которые БК-0010.01 добавляет к любому числу слева и справа.

### Способы заполнения массивов

1. Ввод с клавиатуры:

```
10 REM ЗАПОЛНЕНИЕ МАССИВА С КЛАВИАТУРЫ
20 DIM A(3,4)
30 FOR I=1 TO 3
40 FOR J=1 TO 4
50 INPUT A(I,J)
60 NEXT J,I
70 ? "-----"
80 FOR I=1 TO 3
90 FOR J=1 TO 4
100 PRINT A(I,J);
```

```
110 NEXT J
```

```
120 ?
```

```
130 NEXT I
```

2. Ввод с использованием датчика случайных чисел:

```
10 REM ЗАПОЛНЕНИЕ МАССИВА ВВОДОМ СЛУЧАЙНЫХ ЧИСЕЛ
```

```
20 DIM A(2,3)
```

```
30 FOR I=1 TO 2
```

```
40 FOR J=1 TO 3
```

```
50 A(I,J)=INT(RND(1)*10-5)
```

```
60 ? A(I,J);
```

```
70 NEXT J
```

```
80 ?
```

```
90 NEXT I
```

3. Ввод значений, записанных в тексте программы:

```
10 REM ВВОД МАССИВА ЧЕРЕЗ ЗАДАННЫЕ В ПРОГРАММЕ ДАННЫЕ/READ,DATA/
```

```
20 DIM A (4)
```

```
30 FOR I=1 TO 4
```

```
40 READ A (I)
```

```
60 NEXT I
```

```
70 DATA ЛЕНА,ОЛЯ,КАТЯ,КОЛЯ
```

```
80 FOR I=1 TO 4
```

```
90 ? A (I)
```

```
100 NEXT I
```

4. Расчет их по формулам:

```
10 REM ВВОД МАССИВА С ИСПОЛЬЗОВАНИЕМ ФОРМУЛ
```

```
20 DIM T%(9,9)
```

```
30 FOR I=1 TO 9
```

```
40 FOR J=1 TO 9
```

```
50 T%(I,J)=I*J
```

```
60 PRINT T%(I,J);
```

```
70 NEXT J
```

```
80 PRINT
```

```
90 NEXT I
```

### Задачи на обработку числового массива

1. Программа вычисления суммы элементов массива.

а) одномерного:

```
10 REM СУММИРОВАНИЕ ЭЛЕМЕНТОВ ОДНОМЕРНОГО МАССИВА
```

```
20 DIM A(N)
```

```
30 REM ВВОДИМ ЭЛЕМЕНТЫ МАССИВА
```

```
40 FOR I=1 TO N
```

```
50 INPUT A(I)
```

```
60 NEXT I
```

```
70 REN S-СУММА
```

```
80 S=0
```

```
90 REM К СУММЕ ДОБАВЛЯЕМ ПО ОЧЕРЕДИ ЭЛЕМЕНТЫ МАССИВА
```

```
100 FOR I=1 TO N
```

```
110 S=S+A(I)
```

```
120 NEXT I
```

```
130 REM ПЕЧАТАЕМ РЕЗУЛЬТАТ
```

```
140 PRINT "S=";S
```

б) двумерного:

```
10 REM СУММИРОВАНИЕ ЭЛЕМЕНТОВ ДВУМЕРНОГО МАССИВА
```

```
20 DIM A(2,3)
```

```
30 FOR I=1 TO 2
```

```
40 FOR J=1 TO 3
```

```

50 A(I,J)=INT(RND(1)*10)
60 PRINT A(I,J);
70 S=S+A(I,J)
80 NEXT J
90 PRINT
100 NEXT I
110 ? "-----"
120 PRINT "СУММА S=" S

```

Следует заметить, что при первом обращении к переменной БК-0010.01 автоматически присваивает ей значение 0.

2. Вычисление произведения элементов массива.

а) одномерного:

```

10 DIM X(3)
20 P=1
30 FOR I=1 TO 3
40 X(I)=INT(RND(1)*10)
50 P=P*X(I)
60 NEXT I
70 ? P

```

б) двумерного:

```

10 REM ПРОИЗВЕДЕНИЕ ЭЛЕМЕНТОВ ДВУМЕРНОГО МАССИВА
20 P=1
30 DIM X(2,3)
40 FOR I=1 TO 2
50 FOR J=1 TO 3
60 X(I,J)=INT(RND(1)*10)
70 PRINT X(I,J);
80 P=P*X(I,J)
90 NEXT J
100 PRINT
110 NEXT I
115 ? "-----"
120 PRINT "ПРОИЗВЕДЕНИЕ P="P

```

3. Поиск суммы элементов в каждой строке двумерного массива:

```

10 DIM A(20,20) ' ПРЕДПОЛАГАЕМ, ЧТО ЭЛЕМЕНТЫ ВВЕДены
20 FOR I=1 TO 20
30 ? "СУММА ЭЛЕМЕНТОВ I СТРОКИ"
40 S=0
50 FOR J=1 TO 20
60 S=S+A(I,J)
70 NEXT J
80 ? "СУММА СТРОКИ";S
90 NEXT I

```

4. «Вычислительный эксперимент». Подбросьте 20 раз монету и при выпадении «решки» вводите 1, а в случае «орла» — 0. Каким будет вычисленное среднее значение этой последовательности? Получились ли новые средние значения близкими к предыдущим?

```

10 DIM A(20)
20 FOR I=1 TO 20
30 INPUT A(I)
40 S=S+A(I)
50 NEXT I
60 P=S/20
70 ?"P="P

```

## Задачи на обработку символьного массива

1. Подсчитайте, сколько слов начинается с буквы «К».

а) среди слов, записанных в операторе DATA:

10 REM СКОЛЬКО СЛОВ НАЧИНАЕТСЯ С БУКВЫ 'К'

20 DIM A (4)

30 FOR I=1 TO 4

40 READ A (I)

50 IF MID (A (I),1,1)="K" TH K=K+1

60 NEXT I

70 PRINT "КОЛИЧЕСТВО СЛОВ С БУКВЫ'К'."K

80 DATA КОТ,КОЗА,ЛИСА,КОРОВА

б) в тексте, введенном с клавиатуры:

10 REM СКОЛЬКО СЛОВ В ТЕКСТЕ НАЧИНАЕТСЯ С БУКВЫ 'К'

20 INPUT "ВВЕДИТЕ ТЕКСТ ";A

30 FOR I=1 TO LEN(A )

40 IF MID (A ,I,1)=" " TH 50 EL 60

50 IF MID (A ,I+1,1)="K" TH L=L+1

60 NEXT I

70 ? "-----"

80 IF MID (A ,I,1)="K" TH PRINT "СЛОВ НАЧИНАЮЩИХ С 'К'." L+1 ELSE  
PRINT"СЛОВ НАЧИНАЮЩИХСЯ С 'К'."L

(При вводе текста следует использовать только заглавные русские буквы. — Прим. ред.)

2. Фамилии участников олимпиады хранятся в одном массиве, а количество баллов — в другом. Составьте программу, которая будет печатать фамилию победителя, если он набрал 100 баллов.

10 REM ПЕЧАТАЕМ ФАМИЛИЮ ПОБЕДИТЕЛЯ, НАБРАВШЕГО 100 БАЛЛОВ

20 DIM A (4),B(4)

30 FOR I=1 TO 4

40 READ A (I)

50 NEXT I

60 FOR I=1 TO 4

70 READ B(I)

80 NEXT I

90 DATA ИВАНОВ,ПЕТРОВ,БОГДАНОВ,СЕРОВ

100 DATA 23,95,100,90

110 FOR I=1 TO 4

120 IF B(I)=100 TH PRINT "ПОБЕДИТЕЛЬ ОЛИМПИАДЫ -"A (I)

130 NEXT I

## Задачи на нахождение элементов, отвечающих заданному условию

1. Подсчет количества элементов в заданном массиве, превосходящих число P.

10 REM КОЛИЧЕСТВО ЭЛЕМЕНТОВ В ЗАДАННОМ МАССИВЕ, ПРЕВОСХОДЯЩИХ  
ЧИСЛО P

20 DIM A(2,3)

30 INPUT "ВВЕДИТЕ ЧИСЛО";P

40 FOR I=1 TO 2

50 FOR J=1 TO 3

60 A(I,J)=INT(RND(1)\*15-8)

70 ? A(I,J);

80 IF A(I,J)>P TH K=K+1

90 NEXT J

100 ?

110 NEXT I

120 ? "-----"

130 ? "ЧИСЕЛ БОЛЬШИХ "P "=""K

2. Подсчет количества положительных элементов:

```
10 REM КОЛИЧЕСТВО ПОЛОЖИТЕЛЬНЫХ ЭЛЕМЕНТОВ МАССИВА
20 DIM A(2,3)
40 FOR I=1 TO 2
50 FOR J=1 TO 3
60 A(I,J)=INT(RND(1)*10-5)
70 ? A(I,J);
80 IF A(I,J)>0 THEN K=K+1
90 NEXT J
100 ?
110 NEXT I
120 ? "_____”
130 ? "ПОЛОЖИТЕЛЬНЫХ ЧИСЕЛ "K
```

3. Подсчет количества четных элементов:

```
10 REM КОЛИЧЕСТВО ЧЕТНЫХ ЭЛЕМЕНТОВ МАССИВА
20 DIM A(2,3)
40 FOR I=1 TO 2
50 FOR J=1 TO 3
60 A(I,J)=INT(RND(1)*10-5)
70 ? A(I,J);
80 IF (A(I,J)/2=INT(A(I,J)/2) AND A(I,J)<>0) THEN K=K+1
90 NEXT J
100 ?
110 NEXT I
120 ? "_____”
130 ? "ЧЕТНЫХ ЧИСЕЛ "K
```

4. Поиск максимального элемента в двумерном массиве:

```
10 REM МАКСИМАЛЬНЫЙ ЭЛЕМЕНТ В МАССИВЕ
20 DIM A(2,4)
30 FOR I=1 TO 2
40 FOR J=1 TO 4
50 A(I,J)=INT(RND(1)*10)
60 ? A(I,J);
70 IF A(I,J)>M THEN M=A(I,J)
80 M=A(I,J)
90 NEXT J
100 ?
110 NEXT I
120 ? "_____”
130 ? "МАКСИМАЛЬНЫЙ ЭЛЕМЕНТ МАССИВА-"M
```

5. «Суперзамок». Секретный замок для сейфа состоит из 10 расположенных в ряд ячеек, в которые надо вставить игральные кубики. Но дверь открывается только в том случае, когда в любых трех соседних ячейках сумма точек на передних гранях кубиков равна 10. (Игральный кубик имеет на каждой грани от 1 до 6 точек.) Напишите программу, которая разгадывает код замка при условии, что два кубика уже вставлены в ячейки.

```
10 REM СЕКРЕТНЫЙ ЗАМОК ДЛЯ СЕЙФА
20 REM ИГРАЛЬНЫЙ КУБИК ИМЕЕТ НА КАЖДОЙ ГРАНИ ОТ 1 ДО 6 ТОЧЕК
30 DIM A(10)
40 INPUT X ' ПЕРВЫЙ КУБИК
50 INPUT U ' ВТОРОЙ КУБИК
55 IF X+U<4 THEN ? "ОШИБКА! ПРИ ТАКИХ НАЧАЛЬНЫХ ДАННЫХ РЕШЕНИЯ
    НЕТ." ELSE 60
56 GOTO 40
60 ?
70 A(1)=X
80 A(2)=U
90 ? A(1)
```

```

100 ? A(2)
110 FOR I=3 TO 10
120 A(I)=10-A(I-1)-A(I-2)
130 ? A(I)
140 NEXT I

```

6. Составьте программу, подсчитывающую количество всех возможных трехзначных чисел, записанных цифрами 1, 2, 3, 4 и 5. (В числе не должно быть одинаковых цифр.)

```

10 FOR I=1 TO 5
20 FOR J=1 TO 5
30 FOR K=1 TO 5
40 IF(I=J AND I=K) OR (I=J AND K<>I) OR (J=K AND I<>J) OR (I=K AND J<>I) TH 60
EL T=T+1
50 NEXT K,J,I
60 PRINT T
70 END

```

7. При поступлении в институт абитуриенты, получившие двойку на первом экзамене, ко второму не допускаются. В массиве А записаны оценки экзаменующихся. Подсчитать, сколько человек не допущено ко второму экзамену.

```

10 DIM A(10)
20 K=0
30 FOR I=1 TO 10
40 INPUT A(I)
50 IF A(I)=2 TH K=K+1
60 NEXT I
70 PRINT "НЕ ДОПУЩЕНО АБИТУРИЕНТОВ:"K

```

### Задачи на сортировку массива

1. Задан одномерный массив из 12 элементов. Сформировать из него двумерный размером 3x4.

```

10 REM ДВУМЕРНЫЙ МАССИВ ИЗ ОДНОМЕРНОГО
20 DIM A(9),B(3,3)
30 FOR I=1 TO 9
40 READ A(I)
45 ? A(I);
50 NEXT I
55 ?
56 ? "-----"
60 FOR I=1 TO 3
70 FOR J=1 TO 3
80 K=K+1
90 B(I,J)=A(K)
100 NEXT J
110 NEXT I
120 DATA 1,2,3,6,5,4,7,8,9
130 FOR I=1 TO 3
140 FOR J=1 TO 3
150 PRINT B(I,J);
160 NEXT J
165 ?
170 NEXT I

```

Возможные варианты подобных задач: сформировать из одномерного массива двумерный из элементов, стоящих на нечетных местах; сформировать из двумерного массива одномерный; обнулить элементы массива, стоящие на четных местах.

2. Упорядочивание одномерного массива, заданного датчиком случайных чисел, по возрастанию:

```

10 REM УПОРЯДОЧИТЬ ОДНОМЕРНЫЙ МАССИВ ПО ВОЗРАСТАНИЮ
20 DIM A(4)
30 FOR I=1 TO 4
40 A(I)=INT(RND(1)*10)
50 ? A(I);
60 NEXT I
70 ?
80 FOR I=1 TO 4
90 M=I
100 FOR J=I TO 4
110 IF A(J)<A(M) THEN 120 ELSE 130
120 M=J
130 NEXT J
140 T=A(I)
150 A(I)=A(M)
160 A(M)=T
180 ? A(I);
190 NEXT I

```

3. Упорядочивание двумерного массива по столбцам и строкам (предполагается, что значения элементов вводятся в строках 10—70):

```

80 FOR I=1 TO 4
90 FOR T=1 TO 4
100 K=T
120 M=I
130 FOR Y=K TO 4
140 FOR J=I TO 4
150 IF A(J,Y)>A(M,K) TH 180 EL 170
160 K=Y
170 M=J
180 NEXT J,Y
190 P=A(I,J)
200 A(I,J)=A(M,K)
210 A(M,K)=P
220 ?A(I,T);
230 NEXT T
240 ?
250 NEXT I

```

4. В заданном одномерном массиве поменяйте местами элементы, стоящие на четных местах, с нечетными.

```

10 REM ПЕРЕСТАВИТЬ ЧЕТНЫЕ И НЕЧЕТНЫЕ ЭЛЕМЕНТЫ
20 DIM A(6)
30 FOR I=1 TO 6
40 A(I)=INT(RND(1)*10)
50 ? A(I);
60 NEXT I
70 ?
75 ? "-----"
80 FOR I=1 TO 5 STEP 2
90 J=I+1
100 T=A(I)
110 A(I)=A(J)
120 A(J)=T
130 ? A(I) A(J);
140 NEXT I

```



# ОЛИМПИАДА ПО ИНФОРМАТИКЕ

---

**В. М. Кирюхин,**  
канд. техн. наук

## VI ВСЕРОССИЙСКАЯ ОЛИМПИАДА ШКОЛЬНИКОВ ПО ИНФОРМАТИКЕ

Нынешняя олимпиада школьников по информатике проходила в период с 23 по 30 марта 1994 г. в подмосковном городе Троицке. Третий год подряд это место становится центром притяжения лучших школьников в области информатики, большинство из которых — победители республиканских, областных и региональных олимпиад. И это не случайно. Интерес к такого рода соревнованиям не ослабевает, а, наоборот, постоянно возрастает, несмотря на то что компьютерная техника и информационная технология стали для всех привычным делом. Стремительное развитие этого направления науки и техники требует прилива свежих сил, и в этой связи важно не только освоение уже имеющихся достижений в области новых информационных технологий, но и создание собственного научного потенциала, способного конкурировать с ведущими зарубежными фирмами.

Несмотря на то что в проведении олимпиады были задействованы все компьютерные резервы г. Троицка, принять всех желающих оказалось невозможным. Тем не менее практически все регионы России прислали своих представителей. 113 школьников из 73 территорий страны на сей раз участвовали в соревновании, и организаторы олимпиады — Министерство образования РФ, департамент по образованию администрации Московской области и фонд новых технологий в образовании «Байтик» — сделали все возможное, чтобы этот смотр молодых дарований Рос-

сии в области информатики прошел на высоком уровне.

Традиционно олимпиада проходила в два тура. На каждом туре для решения представленных участникам задач было отведено 4,5 часа. В распоряжение каждого участника был выделен персональный компьютер, и допускалось использование одной из систем программирования: Turbo Pascal v. 6.0, Microsoft C v. 5, Borland C++ v. 2.0, GWBasic, Turbo Basic, Quick Basic. В отведенное время требовалось formalizовать представленную задачу, разработать по возможности лучший алгоритм ее решения, а также написать и отладить программу на одном из допустимых языков программирования. Результатом решения каждой задачи являлась работоспособная программа в исполняемом виде.

Отбором и подготовкой задач для олимпиады занималось авторитетное жюри, в состав которого входили известные ученые и специалисты школьной информатики. Среди них канд. техн. наук С. М. Окулов (г. Киров), канд. физ.-мат. наук В. В. Прохоров (г. Екатеринбург), канд. физ.-мат. наук Е. В. Андреева (г. Троицк), канд. техн. наук С. Г. Волченков (г. Ярославль), канд. техн. наук С. К. Коваленко (г. Москва), А. А. Суханов (г. Санкт-Петербург), А. В. Алексеев (г. Красноярск), И. А. Серегин (г. Нальчик), А. П. Марченко, А. П. Овсянников, В. П. Попов (г. Троицк). Возглавлял работу жюри автор этих строк.

При выборе олимпиадных задач жюри исходило из следующих соображений. Во-

первых, поскольку олимпиада по информатике, а не по программированию, то задачи должны представлять школьникам возможность проявить свои способности на всех этапах решения задач с использованием компьютера, начиная с формализации и кончая тестированием и отладкой программы. При этом учитывалась важность всех этапов: умение анализировать условие задачи с целью выбора или разработки алгоритма, в максимальной степени отвечающего поставленной задаче и существенно облегчающего дальнейшие этапы ее решения; знание самих алгоритмов решения различных классов задач, их преимуществ и недостатков, особенностей их использования в различных ситуациях, сложностей их реализации; практические навыки работы с соответствующей системой программирования, особенно при отладке программ; умение выбрать правильные тесты, позволяющие при отладке обнаружить имеющиеся ошибки в алгоритме или программе или подтвердить правильность решения.

Во-вторых, задачи должны определять практическую направленность информатики и привлечь внимание участников к совершенно неожиданным проявлениям информатики в реальной жизни. Более того, задачи должны стимулировать дальнейшее изучение информатики и показывать все ее многообразие и сложность по сравнению с тем, что преподается в рамках школьной информатики.

В-третьих, задачи должны были быть разумной сложности. С одной стороны, при их решении не должны были использоваться специальные знания, а с другой стороны, они должны были предоставлять возможность всем школьникам продемонстрировать свои знания и умения в полной степени.

Тщательный анализ имевшихся в распоряжении жюри условий задач позволил выбрать задачи, которые описаны ниже. При этом на первом туре участникам олимпиады было предложено две задачи, а на втором — одна.

Следует отметить разносторонний характер выбранных задач. Так, задача первого тура «Больница» прежде всего основана на проведении анализа возможных

решений при различных соотношениях используемых параметров. Такой класс задач очень важен при изучении и практическом использовании информатики, так как учит сначала тщательно подумать над задачей, прежде чем нажимать клавиши клавиатуры компьютера.

Во второй задаче первого тура — «Паркет» — за кажущейся простотой скрывается ряд интересных и не таких уж простых подходов к ее решению. Во-первых, в ней не требовалось представить на проверку работоспособную программу. Необходимо было заполнить только соответствующую таблицу. Но более или менее серьезный анализ показывает, что достичь поставленной цели без работоспособной программы нельзя. Во-вторых, если полностью решать задачу, то без знаний и умений работать с целочисленной арифметикой этого сделать невозможно. Однако умение соизмерять точностные характеристики компьютера с требуемыми показателями позволяет получить неполное решение задачи, а небольшую часть таблицы можно заполнить, вообще не используя компьютер.

Очень интересной оказалась задача второго тура «Буквоед». Жюри отлично понимало сложность этой задачи самой по себе, тем более что она относится к задачам искусственного интеллекта. Однако «олимпиадность» этой задачи и заключается в умении упростить ее до того уровня, который под силу одолеть конкретному участнику. Именно с таких позиций оценивалось решение этой задачи: кто сумел придумать более сложный алгоритм, распознающий большее число особенностей написания заданных букв, тот и получал лучшую оценку.

Результаты решения задач на этой олимпиаде приятно удивили и членов жюри, и тех, кто проверял представленные участниками решения. Особенно важно было, что наряду со школьниками выпускных классов отлично выступали учащиеся VII-X классов, хотя задачи были одинаковые для всех участников. Особо хотелось бы отметить выступление ученика VII класса средней школы № 64 г. Санкт-Петербурга Дурова Николая, который занял общее пятое место в категории невыпускных

классов и значительно опередил многих одиннадцатиклассников.

Проверка решений участников олимпиады осуществлялась по тестам. По результатам тестирования каждая задача первого тура оценивалась из 50 баллов. Задача второго тура оценивалась из 100 баллов. Таким образом, максимальное количество баллов, которое мог набрать каждый участник по результатам олимпиады, составляло 200 баллов.

По итогам олимпиады абсолютным чемпионом стал Антон Лапунов, ученик XI класса физико-математического лицея г. Кирова. Второе место занял Виктор Баргачев, ученик X класса лицея г. Санкт-Петербурга, на третьем месте — Олег Голубицкий, ученик XI класса СУНЦ МГУ. В общей сложности победителям олимпиады было вручено 12 дипломов первой степени, 13 дипломов второй степени, 11 дипломов третьей степени и 26 поощрительных дипломов. Список победителей VI Всероссийской олимпиады по информатике по двум возрастным категориям (выпускные классы и VII-X классы) приведен ниже.

Торжественно прошло закрытие олимпиады. Помимо почетных дипломов всем победителям были вручены памятные подарки, любезно предоставленные спонсорами олимпиады: ведущей российской фирмой в области сетевых технологий «ЭЛКО

Технологии», известной зарубежной фирмой Microsoft, а также фирмами «ХОПЕР Инвест», АО «Тристик», ассоциацией «ИНТО». Практически все участники олимпиады получили в подарок от редакций компьютерных журналов «Компьютер Пресс» и «PC Magazine», также спонсоров олимпиады последние номера журналов, а часть победителей получила годовые подписки на эти журналы. Специальные призы получили также Олег Голубицкий (г. Москва) — за лучшее решение задачи «Больница» (34 балла), Виктор Баргачев (г. Санкт-Петербург) — за лучшее решение задач «Паркет» (50 баллов) и «Буквоед» (97 баллов), а также Николай Дуров (г. Санкт-Петербург) — как самый молодой участник, добившийся наилучших результатов (101 балл).

По итогам олимпиады и по результатам предварительных сборов сформирована команда России для подготовки к международной олимпиаде по информатике, которая состоялась в этом году в Швеции в период с 2 по 10 июля 1994 г. Отрядным является тот факт, что трое кандидатов — Антон Лапунов, Виктор Баргачев, Роман Елизаров — уже участвовали в международной олимпиаде, которая проходила в прошлом году в Аргентине. Более того, Виктор Баргачев и Роман Елизаров получили там серебряные медали, а Антон Лапунов — бронзовую.

### Победители VI Всероссийской олимпиады школьников по информатике\*

Среди учащихся XI-х классов дипломы первой степени получили:

<b>Лапунов Антон</b>	— учащийся физ.-мат. лицея г. Кирова;
<b>Голубицкий Олег</b>	— учащийся СУНЦ МГУ;
<b>Филиппов Василий</b>	— учащийся ср. школы № 45 г. Санкт-Петербурга;
<b>Елизаров Роман</b>	— учащийся физ.-мат. школы № 45 г. Санкт-Петербурга
<b>Тарасов Алексей</b>	— учащийся СУНЦ МГУ;
<b>Вирченко Антон</b>	— учащийся ср. школы № 43 г. Москвы
<b>Харькин Виталий</b>	— учащийся ком. школы г. Уфы, республика Башкортостан;
<b>Ушаков Владимир</b>	— учащийся ср. школы № 82 г. Московской обл.;

\* От редакции. Данные о победителях напечатаны в соответствии с информацией, предоставленной автором.

дипломы второй степени получили:

**Стратонников Алексей**

— учащийся ср. школы № Ленинградской обл.;

**Савенко Дмитрий**

— учащийся ср. школы № 4 г. Красноярска;

**Аникин Александр**

— учащийся физ.-мат. школы № 30 г. Санкт-Петербурга;

**Новиков Алексей**

— учащийся техн. лицея г. Астрахани;

**Лебедев Игорь**

— учащийся лицея Алтайского края;

**Бозиев Артур**

— учащийся лицея № 2 г. Нальчика;

**Голынский Александр**

— учащийся СУНЦ МГУ;

**Осиновский Дмитрий**

— учащийся ср. школы № 25 г. Курска;

дипломы третьей степени получили:

**Битюков Сергей**

— учащийся СУНЦ МГУ;

**Кленин Александр**

— учащийся ср. школы № 23 г. Владивостока;

**Исаев Андрей**

— учащийся ср. школы № 51 г. Самары;

**Белоусов Олег**

— учащийся ср. школы № 8 г. Вологды;

**Мазуркин Николай**

— учащийся ср. школы № 11 г. Йошкар-Олы, республика Марий Эл;

**Махмудов Руслан**

— учащийся ср. школы № 41 г. Ижевска, республика Удмуртия;

**Дрожжин Роман**

— учащийся ср. школы № 41 г. Ижевска, республика Удмуртия.

Среди учащихся VII—X-х классов

дипломы первой степени получили:

**Баргачев Виктор**

— учащийся X класса физ.-мат. лицея г. Санкт-Петербурга;

**Дужин Федор**

— учащийся X класса ср. школы № 7 г. Ярославля;

**Беров Виталий**

— учащийся X класса физ.-мат. лицея г. Кирова;

**Семенов Олег**

— учащийся X класса физ.-мат. лицея г. Санкт-Петербурга;

дипломы второй степени получили:

**Дуров Николай**

— учащийся 7 класса ср. школы № 64 г. Санкт-Петербурга;

**Пономарев Анатолий**

— учащийся VIII класса лицея № 2 г. Троицка Московской обл.;

**Кашменский Евгений**

— учащийся X класса ср. школы № 130 г. Новосибирска;

**Ким Валерий**

— учащийся X класса лицея № 2 г. Нальчика;

**Захаров Дмитрий**

— учащийся IX класса ср. школы № 19 г. Орла;

дипломы третьей степени получили:

**Марков Сергей**

— учащийся IX класса лицея № 2 г. Троицка Московской обл.;

**Четвертков Станислав**

— учащийся X класса ср. школы № Московской обл.;

**Теллонен Артем**

— учащийся X класса ср. школы № 17 г. Петрозаводска;

**Шульженко Андрей**

— учащийся X класса СУНЦ МГУ.

Автор задачи — В. В. Прохоров, канд. физ-мат. наук, институт математики и механики Уральского отделения РАН, г. Екатеринбург

## ЗАДАЧА «БОЛЬНИЦА»\*

I тур, 25.02.94

### Условие задачи.

В процедурном кабинете  $N$  кушеток. Каждый из  $M$  пациентов должен принять на каждой кушетке процедуру. Некоторые пациенты могут иметь некоторое (одно и то же) кожное заболевание (кто именно — неизвестно, но их не более  $K$ ). Некоторые из  $N$  кушеток являются источником этого заболевания (таковых наверняка не больше  $L$ ). При соприкосновении с такой кушеткой пациент заражается.

Заболевание передается не только при непосредственном контакте с больным, но и при контакте с зараженными поверхностями предметов, которые становятся таковыми при соприкосновении с зараженными поверхностями других предметов.

Необходимо исключить инфицирование здоровых пациентов и увеличение количества зараженных кушеток после принятия процедур.

Для решения проблемы можно использовать стерильные простынки, поскольку сквозь ткань данная инфекция не передается. Из-за дефицитности простынок требуется определить алгоритм их перестилания на кушетках, обеспечивающий выполнение указанных требований при минимально возможном расходе простынок. Разрешается стелить на кушетку друг на друга не более  $P$  простынок.

Компьютерная программа должна:

запрашивать параметры  $M, N, K, L, P$ ;  
выдавать минимальное требуемое количество простынок;

· выдавать оптимальный в указанном смысле процесс застилания кушеток и принятия процедур, где

⟨Процесс⟩ ::= ⟨Процедура⟩, ⟨Процедура⟩, ...

⟨Процедура⟩ ::= ⟨Номер пациента⟩ / ⟨Простынка⟩ / ⟨Простынка⟩ / ... / ⟨Номер кушетки⟩

⟨Простынка⟩ ::= ⟨Номер простынки⟩  
⟨Ориентация простынки⟩

Здесь ⟨Ориентация простынки⟩ — «а», если простынка лежит лицевой стороной вверх, и «b» — в противном случае.

### Пример.

При  $M=1, N=3, K=0, L=1, P=2$  программа должна выдавать минимальное количество «2» и, например, последовательность  $1/1a/1, 1/2a/2, 1/2a/1b/3$ .

### Примечания.

1. Контакт больного пациента с зараженной поверхностью ни к каким неприятным последствиям не приводит.

2. Общее время проведения процедур, их очередность, время ожидания пациентами очереди значения не имеют.

### Оценка задачи.

Максимальное количество баллов — 50.

\* От редакции. К сожалению, задача приводится без решения. Печатается в соответствии с текстом, любезно предоставленным В. М. Кирюхиным, председателем жюри.

С. Г. Волченков,

доцент Ярославского гос. педагогического института

## ЗАДАЧА «ПАРКЕТ»

I тур, 25.03.94

**Условие задачи.** Комнату размером  $N \times M$  единиц требуется покрыть одинаковыми плитками паркета размером  $2 \times 1$  единиц без пропусков и наложений ( $M \leq 20$ ,  $N \leq 8$ ,  $M, N$  — целые). Пол можно покрыть паркетом различными способами. Например, для  $M=2$ ,  $N=3$  все возможные способы укладки приведены на рис. 1:

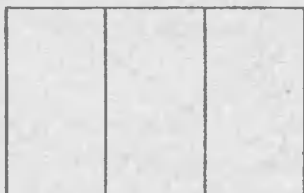
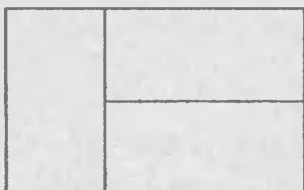
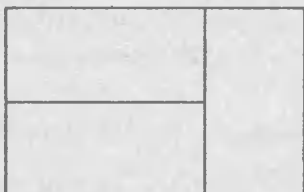


Рис. 1

**Задание.** Требуется определить количество всех возможных способов укладки паркета для конкретных значений  $M \leq 20$  и  $N \leq 8$ . Решением задачи является таблица, содержащая 20 строк и 8 столбцов.

Элементом таблицы является число, являющееся решением задачи для соответ-

ствующих  $M$  и  $N$ . На месте ненайденных результатов должен стоять символ «\*».

На рис. 2 приведен пример таблицы:

	1	2	3	4	5	6	7	8
1	0	1	0	*	*	*	*	*
1	1	2	3	*	*	*	*	*
1	*	*	*	*	*	*	*	*
...	...	...	...	...	...	...	...	...
20	*	*	*	*	*	*	*	*

Рис. 2

Таблица должна быть выровнена по столбцам и помещена в текстовый (ASCII) файл с именем «имя.RES».

**Примечание.** Результат решения задачи будет оцениваться по содержимому файла «имя.RES».

**Оценка задачи.** Максимальное количество баллов — 50. Чем больше правильно заполненных элементов таблицы, тем выше результат.

Предложенная задача имеет одну особенность, отличающую ее от большинства олимпиадных задач. Вообще говоря, от участников олимпиады не требовалось написать программу, результатом работы которой явился бы файл «Имя.RES». Этот файл надо было представить в качестве результата собственной работы, а каким образом он получен — все равно. Более того, часть результатов в требуемой таблице можно было получить с помощью несложных рассуждений и, соответственно, увеличить машинное время для остальных расчетов.

Большинство участников отметили очевидный факт, что при нечетных  $M$  и  $N$  ответ равняется нулю. Также легко сделать вывод, что при  $M=1$  и четном  $N$  ответ равен 1. Нетрудно определить симметричность искомой таблицы относительно главной диагонали. Чтобы заполнить второй столбец, надо выполнить не очень сложные рассуждения и вычисления. Так, если за  $S(i)$  обозначить количество способов, которыми можно уложить паркет в комнате размером  $2 \times i$ , то легко вывести рекуррентную формулу:

$$S(i) = S(i-1) + S(i-2), \quad (1)$$

верную для всех  $i > 2$ . В самом деле, у одной из стен комнаты плитки паркета должны быть уложены либо как на рис. 3а, либо как на рис. 3б.

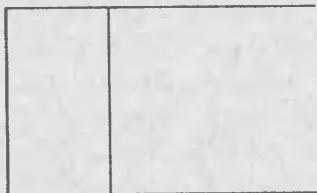


Рис. 3а

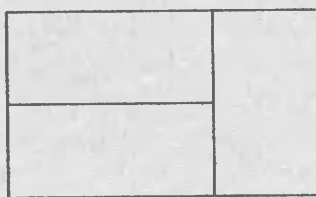


Рис. 3б

Одному из этих вариантов соответствует первое, а другому — второе слагаемое формулы (1). Теперь вместе с очевидными равенствами:  $S(1) = 1$ ,  $S(2) = 2$  — мы получим в качестве чисел, стоящих во втором столбце искомой таблицы, ряд Фибоначчи.

Некоторые участники пытались вывести подобную рекуррентную формулу для

третьего столбца, но почти все потерпели неудачу.

Итак, осталось рассмотреть случаи, когда  $M > 2$ ,  $N > 2$ .

Многие участники продемонстрировали «высший пилотаж» в организации перебора вариантов на компьютере и получили результаты, например, при  $M=8$  и  $N=6$  (количество вариантов равно 4213133). К сожалению, никакие ухищрения в переборе не могли продвинуть участников за барьер очень больших чисел. Для заполнения оставшихся элементов таблицы необходимо было применить методы, отличные от переборных. Ниже приводится описание авторского решения, которое основывается на методе динамического программирования. Рассмотрим какую-нибудь комнату, выложенную паркетом (рис. 4).

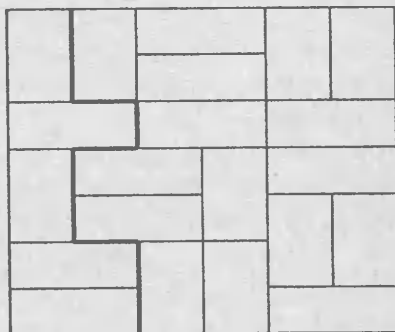


Рис. 4

Попытаемся «разрезать» эту комнату на части для исследования возможных способов паркета в каждой из них. Договоримся проводить разрез по выбранной нами вертикальной черте так, чтобы не трогать отдельных плиток паркета, лежащих на пути. Для этого каждую такую плитку будем обходить справа. Кроме того, отсекать от комнаты будем только самую узкую полоску слева. На рис. 4 такое сечение выделено жирной чертой.

Какие еще сечения, соответствующие

Другим укладкам паркета, могут быть получены и сколько их всего? Ответить на этот вопрос нетрудно, так как, двигаясь сверху вниз, на каждом шаге мы либо обходим выступающую клетку, либо нет. Чтобы дойти до низа, мы должны преодолеть один из  $2N$  вариантов. То, что некоторые из них не могут быть реализованы на практике, нас пока не интересует.

Все варианты легко занумеровать, так как каждый почти в чистом виде представляет собой запись двоичного числа: обход плитки соответствует «1», отсутствие обхода — «0». Так, сечение на рис. 4 получит двоичный номер 0010011, что соответствует десятичному числу 19. Равный край комнаты соответствует сечению номер 0.

Обозначим парой  $(i, k)$  комнату размером  $N \times 1$ , левый край которой не ровный, а представляет собой сечение с номером  $k$ . Обозначим через  $T(i, k)$  количество вариантов укладки паркета в такой комнате.

Данная нам задача превращается в нахождение  $T(M, 0)$ .

Некоторые значения  $T(i, k)$  вычисляются просто, например  $T(0, 0) = 1$ , так как комната нулевой ширины укладывается паркетом (при этом не используется ни одной плитки),  $T(0, j)$  для всех ненулевых  $j$  равны нулю, так как соответствующих комнат не существует.

А теперь попытаемся найти все  $T(i, k)$ , предположив, что мы знаем значения  $T(i-1, j)$  для всех  $j$ . Если нам это удастся, то, основываясь на вышеуказанных значениях  $T(0, j)$ , мы получим все  $T(1, j)$ , затем  $T(2, j)$  и т. д. до нахождения  $T(M, 0)$ .

Для каждой комнаты  $(i, k)$  рассмотрим все комнаты  $(i-1, j)$ . При этом сечение с

номером  $J$  может быть либо совместимым с сечением  $k$ , если существует такая укладка комнаты  $(i, k)$  паркетом, при которой от этой комнаты можно отсечь комнату  $(i-1, j)$ , либо сечение  $j$  несовместимо с сечением  $k$ . Заметим, что информацию о совместимости сечений можно получить сразу, исследовав, какие сечения  $k$  можно получить, добавляя к сечению  $j$  слева различные множества плиток паркета. Их не так уж много, так как добавлять можно не более одного слоя плиток.

Если мы знаем, какие сечения совместимы друг с другом, то остается вычислить все  $T(i, k)$  по рекуррентной формуле:

$$T(i, k) = \sum_{\substack{\text{по всем} \\ \text{сечениям } j, \\ \text{совместимым} \\ \text{с } k}} T(i-1, j), \quad (2)$$

так как в задаче было задано  $N \leq 8$ , то всего сечений не более 256, т. е. для решения задачи в самом трудоемком случае ( $M=20$ ,  $N=8$ ) требуется заполнить таблицу  $T(i, k)$  размером  $20 \times 256$ . Первая строка этой таблицы вычисляется в соответствии с замечаниями о значениях  $T(0, j)$ . Каждая последующая строка вычисляется по формуле (2). При хорошей реализации программа вычисления всех значений файла «имя.RES» работает 30—40 минут.

Отметим еще один небольшой подводный камень задачи. Чтобы получить правильный результат при больших  $M$  и  $N$ , кроме всего прочего, надо организовать вычисления с повышенной точностью. Ответ при  $M=20$ ,  $N=8$  оказывается равным 3547073578562247994, т. е. содержит 19 значащих цифр!



Е. В. Андреева

## ЗАДАЧА «БУКВОЕД»

II тур, 27.03.94, г. Троицк

Условие задачи. Во входном файле содержится закодированное изображение электронного табло, состоящего из 25 строк и 80 столбцов лампочек. Известно, что на табло высвечивались одна или несколько заглавных печатных букв: А, В, Ж, Л, М, О, С, Ф, Ю. Символы, отличные от перечисленных букв, на табло отсутствовали. Пример фрагмента табло представлен на рис. 1, где звездочки соответствуют горящим лампочкам.

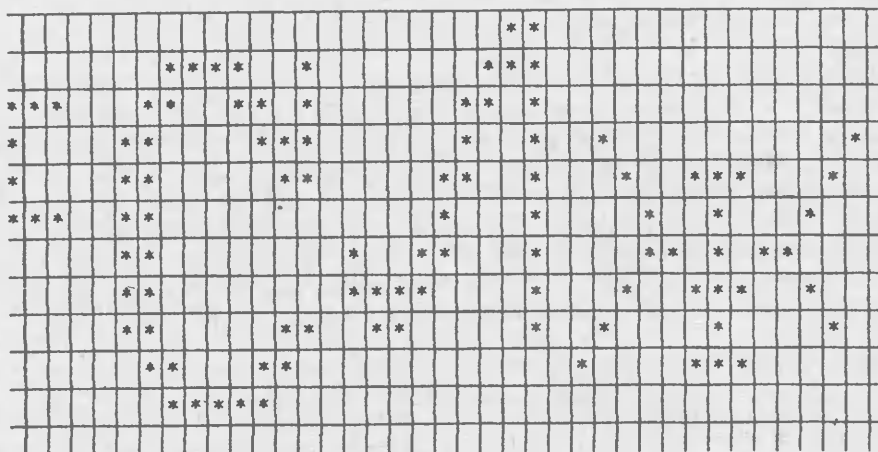


Рис. 1

Две горящие лампочки, соседствующие на табло по горизонтали, вертикали или диагонали, принадлежат одной и той же букве. Буквы могут быть любого размера, толщины, начертания («рукописные» буквы не допустимы). Буквы расположены вертикально. Изображения букв не касаются и не пересекаются. «Линии», образующие буквы, не имеют разрывов и полостей. На рис. 2 приведены примеры недопустимого изображения букв.

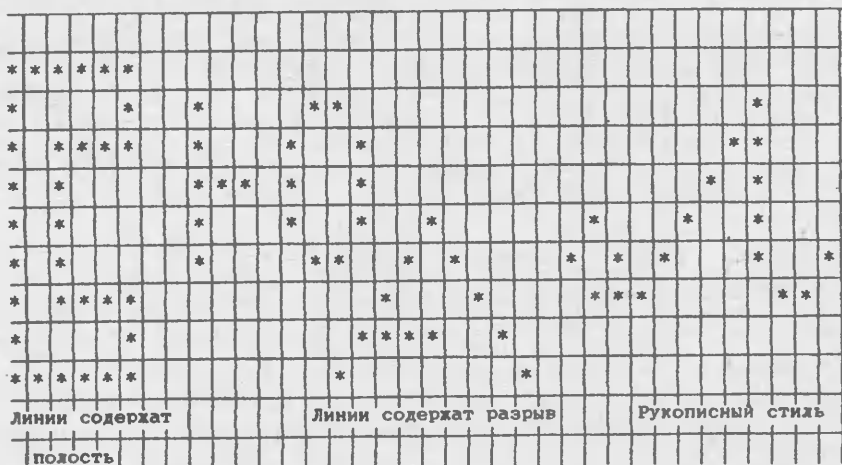


Рис. 2



каждая из перечисленных букв может как встречаться несколько раз, в том числе и в различных модификациях, так и отсутствовать.

6. Вам предоставлен файл test.dat, в котором закодировано изображение рис. 1.

7. При решении задачи рекомендуется пользоваться описанной выше структурой данных (сериями) для хранения и обработки информации.

### Решение подзадачи отделения объектов

Прежде чем приступить непосредственно к распознаванию букв, необходимо определить, к какому объекту относится та или лампочка. В рамках данного электронного табло эту задачу довольно легко решить с помощью двумерного массива  $80 \times 25$  и рекурсивной процедуры «закраски объекта»: помечаем лампочку и запускаем эту же процедуру для всех соседних непомяченных лампочек. Однако этот подход неприменим к большому табло или, например, к графическому экрану ( $640 \times 350$ ), так как двумерный массив такого размера хранить в памяти персонального компьютера без усложнения структуры данных не представляется возможным.

Поэтому будем хранить исходные данные так, как это сделано во входном файле, — с помощью описания всех серий лампочек. Пусть табло содержало  $K$  серий. Нетрудно убедиться, что для табло нашего размера, при условии отсутствия на нем каких-либо символов, кроме перечисленных выше букв, максимальное количество серий равно 960 (если все табло будет состоять из букв  $O$  высотой 25 строк и шириной 3 столбца, с толщиной «линий» в один символ, то всего на табло будет 20 букв, каждая из которых будет содержать 48 серий). Каждую серию будем хранить с помощью 3 чисел, которыми она характеризуется (см. описание входных данных), а также номера объекта, к которому эта серия принадлежит, а пока это неизвестно, данную характеристику каждой серии заменим числом 0. В примере 1 приведена структура исходных данных и процедура нумерации серий, согласно номеру объекта, к которому данная серия принадлежит. Фрагменты программы написаны на языке Паскаль.

Пример 1.

```
const kmax=960;
var
  s: array[1..kmax] of record
    lin, fin, len, num: integer
  end;
  k, l: integer; {k-реальное количество серий}
function near(i1, i2: integer): boolean;
{данная функция определяет, являются ли входные серии соседними в одном и том же объекте}
begin
  near := (abs(s[i1].lin - s[i2].lin) = 1) and
    ((s[i1].beg >= s[i2].beg - 1) and (s[i1].beg <= s[i2].beg + s[i2].len) or
    (s[i2].beg >= s[i1].beg - 1) and (s[i2].beg <= s[i1].beg + s[i1].len))
end;
procedure fill(i: integer; var l: integer);
{данную процедуру, которая "отслеживает" один объект, можно написать эффективнее, но в рамках исходного размера табло это лишь усложнит текст программы}
var j: integer;
begin
  s[i].num := 1;
  for j := 1 to k do
    if (s[j].num = 0) and near(i, j) then fill(j, l)
  end;
end;
procedure number;
var j: integer;
```

```

begin
  l:=0;
  for j:=1 to k do
    if s[j].num=0 then
      begin
        l:=l+1;
        fill(j,l)
      end
  end
end; (по окончании работы данной процедуры в переменной l будет
      находиться число различных букв на табло)

```

После выполнения процедуры number желательно отсортировать массив  $s$  в порядке возрастания значений в поле  $num$ , не меняя при этом взаимного расположения серий в одном и том же объекте.

### Решение подзадачи подсчета количества контуров у объектов

Все буквы, перечисленные в условии задачи, можно разделить на три группы, используя один безусловный признак: количество замкнутых контуров, окружающих букву. Так, буквы Ж, Л, М, С имеют лишь один внешний контур, буквы А, О, Ю имеют два-три контура. Поэтому, подсчитав количество контуров у буквы, мы существенно упрощаем задачу распознавания.

Будем считать, что все лампочки, с которых начинается или которыми заканчивается любая серия, образуют контуры объекта. Если серия состоит из одной лампочки, то последняя входит в контур дважды. Каждый объект имеет одну замкнутую внешнюю часть контура и может иметь несколько замкнутых внутренних частей контура. Таким образом, во все контуры объекта, состоящего из  $k_1$  серий, войдет ровно  $2 \times k_1$  лампочек.

Сначала мы отследим внешний контур, и, если не у всех серий и начало, и конец вошли в него, значит, у объекта есть еще внутренние контуры, которые мы будем отслеживать по очереди. Любой контур, внешний или внутренний, отслеживается следующим образом. Ищем начало или конец какой-либо серии, не вошедший в контур, и запоминаем его координаты, как координаты начала контура. Если это начало серии, то начнем двигаться вверх, а если конец, то вниз. При движении вверх мы будем переходить на начало самой левой серии, расположенной строкой выше текущей серии и соседствующей с ней. Смена движения вверх на движение вниз может произойти в двух случаях: 1) выше текущей серии, соседствующей с ней; 2) если серия, находящаяся левее текущей в той же самой строке, соседствует с той же самой серией в строке выше, что и текущая серия. Примеры, иллюстрирующие эти два случая, изображены на рис. 4. В первом случае мы переходим на конец текущей серии, а во втором — на конец серии, расположенной

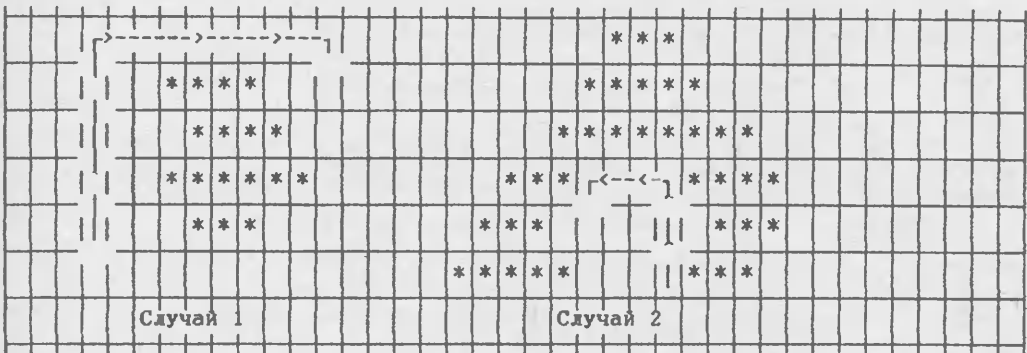


Рис. 4



```

                j:=kk; yc:=yc-1; xc:=s[j].beg
            end
        end;
    down: begin
        s[j].num:=s[j].num+2;
        kk:=i2+1;
        repeat {ищем соседствующую серию ниже справа}
            kk:=kk-1
        until near(kk,j) or (s[kk].lin=s[j].lin);
        if (s[kk].lin=s[j].lin) {серию не нашли} then
            begin xc:=s[j].beg; mov:=up end
        else
            if (j<k) and (near(j+1,kk) then {сдв-àé 2}
                begin
                    j:=j+1; xc:=s[j].beg; mov:=up
                end
            else begin
                    j:=kk; yc:=yc+1; xc:=s[j].beg+s[j].len-1
                end
            end;
        until (xc=x) and (yc=y)
    end;
    for i:=i1 to i2 do
        s[i].num:=11; {восстанавливаем номер объекта}
    end;
end;

```

### Распознавание букв В и Ф

Теперь, когда мы можем подсчитать количество контуров у буквы, становится ясно, что символы, имеющие три контура, могут быть только буквами В или Ф. Так что, согласно условию задачи, можно, не производя дальнейшего распознавания, выдать ответ с помощью этих двух символов. Однако попробуем определить сравнительную топологию этих двух букв. Для этого составим карту количества серий в каждой строке буквы сверху вниз, а именно подсчитаем количество серий в первой строке лампочек, изображающих интересующую нас букву, и запишем результат в первый элемент вспомогательного одномерного массива. Затем рассмотрим серии второй строки, если их количество то же, что и в первой, то перейдем к третьей строке, в противном случае запишем новый результат во второй элемент массива, отображающего карту символа, будем записывать лишь число, отличное от предыдущего элемента массива. Например, «стандартная» буква В может иметь карту: 1, 2, 1, 2, 1, а буква Ф — 1, 3, 1. В общем же случае карты будут довольно разнообразны.

Выделим такое свойство карт серий для этих двух букв, которое сделает такие символы однозначно распознаваемыми. Так, если карта начинается с трех серий, то это буква Ф; если же карта вообще не содержит числа три, то это буква В. Если же карта начинается с В, после него всегда будет следовать число 2, а для буквы Ф после тройки в карте серий будет одно из чисел — 1 или 5. Таким образом, распознавание данных букв можно считать завершенным.

### Распознавание букв А, О, Ю

Очевидно, что все перечисленные в данном пункте буквы всегда имеют два контура. Распознавание этих символов между собой с помощью карты серий не столь очевидно, как в предыдущем случае, поэтому рассмотрим другой способ описания топологии букв.

Рассматривая один из перечисленных символов построчно сверху вниз, найдем первую строку, в которой находится ровно 2 серии. Соединим эти серии дополнительным рядом лампочек, расположив их строкой выше, так, чтобы начало этого ряда касалось левой из двух серий, а конец — правой. При этом буквы А и О топологических изменений не получают (они вообще могут остаться неизменными). Аналогичную процедуру сделаем, рассматривая буквы в направлении снизу вверх, что опять же не принесет изменений букве О. С полученным в результате этих двух преобразований объектом обратимся к процедуре подсчета количества контуров. Очевидно, что буква О по-прежнему будет иметь 2 контура, преобразованная буква А — три контура, а дополненная буква Ю — четыре контура.

Существуют, однако, экзотические случаи написания буквы Ю, не попадающие под данную схему. Так, если левая палочка этой буквы и сверху, и снизу существенно короче, чем «бублик», то топология О будет обязательно содержать в карте серий число три (в общем случае, это для буквы Ю не является обязательным, она даже может иметь карту серий 1, 2, 1, совпадающую с буквой О полностью). Если же палочка у Ю короче только сверху, то эту букву можно спутать с буквой А, но такая буква Ю в качестве фрагмента своей карты серий будет содержать последовательность 3, 2, 3, а буква А, скорее всего, не будет содержать в карте серий тройку вообще.

### Распознавание букв Ж, Л, М, С

Написание вариантов этих букв столь различно, что применение рассмотренной выше формализации является затруднительным. Тем не менее существует довольно простой способ, позволяющий отличить буквы Ж, Л, М друг от друга. Так, если при подсчете числа контуров (а данные буквы имеют лишь один внешний контур) мы сосчитаем также, сколько раз изменялось направление движения с направления «вверх» на направление «вниз», то легко заметить, что у буквы Ж это произойдет 5 или более раз, у буквы М — 4 раза, а у буквы Л — 2 или 3 раза. К сожалению, у буквы С, которая также не содержит внутренних контуров, смена направления движения может происходить сколько угодно раз в зависимости от ее написания: от 1 до 5. Однако если смена движения вверх на противоположное у буквы С произошла два и более раз, то такая буква обязательно будет содержать в карте серий фрагмент 2, 1, 2, не характерный ни для одной из букв, рассматриваемых в данном пункте. Таким образом, распознавание букв, перечисленных в условии задачи, можно считать завершенным.

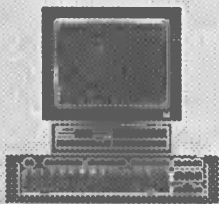
### Заключение

В этой статье предложено четыре подхода к решению задачи распознавания образов, которые ни в коей мере нельзя считать исчерпывающими. Более того, решение исходной задачи стало возможным лишь благодаря условию, согласно которому какие-либо другие объекты, помимо перечисленных, в распознавании участвовать не могли. То есть рассмотренные объекты были формализованы не абсолютно, а относительно друг друга.

Помимо рассмотрения задачи распознавания образов в работе решены: 1) задача разделения объектов для больших размерностей и 2) задача отслеживания всех контуров произвольного объекта. Данные задачи имеют самостоятельную ценность.

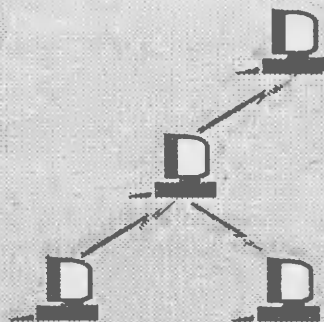


*"Комедж" ⇒ услуги ⇒ успех и удача!  
Работа со школами многое значит!  
Программы, компьютеры ...  
- нужен деловой совет?  
Для нас ничего невозможного нет!*



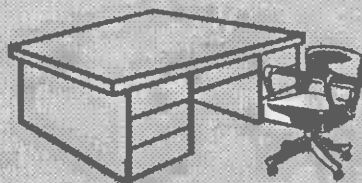
- ▶ Компьютерные классы на базе IBM-совместимой техники "под ключ"
- ▶ Мультимедиа классы

- ▶ Проектирование и установка учебных локальных сетей
- ▶ Организация абонентного обслуживания в глобальных телекоммуникационных сетях



- ▶ Учебные и административные программы

- ▶ Мебель для учебных заведений и офисов



*Доставка техники и мебели осуществляется по Москве и Московской области*

*Адрес: 107005, Москва, Валовский пер., д.11, Фирма "Комедж"  
Телефон: 265-62-65 Телеракс: 265-62-65*



# ИНФОРМАТИКА В МЛАДШИХ КЛАССАХ

Ю. М. ГОРВИЦ,

ассоциация «Компьютер и детство», г. Москва

## ЗАЧЕМ НУЖНЫ КОМПЬЮТЕРЫ В ДОШКОЛЬНЫХ УЧРЕЖДЕНИЯХ?

Фундаментальной характеристикой развития человеческой цивилизации является получение, накопление, обработка и потребление информации. В информатизированном обществе без овладения начальной компьютерной грамотностью и умения использовать компьютерные средства для решения определенных задач немислима реализация творческого потенциала человека в современной науке, культуре, производстве, деловой и иных сферах жизни.

Новые подходы к пониманию роли компьютерной техники и других средств новых информационных технологий (НИТ), включающих в себя современные средства коммуникаций, интерактивное видео, системы видеотекста и мультимедиа, радиокomпьютеры и т. п., а также методы их использования разрабатываются на фоне более общих исследований, связанных с изменениями в обществе и, в частности, в образовании, как одном из социальных институтов, непосредственно влияющем на эти изменения.

Современное общество характеризуется, с одной стороны, нестабильностью, быстрой изменчивостью и трудной предсказуемостью, с другой стороны, все большей «открытостью», взаимопроникновением накопленных знаний и опыта. Проекты, направленные в будущее, требуют громадных инвестиций, которые предполагают межгосударственную кооперацию. Соответственно, возникает необходимость в новых знаниях, информационной грамотности, способности самообразовываться.

Так возникает социальный заказ на новый тип образования — инновационный, в котором информационные технологии призваны сыграть системообразующую, интегрирующую роль.

## Персональные компьютеры в начальном образовании

Проблема использования НИТ, основанных в первую очередь на компьютерной технике, на дошкольном уровне образования является относительно новой и требует проведения широкого комплекса исследований междисциплинарного характера, поскольку затрагивает различные аспекты деятельности детей в новой среде. В первую очередь это изучение особенностей психолого-педагогического контекста использования компьютеров.

«Компьютер не может быть включен в среду детского сада чисто механически, — считают наши известные ученые С. Л. Новоселова и Е. В. Зворыгина. — Развитие дошкольника обогащается естественным образом лишь в условиях тесной связи содержания компьютерной игры с уровнем развития других форм детской деятельности». Профессор МГУ О. К. Тихомиров полагает, что при этом необходимо прогнозировать и учитывать психологические последствия компьютеризации, которые появляются в психике ребенка и относятся к эмоциональным, мотивационным и когнитивным процессам.

Различным психолого-педагогическим аспектам использования компьютера в деятельности детей дошкольного возраста посвящены исследования последних лет, проводимые отечественными и зарубежными специалистами.

С мая 1988 г. в Балтиморе (штат Мэриленд, США) началась реализация проекта, целью которого было использование компьютеров в дошкольном воспитании. В ходе эксперимента в шесть дошкольных заведений были переданы по два компьютера с программным обеспечением. Через

некоторое время благодаря восторженному отклику со стороны детей, родителей, персонала и спонсоров этот опыт распространился на города еще шести штатов.

Следует отметить, что в США, странах Западной Европы и Японии практика использования персонального компьютера (ПК) маленькими детьми связана еще и с широким распространением дешевых домашних компьютеров, а также с производством огромного числа компьютерных игр для досуга детей и взрослых в домашних условиях.

Этот класс компьютерных программ насчитывает сотни наименований, однако большинство реализуемых через розничную торговлю программ содержат небольшое количество относительно тривиальных идей (погоня, лабиринт, имитация военных действий) и отличаются лишь внешним оформлением и элементами пользовательского сервиса. Рассчитаны они на широкий возрастной диапазон — от младших школьников до взрослых, однако в некоторые несложные игры с успехом играют и дети дошкольного возраста.

В разработке программных средств для детей естественным следует считать появление обучающих и развивающих программ. В последние годы количество программ для малышей и число фирм-изготовителей таких программ увеличивается.

С 80-х гг. в связи с появлением относительно недорогих ПК, бурным развитием программного обеспечения (ПО) начался процесс внедрения ПК в различные сферы деятельности человека: промышленность, науку, спорт и, естественно, в сферу образования, при этом основным потребителем техники оставались высшие учебные заведения. Но уже в 1980—1983 гг. персональные компьютеры в больших количествах стали появляться в школах многих развитых стран мира, а примерно с 1985 г. они стали широко использоваться в начальной школе и детских дошкольных учреждениях.

В США уже к 1985 г., по данным обследования министерства образования, 86% школ имели по меньшей мере одну ЭВМ, а на один компьютер приходилось 72 учащихся. Среднее же количество составляло восемь машин на одну школу. К 1990 г. количество учащихся на один ПК уменьшилось до 14.

Динамику роста числа компьютеров в школах США можно проследить по следующим данным журнала «Personal Computers»:

Год	Млн. шт.
1981	0,127
1982	0,263
1983	0,492
1984	0,828
1985	1,275
1986	1,996
1987	2,535
1988	3,272
1989	4,060
1990	4,872

Не будем забывать и о том, что большинство семей в США имеют домашний компьютер.

Ситуация, сложившаяся в области компьютеризации образования в Японии, оказалась неожиданной — в этой высокоразвитой в техническом отношении стране на один компьютер в 1989 г. приходилось 80 учащихся, в то время как в США эта цифра в 1989 г. была равна 20. (Business Week, апрель 1992). Японское руководство, обеспокоенное отставанием в образовании по сравнению со сферами науки и технологии производства, предприняло значительные усилия по преодолению такого положения. Общие затраты на компьютеры и программное обеспечение для детей, в соответствии с пятилетним планом, реализация которого началась в 1991 г., составили 1,7 млрд. долларов, при этом министерство образования выделило на указанные цели 400 млн. долларов, остальные средства были вложены местными властями и крупными фирмами.

С развитием компьютеризации получили применение машины, оснащенные компакт-дисками большой емкости (CD-ROM), — так называемые мультимедиа, в которых для учебных целей используется программное обеспечение нового поколения, построенное по принципу интеграции различных форм представления информации (графической, текстовой, звуковой и др.).

Аналогичные системы применяются в США и странах Западной Европы, но их пока еще высокая стоимость ограничивает использование таких систем в образовании, и в особенности для детей младшего возраста.

Школы Великобритании с 1986 г. оснащаются главным образом компьютерами типа BBC. Типичным является класс, имеющий 6 или 12 компьютеров, объединенных в локальную сеть. Весьма большое количество микрокомпьютеров используется в домашних условиях.

Во Франции благодаря развитой государственной системе образования к 1987 г. все начальные школы были оснащены компьютерами. С 1984 г., когда Министерством национального образования был утвержден план «Информатика для всех», по 1987 г. в начальных школах была установлена 33 171 машина, при этом общая стоимость плана достигла 1,79 млрд. франков.

Широко используются компьютеры в начальном образовании детей 5—8 лет в Дании, Швейцарии, Болгарии, Испании, КНР, Южной Корее и других странах.

В разных странах находят воплощение различные модели обучения детей дома и в школе, в том числе в младших классах и детских садах, ориентированных на применение компьютеров. Это можно объяснить и различием в инфраструктуре национальных систем образования, и историко-культурными традициями в воспитании и образовании детей. Но вместе с тем прослеживается и определенное сходство в подходах к применению НИТ в начальном образовании.

Во многих странах исследователи пытаются найти ответы на естественные вопросы:

- Зачем все это обилие компьютеров используется в начальном образовании?
- Какие программные средства разрабатываются и применяются?
- Дает ли эта технология что-нибудь детям?

Специалисты образования, как показывает анализ зарубежной и отечественной литературы, идут следующими путями в применении НИТ:

1. Компьютер как средство для обучения различным дисциплинам, как инструмент поддержки предметных уроков и других видов занятий.
2. Использование компьютера для выполнения учебных и реальных задач и для реализации различных видов деятельности.
3. Компьютер и информационная технология как объект изучения в рамках предмета «Информатика», «Основы информатики и вычислительной техники», «Информационная культура» и т. п.
4. Компьютер как средство развития ребенка.

Здесь мы не обсуждаем такую существенную как по времени, так и по затратам на производство область, как компьютер для досуга, поскольку образовательные цели в развлекательных компьютерных про-

граммах практически не ставятся и не достигаются.

Рассмотрим подробнее особенности использования НИТ в каждом из этих направлений.

### Компьютер как средство для обучения

За рубежом для детей младшего возраста, главным образом младших школьников, разрабатывается немало обучающих программ. В основном это внешне привлекательные программы по обучению чтению и письму, а также математике.

Из нескольких десятков учебных программных средств каталога фирмы IBM за 1989 г. для дошкольников предназначены всего 5 программ по грамматике и обучению чтению, 4 по математике и всего одна программа, которую в нашей терминологии можно назвать общеразвивающей (способствует усвоению детьми понятий противоположного направления, симметрии, соответствия цветов, осознания пространственных представлений).

В каталоге «Hartley Educational Courseware» (1989 г.) приведены среди других 34 программы для детей в возрасте 4—7 лет, из которых 19 программ обучают грамматике, 7 — математике и 8 — развивающих. При этом к развивающим относятся программы, связанные с классификацией объектов, нахождением соответствия, выделением форм и цвета и другие, т. е. такие, которые в отечественной дошкольной дидактике прямо относятся к теме «Развитие начальных математических представлений».

В каталоге фирмы «Wings for Learning», специализирующейся на продаже педагогических программ, всего 5 учебных программ ориентированы на детей дошкольного возраста.

Однако в одном из последних и наиболее полном каталоге «High/scope Buyer's Guide to Children's Software 1992»<sup>\*</sup> описаны уже более 500 программ для детей от 3 до 7 лет, разработанные 92 фирмами для компьютеров моделей Apple, IBM, Commodore-64, Macintosh, Apple IIGS, Atari и Amiga.

<sup>\*</sup> Мы предполагаем подготовить и опубликовать в журнале «Информатика и образование» обзор программ, опубликованных в этом каталоге. — *Прим. ред.*

## Компьютер как инструмент

Практически во всех начальных школах США, Канады, России и других стран педагоги на занятиях в компьютерных классах стараются попутно с основными дидактическими заданиями научить учащихся пользоваться компьютером как инструментом для выполнения других задач.

Акцент при этом делается на обучение дошкольников и младших школьников умению пользоваться клавиатурой компьютера. С помощью простых дидактических приемов объясняется также назначение различных устройств компьютера, поэтому даже дети 6—7 лет могут осознанно употреблять такие сугубо технические термины, как «дисплей», «диск», «клавиша» или «кнопка», «принтер» и т. д. Для них это слова из привычной «среды обитания», они ничем не отличаются по трудности от слов «телефон», «магнитофон», «радио», а теперь и «видео», «факс», и других, уже не требующих кавычек слов, относящихся к информационной технике.

В различных странах в начальных классах общеобразовательных школ, где есть вычислительная техника, педагоги используют специальные программы-инструменты для развития у детей навыков решения практических задач с помощью современных технических средств. Так, дети используют простейшие программы набора текстов не только для изучения расположения клавиш на клавиатуре, но и для сочинения и печатания рассказов, стихов, сказок, писем друзьям и т. д. Малыши с помощью несложных «детских» музыкальных программ-редакторов не только постигают основы нотной грамоты (предметное обучение), но и сочиняют песенки, небольшие музыкальные фрагменты, которые могут потом с удовольствием слушать сами или с друзьями, педагогами или родителями, а также использовать для сопровождения создаваемых ими же картинок и мультфильмов. Разнообразные графические изображения (геометрические и произвольные узоры, орнаменты, рисунки) и даже «оживающие» серии таких рисунков (анимации) дети младшего возраста также могут создавать с помощью специально разработанных программ, таких, как «KidPix» фирмы Broderbund (США), «Море» и «Лес» Ассоциации КИД (Россия), некоторых программ системы DIL (Канада).

Текстовые, графические и музыкальные редакторы используются и в школах Японии. В Великобритании программа обучения информатике содержит раздел, связанный с применением ЭВМ в различных областях человеческой деятельности, и т. д.

Известны проекты, в рамках которых дети смогут обмениваться «компьютерной продукцией» — рисунками, письмами, автопортретами — с помощью глобальных компьютерных сетей со сверстниками из других стран. Участниками такого рода проектов станут и малыши нашей страны.

## Компьютер как элемент обучения информатике

В 85—90-х гг. главной линией в информатизации школьного образования являлось знакомство детей с компьютерной техникой как объектом изучения, принципами, методами и средствами структурирования задач и их реализации на псевдо- или реальных алгоритмических языках программирования.

Курс «Основы информатики и вычислительной техники» главным образом ориентировался на безмашинный (позднее — с программной поддержкой на различных моделях персональных компьютеров) вариант преподавания в массовой школе, а методика его преподавания явно была нацелена на развитие у школьников так называемого алгоритмического (иногда «логического») типа мышления. При этом вводились такие понятия, как «алгоритм», «исполнитель», «цикл» и «ветвление», «типы данных» и т. д.

Целью большинства программ такого рода как у нас, так и в зарубежных странах было главным образом знакомство старших школьников с компьютерной техникой и основами программирования. Так, в Великобритании один из разделов действующей в эти годы программы по информатике предполагал обучение программированию с использованием языков Бейсик и Комал. Во Франции и США в этот период также преобладали аналогичные учебные и компьютерные программы.

Многие специалисты в последующем пытались перенести этот подход практически без изменения содержания программ на более младший возраст (что не бесспорно), в результате чего создавались соответствующие программно-методические средства.

Наиболее ярким примером такого подхода в разработке «компьютерных сред» для детей младшего (8—12 лет) возраста может служить в США система ЛОГО, автором которой является известный ученый С. Пейперт, а в России — «Роботландия», созданная группой ученых под руководством профессора Ю. А. Первина. Эти системы отличаются от многих других педагогических программных средств концептуальной

выдержанностью, педагогической обоснованностью и эффективной реализацией.

Система ЛОГО получила широкое распространение во многих странах мира. Для примера приведем мнение болгарского профессора Р. Николова, одного из многих специалистов, пропагандирующих эту систему: «ЛОГО — это среда обучения, учитывающая психологию ребенка с целью формирования алгоритмического мышления. При помощи такой среды можно решить многие общеобразовательные задачи, сформировать определенный стиль мышления, позволяющий принять и освоить основы новой информационной культуры».

Систему ЛОГО многие педагоги используют как инструмент детского творчества, как средство развития логического стиля мышления у детей, психологи — как инструмент для исследования влияния компьютерного обучения на развитие у детей умственных способностей. Однако некоторыми специалистами отмечается сложность в освоении этого средства дошкольниками, а также ограниченность его применения для развития у детей «программного» логического стиля мышления.

### Компьютер как средство развития

В разных странах специалистами изучалась возможность использования компьютера как нового средства для интеллектуального развития детей начиная с дошкольного возраста.

Широкое распространение в западных странах получила разработанная канадской фирмой DIL под руководством М. Л'Эсперанс Лабель обучающая система для использования в детском саду детьми от двух до семи с половиной лет, а также для детей с различными видами отклонений в развитии. Эта система, ориентированная на применение компьютеров типа IBM PC и Apple, методически проработана, снабжена всеми необходимыми дидактическими материалами, включает специализированные программные и аппаратные средства.

В определенном смысле любую детскую компьютерную программу можно считать развивающей, поскольку она способствует развитию восприятия, памяти, воображения и других важных психических свойств личности. Однако наибольшего эффекта в развитии умственных способностей детей дошкольного возраста можно добиться, разрабатывая специальный вид игры, учитывающие закономерности психического и психофизиологического развития детей дошкольного возраста, воз-

можности технических средств, а также законы эргономики, позволяющие с позиции системного подхода спроектировать игру в соответствии с психолого-педагогической задачей.

### Программное обеспечение для дошкольников

Какого же типа компьютерные программы используются для развития и обучения дошкольников?

В первую очередь это игровые программы дидактического («закрытого») типа, в которых в игровой форме детям предлагается решить одну или несколько дидактических задач. К этому классу относятся игры, связанные:

- с формированием у детей начальных математических представлений;
- с обучением азбуке, слоگو- и словообразованию, родному и иностранному языкам, письму через чтение и чтению через письмо;
- с формированием динамических представлений по ориентации на плоскости и в пространстве;
- с эстетическим, нравственным воспитанием;
- с экологическим воспитанием;
- с основами систематизации и классификации, синтеза и анализа понятий.

Следующий большой класс игровых по форме компьютерных программ составляет программы так называемого «открытого» типа, предназначенные для формирования и развития у детей общих умственных способностей, целеполагания, умения управлять игрой с создающимися видеоизображениями, для развития фантазии, воображения, эмоционального и нравственного воспитания. В этих играх нет явно заданной цели — они являются инструментами для творчества, для самовыражения ребенка.

К программам общеразвивающего типа относятся:

- различного рода графические редакторы, в том числе «рисовалки», «раскрашки» и конструкторы (с возможностями свободного рисования на экране прямыми и кривыми линиями, контурными и сплошными геометрическими фигурами и пятнами, закрашиванием замкнутых областей, стиранием и другими элементами коррекции рисунка);
- простые текстовые редакторы для ввода, редактирования, хранения и печати текста;
- конструкторы сред с разнообразными функциональными возможностями, в том числе те, которые служат основой

создания «режиссерских» компьютерных игр;

- музыкальные редакторы для ввода, хранения и воспроизведения простых (чаще одноголосых) мелодий в нотной форме записи;
- конструкторы сказок, совмещающие возможности элементарных текстового и графического редакторов для формирования и воспроизведения иллюстрированных текстов древовидной или сетевой структуры.

Такие игры предполагают множество педагогических методик их использования.

Еще один класс — игры-экспериментирования. В играх этого типа цель и/или правила не заданы явно, они скрыты в сюжете самой игры или в способе управления ею. Поэтому ребенок, прежде чем добиться успеха в решении такой задачи, должен путем поисковых действий прийти к осознанию цели и способа действия, что и является ключом к достижению общего решения игровой задачи. Игры такого типа можно по праву считать также диагностическими, поскольку опытный педагог и тем более психолог по способу решения компьютерных задач, по стилю и способу игровых действий сможет многое сказать о ребенке.

Однако более строго компьютерными психодиагностическими методиками можно считать лишь такие, которые в компьютерном варианте фиксируют заданные параметры, обрабатываемые и (необязательно) интерпретируемые непосредственно на компьютере. Результаты диагностики выводятся на экран дисплея или на печатающее устройство в виде рекомендаций для персонала детского сада и родителей.

К этому классу программ относятся также компьютерные методики экспресс-диагностики различных функциональных систем детского организма, позволяющие в считанные минуты выявлять патологию, отклонение от нормы с целью направления таких детей на дальнейшее обследование или лечение в специализированные медицинские учреждения.

Компьютерные диагностические программы могут использоваться в детском саду для:

- выявления уровня общих умственных способностей детей;
- оценки уровня развития психических и психофизиологических свойств личности (памяти, внимания, восприятия, умственной работоспособности, интеллекта, эмоционального состояния, нервно-психического статуса, моторики, быстроты движения и т. д.);

- выявления творческих способностей детей;
- определения уровня готовности детей к поступлению в детский сад;
- определения уровня психофизиологической и социальной готовности к поступлению детей в школу (физического развития, заболеваемости, физической подготовленности, основных физиометрических параметров растущего организма, факторов риска);
- экспресс-диагностики утомления ребенка в процессе компьютерных занятий;
- ранней диагностики отклонения детей от нормального развития.

Соответственно, можно (и это уже делается) создавать компьютерные программы и программно-технические средства для коррекции, реабилитации детей с отклонениями либо ограничениями умственного и физического развития, с нарушениями развития речи, в том числе для слабослышащих, слабослышащих детей и аутичных детей и т. д.

Компьютерные игры в сочетании с различного рода электронными и электромеханическими игрушками, связанными с компьютером через блок сопряжения ввода-вывода (коммуникационный порт), способствуют развитию у детей мелкой мускулатуры пальцев, различных групп мышц тела, рук и ног, глазоувеличительной координации, вестибулярного аппарата и т. п.

Компьютерные программы используются также для поддержки различных видов деятельности малышей. Например, много существует программ по развитию у детей основ музыкальной грамоты, ритмики и т. д. Компьютеры в этом случае снабжены дополнительно акустическими колонками, усилителем, наушникам, которые соединяются с ним с помощью специальных устройств (типа MIDI-system) либо непосредственно. Могут использоваться также электронные синтезаторы и другие инструменты.

Компьютерные игры для изобразительной деятельности используют печатающее устройство для вывода рисунков, иллюстраций к стихам, сказкам, а также простых подписей к ним и текстов.

В дошкольных учреждениях, как правило, компьютер используется не только детьми. Заведующая детским садом, методист, медсестра, бухгалтер быстро осваивают специальные компьютерные программы для управления детским учреждением. В частности, созданный в Ассоциации «Компьютер и детство» комплекс программ «КИД/администратор» эффективно решает следующие задачи:

- учет сведений о персонале, детях и родителях;
- составление меню блюд с учетом калорийности, стоимости, поставки и расходов продуктов;
- расчет зарплаты, оплаты за детей;
- учет материальных ценностей;
- учет и профилактика заболеваний, контроль за сроками прививок;
- составление планов мероприятий в группах и детском саду;
- составление справок, писем, отчетов.

Появление коммуникационной техники в детских учреждениях (пока это дело все-таки ближайшего будущего) позволит решать и такие задачи, как организация быстрого обмена информацией (в том числе деловой корреспонденцией, отчетными документами, компьютерными программами, результатами детского творчества, данными психологического тестирования и медицинского анализа и т. п.) с другими дошкольными детскими учреждениями, поликлиниками, органами управления образованием, компьютерными исследовательскими центрами и научными учреждениями, местной администрацией и т. д.; подготовка совместных коллективных детских игр, конкурсов, фестивалей, электронных выставок детского творчества, в том числе международных; получение психологических, педагогических, технических, информационных и других рекомендаций и консультаций от специалистов, находящихся в любом месте земли.

### Опыт и текущее состояние проблемы в России

В дошкольных учреждениях нашей страны компьютерная техника появилась с 1986 г., благодаря совместным усилиям специалистов ЦНИИ «Электроника» и НИИ дошкольного воспитания АПН СССР, к которым впоследствии присоединились ведущие ученые и специалисты МГУ, ВНИИ технической эстетики, НИИ общей и педагогической психологии, НИИ возрастной физиологии, НПО «Микроэлектроника» и других научных и промышленных организаций. В 1990 г. эти группы вошли в Ассоциацию «Компьютер и детство» (КИД), которая организационно оформилась как независимая фирма, консолидирующая усилия для комплексного решения проблем, связанных с использованием НИТ в дошкольном детстве.

Главная цель Ассоциации КИД — создание научно обоснованных средств новых информационных технологий и развивающего предметного мира для детей младше-

го возраста, а также методов их органичного использования в современных детских учреждениях

Ассоциация КИД организует, финансирует и координирует научные исследования и разработки, проводит обучение и переподготовку персонала детских учреждений, обучение студентов педвузов и педучилищ по новой специальности — «НИТ в дошкольном воспитании», а также осуществляет внедрение НИТ (в том числе путем поставки под ключ «технологии «КИД/система») в практику дошкольного воспитания.

За период 1986—1992 гг. на территории государств бывшего СССР подобными компьютерными системами было оснащено около 1000 детских садов, реализовано более 450 тыс. копий программ для детей дошкольного и младшего школьного возраста.

Ассоциация КИД — участник нескольких крупных разработок, в том числе совместного с фирмой IBM проекта «Пилотный детский сад», а также проектов: «Внедрения НИТ в образование, социальную сферу и семью муниципального округа «Раменки»», «Детский сад будущего» (совместно с ВНИИ технической эстетики). С 1994 г. Ассоциация КИД является одним из участников «Программы информатизации образования Российской Федерации».

Успешной реализации этих и других проектов могут и должны способствовать средства новых информационных технологий при условии их органичного введения как нового вида деятельности в практику дошкольного воспитания.

Специалистами Ассоциации КИД разработаны десятки оригинальных развивающих, обучающих и диагностических программ и методик, образцы детской компьютерной мебели, игрового и спортивного оборудования, другие изделия для детей.

Ученые и специалисты Ассоциации КИД в области психологии, эргономики, педагогики, медицины и информатики разрабатывают развивающие и обучающие компьютерные игры по следующим направлениям:

- «Первое знакомство с компьютером» — формирует начальные навыки управления компьютером, дает представление о взаимосвязи действий ребенка с происходящим на экране;
- «Почему, отчего?» — способствует развитию у детей базовых представлений о классификации и систематизации объектов по одному и нескольким признакам;
- «Мир природы» — дает начальные представления о географии, экологических процессах, знакомит с картой и

условными обозначениями, обогащает знания детей об окружающем мире;

- «Режиссерские игры» — развивает фантазию, мышление, память, дает возможность свободного творчества, способствует развитию самостоятельной и осознанной творческой деятельности;
- «Ориентация в пространстве» — формирует и развивает способность мысленной ориентации объектов на плоскости и в пространстве;
- «Конструирование» — способствует развитию фантазии, целеполаганию, творческой активности, умению самостоятельно достигать цели, обогащает арсенал средств для конструкторской деятельности детей. Хорошо сочетается с многими другими видами детских занятий. Характеризуется высокой степенью вариативности методических приемов;
- «Комбинаторика» — развивает логическое мышление, формирует понимание взаимосвязи причин и следствий процессов и явлений;
- «Живая математика» — знакомит с началами математики, обучает прямому и обратному счету, составу числа, знакомит с понятиями «множество», «число», «форма», «размер» и др. Дает возможность в игровой форме познакомиться с операциями сложения и вычитания;
- «Русский язык в картинках» — учит правильно читать, дает представление о значении слов, способствует развитию речи;
- «Водитель и пешеход» — развивает способность заранее планировать свои действия, попутно знакомит с дорожными знаками и правилами уличного движения;
- «Закономерности движения» развивает фундаментальные способности детей к выделению (установлению) новых связей и зависимостей окружающего их динамичного мира.

### Влияние компьютерных технологий на детей

Проблема влияния компьютерных технологий на детей очень обширна, но некоторые аспекты вызывают наибольший интерес у специалистов всех стран. К важнейшим из них можно отнести: влияние компьютера на здоровье детей; психическое и психофизиологическое развитие ребенка в условиях использования компьютерных методик обучения, а также социализация детей и роль компьютера в фор-

мировании новых типов взаимодействия со сверстниками и взрослыми.

Естественно, эти аспекты тесно взаимосвязаны, но междисциплинарных исследований, учитывающих эту взаимосвязь, кроме проводившихся Ассоциацией КИД, не так много.

Первостепенное внимание уделяется специалистами вопросу воздействия компьютеров на состояние функциональных систем (сердечно-сосудистой, нервной, зрительной, двигательной и др.) у детей. Многочисленные исследования, проводившиеся среди школьников, позволили сформулировать гигиенические нормы и временные режимы занятий для учебной деятельности.

С 1988 г. ведутся планомерные исследования физиолого-гигиенического характера ведутся и с детьми дошкольного возраста с учетом особенностей игрового характера их деятельности, особенностей используемых технических и программных средств. На их основе вырабатываются временные нормативы ежедневной и недельной занятости детей различного возраста за компьютером, комплекс мер для снятия нагрузки, рекомендации для педагогов и родителей по применению компьютеров без ущерба для здоровья детей.

Разработки в этой области, в частности совместные исследования медиков-гигиенистов и Института возрастной физиологии РАО под руководством д-ра мед. наук чл.кор. РАО Л. А. Леоновой, группы психологов Московского государственного университета под руководством д-ра психол. наук О. К. Тихомирова и НИИ дошкольного воспитания под руководством канд. психол. наук С. Л. Новоселовой, выявили, что при использовании специальных программных средств, разработанных в Ассоциации КИД, в условиях грамотно спланированного педагогического процесса негативного влияния на здоровья и психику ребенка не происходит.

Оценивалось также влияние компьютерных игр на эмоциональный компонент функционального состояния ребенка. Теоретические и экспериментальные исследования, проведенные группой ученых ВНИИ технической эстетики под руководством д-ра психол. наук Л. Д. Чайновой, были направлены на разработку системы показателей, характеризующих оптимальное эмоциональное состояние ребенка при игровой деятельности за компьютером. На основе кропотливой работы, в которой принимали участие психологи, психофизиологи, эргономисты, педагоги и специалисты в области вычислительной техники, были выработаны рекомендации по дальнейшему со-



вершенствованию компьютерных программ и методик их использования с учетом особенностей эмоциональной сферы ребенка дошкольного возраста.

### Явление «информатизации» дошкольного образования: «pro» et «contra»

В многочисленных работах, посвященных данной проблематике, нашли отражение различные аспекты применения компьютеров дошкольниками и младшими школьниками. При этом исследователи отмечают как позитивные, так и негативные стороны этого явления, что, пожалуй, характерно для периода поиска наиболее рационального использования столь нового по своим свойствам и стремительно развивающегося орудия человеческой деятельности, каким безусловно является компьютер.

Приведем наиболее интересные, на наш взгляд, высказывания.

«Благодаря применению компьютеров дети овладевают умением оперировать в умственном плане пространственными представлениями, решать задачи своей конструктивной деятельности» (Л. А. Парамонова, Россия).

«Компьютер обеспечивает возможности для развития наглядно-образного мышления, моторных и вербально-коммуникативных навыков, целеустремленности и социализации» (Т. Йорк, США).

«Использование компьютера для формирования у детей понятий в области элементарной арифметики позволяет перейти от запоминания к постижению глубинных процессов» (С. Линдстрем, Швеция).

«Применение ЛОГО способствует развитию когнитивных способностей ребенка: сериации, оценки значимых элементов задачи, выбора мысленного представления задачи и нахождения решения» (Т. Клементс, США).

«Компьютер может помочь детям в познавательных процессах опираться на зрительный план и одновременно способствовать активному включению речи. Это особенно важно при обучении детей с задержкой речевого развития, а также глухих, аутичных и некоторых умственно отсталых детей» (Дж. Нокс, США).

«Впервые в истории мы являемся свидетелями подлинной революции в обучении — замены учения (запоминания, приобретения навыков, достижения автоматизма в поведении, условных рефлексов и т. д.) умственным развитием (стимуляцией конструктивно-диалектического и гипотетико-де-

дуктивного мышления и игровой деятельности)» (Оливейра Лима, Бразилия).

Как отмечает Л. Цангис (США) с коллегами, анализ работы с 200 детьми в рамках проекта фирмы IBM «Head Start/IBM Partnership» позволил обозначить три основных преимущества, которые дает новая технология. Во-первых, это «равный старт», т. е. равные возможности овладения современными средствами при подготовке к школе детей из семей с разным уровнем жизни. Во-вторых, влияние нового вида деятельности на развитие самооценки и уверенности в своих силах как детей, так и их родителей, а также совершенствование коллективных форм деятельности и взаимодействия между детьми. В-третьих, хорошие программы «открытого» типа способствуют ориентировочно-поисковому характеру обучения, побуждают к обсуждению возможностей выбора и принятия решений, развивают речевую способность у детей. Кроме того, в игре с такого рода программами нет понятия «плохо» или «хорошо», а следовательно отсутствует боязнь ошибиться. При выборе из различных возможностей у детей развиваются творческие способности.

Дж. Киллиан с коллегами (США) представили результаты эксперимента, проведенного для выяснения, как в детском саду дети воспринимают появление новой игрушки — персонального компьютера и что хорошего или плохого может принести внедрение ПК в практику работы детских садов. Главный вывод, который делают авторы: ПК могут принести пользу, если будут со знанием дела внедрены в практику детских садов.

Результаты исследований Р. Хесс и Л. МакГарвей показали, что группы компьютерного обучения достигли по сравнению с контрольной («безмашиной») группой больших успехов. Наиболее заметна эта тенденция там, где занятия проводились дома.

Аналогичные результаты, были получены в ходе диагностического обследования, проведенного в одном из районов Москвы. Из 8 детских садов наилучшие результаты готовности детей к школе (до 98%) показали три, где были установлены компьютеры и в течение ряда лет с детьми проводилась планомерная работа специалистами Ассоциации КИД.

Следует отметить, что существуют разные точки зрения на возможности использования ИИТ детьми дошкольного возраста. Некоторые специалисты не только относятся к внедрению компьютерных технологий в образование с осторожностью, но и прямо предупреждают об негативных воздействиях на детей новых средств.

Так, например, в Германии, несмотря на многочисленные исследования немецких специалистов в средних и младших классах школы, показавших позитивное влияние компьютера на эмоциональную и когнитивную сферу детской психики, общее отношение к компьютеризации «скорее осторожное и даже скептическое».

Упомянутые в зарубежной и отечественной литературе негативные влияния компьютеров на социальное и когнитивное развитие дошкольников следует рассматривать в двух аспектах: влияние психолого-педагогическое и влияние физиолого-гигиеническое. Первый аспект относится скорее к программным средствам и педагогическим методам их применения в дидактике дошкольного учреждения. Второй — к используемым техническим средствам компьютера (системный блок, дисплей, клавиатура и другие устройства).

Как отмечают американские специалисты Вильямс и Маклин, одна из принципиальных проблем, касающихся первого аспекта, формулируется так: «Являются ли компьютеры средством, пригодным для детей дошкольного возраста и оказывающим существенное влияние на их развитие?»

Некоторые исследователи считают, что использование компьютеров дошкольниками и учащимися младших классов приводит к дезориентации ребенка в окружающей действительности, срыву процессов ассимиляции в становлении его мышления, к путанице в интеллектуальном развитии детей. Компьютер, утверждают они, требует формального логического мышления, тогда как маленькие дети полагаются на чувственное мышление. При работе с компьютером у детей происходит разрыв между реальностью и фантазией, желаемым и действительным, дети по своему развитию не готовы к извлечению пользы из непосредственного опыта работы с символами и абстрактной информацией, при этом необходимые для программирования навыки не соответствуют дооперационным способностям ребенка. Есть также мнение, что компьютеры могут служить препятствием для получения необходимого опыта оперирования детьми реальными объектами и событиями и, кроме того, могут ограничить физическую активность и физическое развитие детей.

Очевидно, что указанные два аспекта невозможно рассматривать по отдельности и, кроме того, ими не исчерпывается весь спектр факторов, влияющих на ребенка при использовании компьютеров в дошкольных учреждениях. Помимо психолого-педагогического и физиолого-гигиенического факторов необходимо учитывать также особенности программно-технического и органи-

зационного обеспечения использования НИТ в дошкольных учреждениях.

Анализом перечисленных выше вопросов мы хотели показать, что существуют способы рационального применения современной технологии для образования детей дошкольного возраста, а также специалисты, создающие соответствующие средства. Кроме того, необходимо привлечь внимание всех заинтересованных людей, поскольку разработка и совершенствование этих средств требуют крупных инвестиций для продолжения комплексных междисциплинарных исследований и прикладных разработок.

В результате анализа многочисленных зарубежных источников и отечественного опыта внедрения компьютерных систем в дошкольное образование можно сделать следующие выводы:

1. В мире наблюдается тенденция к все более широкому использованию компьютерных технологий для обучения, развития и развлечения детей дошкольного и младшего школьного возраста. Этому способствует снижение стоимости и увеличение функциональных возможностей техники, а также повышение качества программного обеспечения, следовательно возрастает доступность их для учреждений образования и семьи.

2. Хотя в литературе преобладают позитивные оценки влияния компьютеров на социальное и психическое развитие детей, нельзя отрицать и возможных негативных последствий. Поэтому большое значение имеет разработка корректных педагогических технологий, органично включающих НИТ.

3. Особое внимание необходимо уделять проектированию интерфейсов, облегчающих взаимодействие ребенка с компьютером, в то же время заботясь о развивающем эффекте содержания компьютерных игр.

4. Несомненно, на первое место выходят проблемы создания интегрированной развивающей предметной среды, где компьютер и другие средства НИТ наиболее естественным образом соединяются с другими дидактическими средствами и методами.

5. Вопросы использования НИТ в дошкольном воспитании как первой ступени в восходящей системе образования человека, которому предстоит продолжать образование, жить и работать в трудно прогнозируемом обществе будущего, требуют дополнительных исследований, в том числе фундаментальных на стыке философии, социологии, эргономики, информатики, психологии, педагогики и других наук.

С. В. Вершинин

## КОМПЬЮТЕР ДЛЯ ДОШКОЛЬНИКОВ

В ИНФО—4/93 А. Г. Юдина восторженно рассказывает об использовании языка LOGO в VIII—IX классах. Вполне разделяя этот энтузиазм, попробуем рассмотреть вопрос о нижней границе применимости LOGO.

Основное требование к развивающей среде — это возможность начать с простейших действий и постепенно усложнять их, причем в темпе, определяемом самим ребенком. Именно на создание таких условий и ориентирована среда LOGO.

Однако для доступа к среде кроме способности воспринимать содержание требуется еще и способность справиться с языком представления этого содержания. В данном случае возникает проблема грамотности, да еще и англоязычной. Применительно к дошкольникам она представляется едва ли преодолимым препятствием. В то же время ребенок легко связывает нажатие определенных клавиш с определенными действиями. Не случайно первые программы, обычно предлагаемые детям, — это программы работы с курсором и простейшие графические редакторы.

Исходя из этого, у нас в лаборатории реализован вариант LOGO-среды без словесных команд. Для перемещения и поворота «черепашки» используются четыре клавиши управления курсором, для других команд — дополнительная клавиатура, «ВВОД» и «ПРОБЕЛ». При запуске программы шаг «черепашки» устанавливается равным 10 точкам. Угол поворота — 30 градусов. (Эти значения также меняются соответствующими клавишами.)

Получилось весьма удачное сочетание графического редактора и расширяемой среды программирования, позволяющее справиться с еще одной проблемой работы с детьми младшего возраста — невозможностью постановки учебной задачи. Эта проблема связана с тем, что до некоторого возраста ребенок делает только то, что сам себе определил, и никаких объяснений не воспринимает. Взрослому в этом случае остается только показывать, как сделать то, что хочет ребенок, и начальное обучение сводится к демонстрации рисования, что, кстати, совпадает с концепциями LOGO.

Как показала практика, в три года ребенок хорошо осваивается с основными действиями работы с «черепашкой»: перемещением, рисованием, изменением цвета, ширины линии и др. И с увлечением занимается таким рисованием. Кстати «черепашка» в виде классического вытянутого треугольника воспринимается лучше, чем стилизованное изображение черепахи.

Через некоторое время можно попробовать давать задания рисовать что-то конкретное: квадрат, домик, круг, спираль и др.

Когда непосредственное управление ос-

воено, начинается программирование... Выясняется, что «черепашка» может запоминать последовательность действий, а также многократно повторять действия. Последовательности ставится в соответствие клавиша, при нажатии которой «черепашка» будет ее выполнять. Запомненную последовательность можно использовать и как подпрограмму в другой последовательности.

С обучением «черепашки» начинается плодотворный этап, позволяющий образно вводить многие понятия программирования и алгоритмизации. Практически с первой попытки научить «черепашку» рисовать домик ребенок с удивлением обнаруживает, что вместо домика получается несколько несвязанных линий. При анализе выясняется, что в программе карандаш был включен, но не выключен, и «черепашка» поняла очередное нажатие «ПРОБЕЛА» как выключение карандаша... Еще более эффективные результаты получаются, если использовалось изменение шага или угла поворота. Разбор причин позволяет выйти на разделение программы и данных. Как выясняется, для «черепашки» данными являются: ее положение на экране, направление, шаг, угол поворота, цвет и толщина карандаша, а также включен или не включен карандаш.

При выполнении одной и той же последовательности действий с различными данными можно получать различные рисунки, изменять их размеры, искажать форму и т. д.

Построенные программы сохраняются в виде библиотеки для последующего использования. Можно, например, обучить «черепашку» писать буквы (и присвоить рисунки букв соответствующим клавишам), и таким образом учиться грамотности через программирование...

Диапазон сложности возможных заданий весьма широкий. И хотя набор реализованных базовых команд небольшой, он вполне соответствует заявленной цели, так как моменту, когда ребенок исчерпает имеющиеся возможности, он, по крайней мере, овладевает грамотностью. Продолжение обучения в традиционном LOGO является вполне естественным.

Сколько времени продолжаются занятия? Я уже сказал, что ребенку нравится работать с «черепашкой», и он может заниматься ею по полчаса и более. Однако, если эти полчаса случаются чаще, чем раз в неделю, у 3—4-летнего ребенка интерес снижается быстрее, чем осваиваются новые возможности. Так что, видимо, этого времени вполне достаточно. В дальнейшем интенсивность занятий можно увеличивать.

И. В. Левченко

## ИГРАЕМ В «ТЕЛЕФОН»

Идея создания обучающей программы «Телефон» возникла в результате анализа компьютерных занятий, проводимых с детьми 4—7 лет в центре «Малышок» при педагогическом колледже № 13 г. Москвы. Опыт использования обучающих программ «Фима-строитель», «Фима-художник», «Буква потерялась», «Обучение грамоте», «Юниор» и других показал, что наиболее сложными являются первые занятия, требующие от учителя и учащихся большой мобилизации, так как у детей еще не сформированы навыки работы на клавиатуре компьютера, что снижает эффективность занятий. Известно, что прежде, чем учить ребенка написанию букв, необходимо отработать элементы письма, укрепить кисть руки. Так и при обучении компьютерной грамотности на первом этапе целесообразно использовать специальную программу, которая позволяет целенаправленно формировать умение работать на клавиатуре.

Обучающая программа «Телефон» позволяет отрабатывать правильное нажатие на клавиши компьютера, научить использовать клавиши управления курсором, клавишу ввода информации, цифровые клавиши клавиатуры, работать с манипулятором типа «мышь», а также развивать рефлексию своей деятельности, внутренний план действий, способствует формированию произвольного внимания, развитию познавательных способностей, памяти, воображения.

Практические занятия показали, что программа «Телефон» вызывает у детей познавательный интерес и положительное эмоциональное отношение к вычислительной технике, подготавливает их к дальнейшему систематизированному обучению. Творчески работающий воспитатель или учитель начальных (I—II) классов может использовать данную программу при формировании компьютерной грамотности и элементарных математических представлений у детей, для развития речи, при ознакомлении с окружающей средой. Программа позволяет научить ребенка обобщать, систематизировать и практически применять полученные знания, сформированные умения и навыки.

Программа «Телефон» предназначена для IBM совместимых компьютеров. При

запуске программы на экране монитора появляется зеленая лужайка, освещенная переливающимися лучами солнышка, и дом, в котором обитают животные. Дом трехэтажный, на каждом этаже по три окна. Таким образом, всего изображено девять окон, и у каждого есть свой номер, который соответствует цифрам на клавишах дополнительной клавиатуры. При нажатии на клавишу в соответствующем окне появляется определенная зверюшка. При повторном нажатии на данную клавишу зверюшка исчезает. Чтобы начать работать с программой, детям не обязательно знать все цифры от 1 до 9. Достаточно узнавать начертание единицы и иметь элементарные пространственные представления (справа, слева, выше, ниже), так как расположение окон в доме на экране монитора соответствует расположению клавиш дополнительной клавиатуры. Клавиатуру рекомендуется поставить в наклонном положении, чтобы одни клавиши были несколько выше других.

Вывести изображение зверюшки также можно, используя клавиши управления курсором (клавиши со стрелками). На экране монитора курсор изображен в виде стрелки (указателя). Перемещая курсор нажатием на клавиши, подводим его к требуемому окну и нажатием на клавишу ввода (Enter) выводим изображение на экран. Расстояние, на которое перемещается курсор при однократном нажатии на клавишу, выбрано с учетом того, чтобы перемещение не было затянато и у детей не возникло бы желание прижать клавишу, вместо кратковременного нажатия на нее. Отработка умения прижатия клавиши является задачей следующего этапа.

Программа управляется и с помощью манипулятора типа «мышь». Перемещая «мышь» по столу, указатель подводят к соответствующему окну, и при нажатии на левую клавишу «мыши» в данном окне появляется зверюшка. При повторном нажатии на клавишу зверюшка исчезает. Для лучшего сцепления «мыши» с поверхностью, по которой передвигают данное устройство, целесообразно использовать резиновые коврики.

Обучение необходимо начинать с использования клавиш дополнительной клавиатуры, чтобы зафиксировать взаимо-

связь между нажатием на клавишу и появлением животного на экране монитора. Вывод изображения с помощью управляющих клавиш клавиатуры и устройства «мышь» требует от детей усвоения алгоритма вывода на экран монитора изображения животного (вначале переместить курсор в требуемое место на экране, а затем уже нажать клавишу для вывода изображения). Дети должны усвоить, что управлять программой возможно тремя способами и не следует элементы одного способа заменять элементами другого. Например, если курсор перемещается с помощью «мыши», то вывод изображения осуществляется нажатием на клавишу «мыши», а не на клавишу «ввода» на клавиатуре. Это позволит в дальнейшем избежать неточностей в работе с программами. Комбинация же данных способов и выбор оптимального из них при решении конкретной задачи желательны.

В зависимости от возраста детей данную обучающую программу можно использовать на пяти и более занятиях или уроках. Перед преподавателем стоят следующие задачи:

- научить правильно нажимать на клавиши (нажатие кратковременное, мягкое и не сильное);
- научить работать с манипулятором типа «мышь»;
- освоить знаковое пространство дополнительной клавиатуры;
- научить точно соотносить собственные воздействия на отдельные элементы клавиатуры с результатами, появляющимися на экране монитора;
- научить соотносить движение курсора на экране монитора с направлениями, указанными стрелками на клавишах управления курсором ;
- уточнить пространственные представления учащихся (выше, ниже, справа, слева, следовать за);
- научить сравнивать группы предметов (больше, меньше);
- развивать произвольное внимание и образную память ребенка;
- формировать способности самостоятельно управлять игрой (программой), развивать речь.

Во время проведения занятий целесообразны такие методические приемы, как рассказ, объяснение, демонстрация, экспериментирование, беседа, создание проблемной и игровой ситуации.

Хотелось бы остановиться на некоторых методических рекомендациях.

На первом занятии целесообразно работать с цифровыми клавишами дополнительной клавиатуры для установления зависимости порядкового номера окна с соответствующей клавишей.

Вначале учитель объясняет детям, что в домике живут различные животные и для того, чтобы увидеть их, надо позвонить им по телефону. Телефон кнопочный (клавиши дополнительной клавиатуры), и при нажатии на клавишу с цифрой в окне, обозначенном той же цифрой, появится зверушка.

После объяснения организуется синхронная работа учащихся под руководством учителя: исследование, какая зверушка в каком окне появляется. Учитель просит нажать на клавишу с цифрой один. В окне с этим номером появляется слон. Просит еще раз нажать на эту клавишу, и слон исчезает. На данном этапе необходимо добиваться от детей правильного нажатия на клавиши (кратковременного и не сильного), стремиться к тому, чтобы учащиеся установили взаимосвязь расположения клавиш дополнительной клавиатуры и окошек дома. Целесообразно обсудить выполняемые учащимися действия.

— Почему слон появляется в окне домика? (Мы ему позвонили.)

— Каким образом мы ему позвонили? (Нажали на клавишу.)

— Клавишу с какой цифрой мы нажали, чтобы в окне появился слон? (Один.)

— Почему мы знаем, что надо нажать на клавишу с цифрой один? (Потому что в окне, к которому подходит слон, стоит цифра один.)

Таким же образом исследовать, какие звери подойдут к другим окнам.

После вывода на экран всех зверей можно задать вопросы, кто живет выше, ниже, между, справа, слева определенного зверя. В завершении данного этапа работы целесообразно предложить детям отозвать от окна того зверя, который живет, например, ниже белки, слева от свиньи, между слоном и зайцами, справа от льва, выше зайцев и т.д.

Во время занятий необходимо организовывать физкультминутки для снятия физического и эмоционального напряжения. Целесообразно, чтобы физкультминутки имели смысловое значение. Учитель для этого может ввести понятие «команда» и давать точные указания учащимся для исполнения. На следующих занятиях данная игровая ситуация может получить дальней-

шее развитие в играх «Робот» и «Оператор».

При завершении работы необходимо вспомнить, с какой программой работали учащиеся и как они это делали. Домашним заданием может быть рисунок с изображением любого животного, живущего в домике.

Обучающую программу «Телефон» можно использовать для закрепления навыков количественного и порядкового счета предметов, осознания взаимосвязи количественных и порядковых чисел, для закрепления знания цифр, формирования умения сравнивать группы предметов.

На втором занятии нужно вспомнить с детьми, каким образом они работали с программой, уточнить с ними значение клавиш.

— С какой программой мы работали в прошлый раз? («Телефон».)

— Почему зверюшка знает, что ей надо подойти к окну? (Даю ей команду.)

— Что такое команда? (Точное указание.)

— Каким образом вы даете команду компьютеру? (Нажимаю на клавишу.)

— На клавишу с какой цифрой вы нажали, чтобы показалось животное на первом этаже в левом окне? (Один.)

— Почему? (Потому что в этом окне стоит цифра один.)

Следует дать возможность самому ребенку поэкспериментировать, оказывая при необходимости помощь.

В дальнейшем учитель вводит понятие «курсор», объясняет и демонстрирует, какими клавишами он передвигается. Организовывает синхронную работу учащихся. При этом можно обсудить с детьми, где живут данные звери, чем питаются, как они выглядят и какие у них повадки.

Индивидуальные задания могут быть следующие: позвонить и тем самым позвать к окнам животных, живущих в жарких странах, в наших лесах, домашних животных, диких животных; птиц, зверей.

Домашнее задание может сводиться к составлению диалога между любыми зверятами.

Третье занятие целесообразно посвятить повторению основных понятий и закреплению умения работать с клавишами клавиатуры. Уделить внимание развитию образной памяти. Можно попросить детей вспомнить, какая зверюшка к какому окну подходит, когда мы даем соответствующую команду, а затем проверить это, нажав на необходимую клавишу.

Проверяя домашнее задание, нужно стимулировать речевую активность учащихся.

На следующем этапе занятия учащимся предлагаются загадки. Дети при отгадывании отдают команду для того, чтобы отозвать от окна героя данной загадки. Для данного этапа надо подготовить карточки с изображением животных. В заключение подвести итоги и попросить учащихся дома вспомнить другие загадки о животных.

На четвертом занятии учитель объясняет и демонстрирует работу с манипулятором типа «мышь». Учащиеся синхронно выполняют задания по передвижению курсора по экрану, по установлению курсора на определенное окно, по выводу изображения животного посредством нажатия на левую клавишу «мыши». Следует обратить внимание на правильное расположение кисти руки на данном устройстве и на размещение указательного пальца на левой клавише «мыши».

На следующем этапе урока учитель читает книгу Корнея Чуковского «Телефон», а учащиеся нажатием на клавиши подзывают к окну то животное, с которым разговаривает автор. По окончании диалога учащиеся дают команду данному животному, чтобы оно отошло от окна. В конце можно задать вопрос: «Какое животное звонило не из своей квартиры?» (Слон, так как он звонил от верблюда.)

Дома можно попросить детей сочинить сказку, в которой героями являются животные, живущие в домике.

Пятое занятие необходимо посвятить обобщению накопленного опыта работы с программой «Телефон» и творческой работе учащихся по созданию компьютерных сказок.

Данные методические рекомендации будут полезны воспитателю, преподавателю в организации занятия с детьми с использованием готовых компьютерных программ, в частности обучающей программы «Телефон». Но при проведении практических занятий целесообразно не забывать о следующих направлениях: использовании и разработке алгоритмов, занятиях с готовыми компьютерными программами, фундаментальных понятиях о компьютерах, областях применения компьютеров, роли компьютера в обществе, программировании, технике безопасности. Успех в овладении компьютерной грамотностью зависит от работы по каждому направлению. Это необходимо учитывать и при занятиях с обучающей программой «Телефон».

# НОВЫЕ ПРОЕКТЫ

От редакции

Арсенал понятий, связанных с компьютерным представлением информации, напоминает роскошную палитру Рубенса. Чего там только нет — электронные таблицы, гипертекст, интерактивная графика, когнитивная графика, видео, анимация, мультимедиа, искусственная реальность и, наконец, гипермедиа (объединение идей гипертекста и мультимедиа). Какими же средствами пользуются программисты при создании этих и других компьютерных чудес? Увы, сапожник, как всегда, без сапог. Сегодня, как и сто пятьдесят лет назад, во времена Чарлза Бэббеджа и Августи Ады Лавлейс, программисты вынуждены использовать «язык знаков», современные версии которого превратились в хорошо известные языки программирования: БЕЙСИК, ПАСКАЛЬ, СИ, ПРОЛОГ и т. д. Все эти языки имеют «текстовый» характер и позволяют записать программу в виде невзрачной и тусклой последовательности операторов (какой уж там Рубенс!), т. е. длинной и унылой цепочки символов.

Правда, в последнее время лед тронулся. С появлением визуального стиля программирования возникла возможность не писать программы, а именно рисовать. Тем самым программист-«писатель» превратился в программиста-«художника». В 1988 г. в США вышла книга Н. К. Шу под названием «Визуальное программирование». Термин прижился и стал общепринятым. Одним из пионеров нового направления был киевский ученый И. В. Вельбицкий, еще двадцать лет назад выступивший с идеей Р-технологии, которая явилась одной из первых ласточек визуализации в мире компьютерных программ. С тех пор было придумано немало новых визуальных языков: SDL, FPL, VIPS, STILE, HI-VISUAL и т. д. (см. обзор в журн: Программирование. 1990. № 4, С. 68—79).

Однако скоро сказка сказывается, да не скоро дело делается. Большинство специалистов до сих пор воспринимает языки визуального программирования как некую причуду, далекую от повседневных нужд. Этому в какой-то степени содействует ситуация, сложившаяся в системе образования, поскольку существующие учебники по курсу информатики основаны на принципах текстового программирования и не знакомят учащихся с идеями визуализации.

В помещенной ниже статье предлагается осуществить поэтапную «визуализацию» школьного курса информатики с помощью визуального языка ДРАКОН. При этом используется оригинальный метод формализации блок-схем. Если обычная блок-схема всего лишь «картинка», непригодная для ввода в компьютер и к тому же зачастую неряшливо оформленная, то ДРАКОН-схема — эстетически безупречная «выполняемая графика», т. е. компьютерная программа в строгом смысле слова. Разумеется, это далеко не первая попытка формализации блок-схем — достаточно вспомнить язык SDL, проект Cados, пакет Lab View для Windows и т. д. Однако ДРАКОН выгодно отличается более высокими когнитивно-эргономическими и дидактическими свойствами, что особенно важно в условиях школы.

Редакция приглашает читателей высказаться по существу поднятых в статье вопросов. Одобряете ли вы идею «визуализации» школьного курса информатики? Если да, является ли ДРАКОН удачным выбором? Или же предпочтение следует отдать какому-либо другому визуальному языку?

**В. Д. Паронджанов,**

*ст. науч. сотр., канд. техн. наук, НПО автоматики и приборостроения, руководитель Общественной лаборатории «Человеческий интеллект и информационные технологии XXI века»*

## КАКИМ БУДЕТ ШКОЛЬНЫЙ АЛГОРИТМИЧЕСКИЙ ЯЗЫК XXI ВЕКА?

**Какой ресурс самый ценный?**

Давайте помечтаем и немножко фантазируем. Предположим, наши нищие школы по мановению волшебной палочки превратились в «интеллектуальные здания» [1], битком набитые последними чу-

десами электроники, а экраны ПЭВМ увеличились до размеров письменного стола. Словом, ни аппаратура, ни программное обеспечение нас больше не лимитируют — в запасе есть все, что душа пожелает, и даже больше. А теперь вопрос: как лучше преподавать информатику в роскошно

оборудованных школах будущего? Какой язык программирования победит во всемирном конкурсе на звание «лучший школьный язык XXI века»? Неужели это будет БЕЙСИК? Или все-таки ПРОЛОГ? А может, какой-нибудь отечественный продукт, например КуМир или язык «К» [2, 3]? Или, наоборот, победа достанется экзотическим языкам вроде FPL и STL [4]?

Ответить, конечно, трудно. Или даже невозможно. Поэтому спустимся на ступеньку ниже и зададим вопрос попроще: каким требованиям должен удовлетворять идеальный школьный язык — язык нашей завтрашней мечты? Рискуя ошибиться, выскажу некоторые предположения. Наверное, идеальный язык должен создать наиболее комфортные условия для интеллектуальных занятий школьников. Разумеется, он должен быть общедоступным, человеческим, предельно легким в изучении и удобным в работе. Однако этого мало! Самое главное, он должен позволять и всемерно помогать учащимся приобретать знания и решать сложные и сверхсложные задачи по возможности без большого напряжения, **ЦЕНОЮ МИНИМАЛЬНЫХ ИНТЕЛЛЕКТУАЛЬНЫХ УСИЛИЙ**. Мне кажется, в школе будущего мозг ребенка будет рассматриваться как наиболее ценный ресурс общества, к которому следует применять идеи ресурсосберегающего подхода. Если это верно, перед педагогикой встает двоякая задача: во-первых, вооружить учащихся необходимым запасом знаний и умений, во-вторых, сделать это за минимальное время **ПРИ НАИМЕНЬШИХ ЗАТРАТАХ ЭНЕРГИИ НЕРВНОЙ СИСТЕМЫ РЕБЕНКА**, т. е. при минимальном «расходе» наиболее ценного ресурса.

### Так кто же все-таки победит?

С другой стороны, «общеизвестно, что человеческий мозг в основном ориентирован на визуальное восприятие и люди получают информацию при рассмотрении графических образов быстрее, чем при чтении текста» [5]. Эта простая мысль вступает в острое противоречие с существующим положением дел, так как подавляющее большинство языков программирования, используемых сегодня в мировой школьной практике, — именно текстовые языки. Исходя из этого, можно предположить, что в данной области нас ожидают революционные потрясения и школьный язык будущего, стремящийся наилучшим образом

учесть когнитивно-эргономические особенности человеческого восприятия и нейробиологические процессы, протекающие в мозгу ребенка, окажется уже не текстовым, а графическим (или, как говорят специалисты, языком визуального программирования [4, с. 73]). Хотелось бы верить, что самые сложные программы, написанные на таком языке, будут необыкновенно наглядными и (во всех случаях, когда это возможно) понятными буквально с первого взгляда. Если эту чрезвычайно трудную проблему удастся решить, школа тем самым сделает всего лишь маленький шаг навстречу заветной мечте многих поколений школьников о легкой и безбедной учебе, лишенной ненужных страданий («Мамочка, я никогда не смогу этого понять!»), а также непомерной нагрузки, накапливающегося переутомления и калечащих здоровье стрессов и нервных срывов. Встав на такую позицию — позицию гуманного и бережного отношения к ребенку, мы естественным образом приходим к требованию: изучаемые в школе алгоритмы и программы следует подбирать таким образом, чтобы, обеспечивая быстрый рост знаний и умений школьника, они вместе с тем в максимально возможной степени приближались к идеалу, который выражается поговоркой: «Взглянул — и сразу стало ясно!» Назовем это требование критерием сверхвысокой понимаемости программ или, что одно и то же, критерием сверхнаглядности алгоритмов.

Отсюда проистекает довольно неожиданный вывод: в школе будущего, где забота об интеллектуальном комфорте и, следовательно, о здоровье учащихся будет иметь первостепенное значение, понятность алгоритмов и программ, вероятно, станет одним из важнейших или даже наиболее важным критерием при оценке качества конкурирующих школьных языков. Словом, при прочих равных условиях победит тот язык, который позволит писать наиболее понятные алгоритмы и программы.

### Прыжок ДРАКОНа: из космоса в школьный коридор

Если предположить, что перечисленные требования не являются надуманным и дешевым заигрыванием с детьми, а действительно учитывают потребности учебной практики и верно отражают пока еще, быть может, неочевидные и подспудные, но



в перспективе весьма важные и ведущие к успеху тенденции развития языковых средств школьной информатики, то, по-видимому, одним из вероятных претендентов на звание «самый легкий и наглядный школьный язык» может стать язык ДРАКОН (Дружелюбный Русский Алгоритмический язык, Который Обеспечивает Наглядность). ДРАКОН разрабатывается в НПО автоматики и приборостроения (Москва) совместно с Институтом прикладной математики РАН, являясь обобщением опыта, накопленного при проектировании орбитального космического корабля «Буран» [6].

Предвижу недоумение читателя. Мол, какая связь между «Бураном» и школьной информатикой? Не получается ли «в огороде бузина, а в Киеве дядька»?

Нет, не получается. И вот почему. Как известно, «в современных условиях качественная программа должна обладать помимо надежности и эффективности еще и такими важнейшими свойствами, как понимаемость и сопровождаемость» [7]. Обратите внимание — понимаемость сегодня рассматривается как ВАЖНЕЙШЕЕ свойство программы. Почти все согласны, что «от 50 до 90% времени, расходуемого на сопровождение программ, тратится на их понимание» [8], поэтому многие специалисты считают, что понимаемость алгоритмов и программ очень актуальная и «болезненная» проблема [9]. Данное понятие (обычно выражаемое такими словами, как понимаемость, познаваемость, понятность, comprehensibility) определяется как «свойство программы минимизировать интеллектуальные усилия, необходимые для ее понимания» [7].

Проблема понимаемости алгоритмов сегодня становится критической не только для школьной информатики, но и при разработке больших проектов, таких, как «Буран». Дело в том, что при создании сложнейшего комплекса бортовых и наземных программ «Бурана» приходится расплетать хитроумный клубок донельзя запутанных проблем. Поэтому в бой вступает целая армия специалистов разных профессий из множества разных организаций, каждый из которых поначалу знает лишь свой участок работы и смутно представляет, что творится у соседей. Это именно тот случай, когда запредельная сложность проблемы и связанная с нею узкая специализация приводят к смешному, но, увы, реальному парадоксу, когда «специалисты по клизме» не понимают «специалистов по наконечнику».

Тем не менее создаваемые ими алгоритмы, больше напоминающие первозданный алгоритмический хаос, раздираемый молниями вопиющих неувязок, в конечном счете должны превратиться в единый филигранный узор, управляющий «Бураном» с баснословной точностью и надежностью.

Таким образом, гвоздь проблемы стар как мир: чтобы избежать печальной участи строителей вавилонской башни, участники сложного проекта должны научиться ОЧЕНЬ ХОРОШО ПОНИМАТЬ ДРУГ ДРУГА. В противном случае многочисленные ошибки «на стыках» помешают успеху разработки. По этой причине в НПО автоматики и приборостроения (которое является разработчиком системы управления «Бурана» [6] и главным дирижером сводного оркестра космических алгоритмов и наземных программ) на протяжении многих лет ведутся фундаментальные исследования, направленные на решение проблемы понимаемости алгоритмов и программ, результаты которых нашли свое воплощение в концепции языка ДРАКОН. При создании языка была поставлена задача: за счет использования когнитивно-эргономического подхода к проектированию синтаксиса и семантики добиться кардинального улучшения качества программного обеспечения по критерию «познаваемость программ».

Таким образом, сверхвысокая понимаемость алгоритмов и программ — это и есть тот золотой мостик, по которому ДРАКОН может «перебежать» из исследовательских лабораторий ракетчиков в классные комнаты массовой школы.

### В чем изюминка ДРАКОНа?

ДРАКОН родился не на пустом месте — он получен путем формализации и неклассической структуризации хорошо известных блок-схем алгоритмов и программ [10]. Поскольку ДРАКОН — язык визуального программирования, он имеет не один, а два набора синтаксических правил. Текстовый синтаксис фиксирует правила записи и соединения текстоэлементов, а графический синтаксис — правила изображения и соединения графоэлементов. Таким образом, первая «изюминка» состоит в том, что ДРАКОН адресуется не только к словесно-логическому мышлению школьника, но, сверх того, активизирует интуитивное, образное, правополушарное мышление, стимулируя его не написан-

ной, а именно нарисованной программой, т. е. программой-картинкой\*.

Далее. Чтобы облегчить труд школьника по принципу «от простого к сложному», вводятся две разновидности языка: ДРАКОН-1 (который попроще) и ДРАКОН-2 (этот посложнее). Оба языка имеют одинаковый графический синтаксис, что зрительно делает языки почти близнецами, обеспечивая преемственность обучения, и отличаются только текстовым синтаксисом. В ДРАКОН-1 используется текстовый синтаксис самого легкого языка — родного (имеется в виду естественный язык с национальной лексикой). Поэтому, строго говоря, ДРАКОН-1 не язык программирования, а всего лишь частично формализованный набор правил (графический псевдокод), позволяющий рисовать наглядные картинки, которые используются преимущественно во вводной части учебного курса. Что касается ДРАКОНа-2, это настоящий язык программирования, обеспечивающий автоматическую интерпретацию программ, а также трансляцию и исполнение объектных модулей. ДРАКОН-2 изучается в основной части курса.

*Вторая изюминка.* Расщепление единого набора синтаксических правил на две части (текстовый и графический синтаксис) позволяет упростить изучение языка по методу «разделяй и властвуй». Уже во вводной части курса, разглядывая и придумывая увлекательные и забавные картинки, нарисованные на языке ДРАКОН-1, учащиеся почти полностью осваивают графический синтаксис и семантику, что составляет примерно 50% от общего объема синтаксических и семантических правил. Тем самым готовится мощный трамплин для перехода к основной части учебного курса.

*Третья изюминка* заключается в том, что графический синтаксис ДРАКОНа является «самообъясняющим»: во многих случаях, хотя и не всегда, он интуитивно самоочевиден и почти не требует пояснений, в чем легко убедиться, взглянув на рис. 1—10.

*Четвертая изюминка* — это особый ме-

тодический прием, облегчающий изучение языка ДРАКОН-1, который можно назвать так: «не мучай ребенка, начни с привычного». На первом этапе, пока школьник еще не освоил язык, не стоит показывать ему незнакомые, мудреные или абстрактные алгоритмы, так как подобное забегание вперед неоправданно увеличивает нагрузку на мозг. Следует использовать тот факт, что в памяти любого ребенка хранится огромный запас удивительно сложных, разнообразных и многофакторных «бытовых» алгоритмов, в которых воплощен его повседневный, литературный, игровой и сказочный опыт. Рассматривая такие родные и привычные в бытовом отношении алгоритмы, как «яйцо всмятку», «строгая мама», «поездка на автобусе», «рыбная ловля» и т. д. (см. рис. 1—8), школьник испытывает приятное чувство узнавания своих собственных мыслей и знаний, что, несомненно, облегчает процесс изучения графического синтаксиса и семантики. Детская память бытовых алгоритмов — бесконечно богатая и почти не освоенная кладовая, которую просто необходимо активно использовать в целях обучения школьников алгоритмическому мышлению. Однако для этого нужно иметь удобный инструмент, специально приспособленный для решения данной задачи. Язык ДРАКОН располагает именно таким инструментом. Он напоминает своеобразное увеличительное стекло, которое позволяет ребенку более глубоко понять и познать себя, заглянув в свой многокрасочный алгоритмический внутренний мир, спрятанный в тайниках собственного подсознания. Выигрыш в том, что изучение наиболее сложных операторов: если-то-иначе, переключатель, цикл, вызов подпрограммы и их комбинаций превращается из мучительного труда в довольно простое дело, а иногда — в интересную игру.

*Пятая изюминка* называется «борьба со словами-паразитами». Чтобы выявить суть вопроса, обратимся к таблице, в которую сведены некоторые ключевые слова, выписанные из школьных учебников информатики [12—15] и соответствующие им операторы БЕЙСИКА [16].

\* В литературе последнего времени подчеркивается, что привычное для нашей речи выражение «написать программу» является отражением традиционного, уже, пожалуй, старомодного или даже устаревшего стиля, в рамках которого при слове «программа» человек «представляет себе ее текст или код, поскольку привык к тому, что программы существуют в виде некоторой последовательности операторов». Однако сегодня положение изменилось. Как отмечают сторонники новых идей, при использовании языков визуального программирования «программы не пишут. Их рисуют» [11].

Таблица

Учебник [12]	Учебник [13]	Учебник [14]	Учебник [15]	Бейсик
алгоритм	алг	алг	алг	program, proc, procedure
—	нач	нач	нач	—
если	если	если	если	if
то	то	то	то	then
иначе	иначе	иначе	иначе	else
—	—	инес	иначе если	elseif
конец ветвления	все	кесли	все	end if, bend
повторять	нц	цикл	нц	—
пока ... повторять	нц пока	пока ... цикл	пока ... нц	while
для	нц для	—	для ... нц	for
конец цикла	кц	кцикл	кц	wend, next
—	выбор	—	выбор	select case
—	при	—	при	case
выполнить алгоритм	вызов	—	—	gosub, call
—	—	—	—	goto

Таблица примечательна тем, что позволяет выявить ахиллесову пятю традиционной школьной информатики, дидактическая слабость которой во многом связана с некритическим восприятием и канонизацией устаревших постулатов текстового программирования. Очевидный недостаток состоит в том, что, во-первых, алгоритмическая семантика названных ключевых слов неоправданно трудна для усвоения, во-вторых, зубрежка неблагозвучных слов-обрубков (алг, кесли, нц, кц) вряд ли может вдохновить школьника, в-третьих, разногласия учебников приводит к тому, что ученики разных «научных школ» с трудом понимают друг друга.

Вспомним историю. Когда программирование в машинных кодах истощило свой эвристический потенциал, оно было отброшено и заменено — сперва автокодами и ассемблерами, затем языками высокого уровня. Сегодня история повторяется. Господствующая ныне парадигма текстового программирования во многих, хотя и не во всех, случаях исчерпала себя. Поэтому представленные в таблице ключевые слова (которые, образно говоря, представляют собой ее одряхлевший «машинный код») зачастую можно безболезненно отбросить

и заменить более легкими для усвоения графическими фигурами, например графо-элементами ДРАКОНа. Правда, при этом потребуются ввести два новых ключевых слова — «да» и «нет», однако их семантика на два порядка проще и хорошо известна даже детям дошкольного возраста.

### Зачем ДРАКОНу ветка?

Недостаток классического структурного программирования состоит в том, что оно предлагает либо очень мелкие структурные элементы (следование, развилка, цикл), либо очень крупные (процедура, функция) и не содержит промежуточных элементов. Чтобы поправить дело, в язык ДРАКОН вводится принципиально новый структурный элемент под названием «ветка», который не имеет аналогов в классическом структурном программировании. ДРАКОН-схема без веток называется «примитив» (см. рис. 1 — 4, 6), с ветками — «силуэт» (см. рис. 5, 7 — 10). При входе в программу-«силуэт» управление всегда передается на крайнюю левую ветку.

С точки зрения ученика, любой незнакомый нетривиальный алгоритм или программа — чрезвычайно сложная проблема,



Рис. 1 Алгоритм:  
как сварить яйцо всмятку



Рис. 2. Алгоритм:  
как сварить яйцо вкрутую



Рис. 3. Алгоритм «Строгая мама»

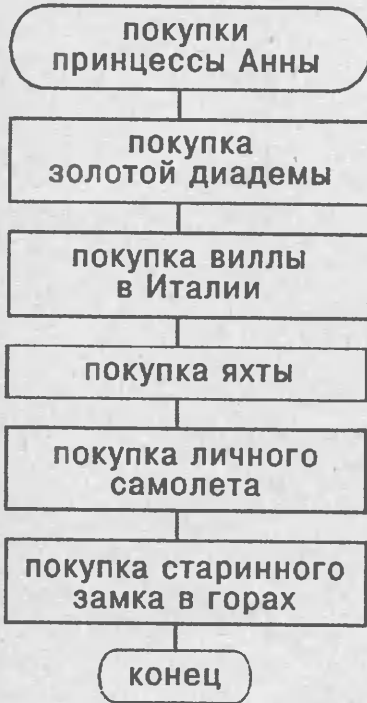


Рис. 4. Дракон-схема, которая называется «Примитив»

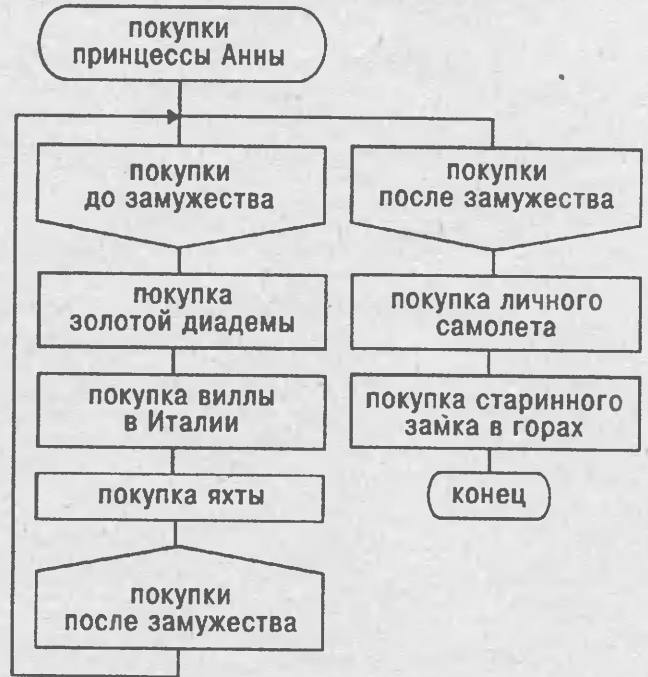


Рис. 5. Дракон-схема, которая называется «Силуэт»

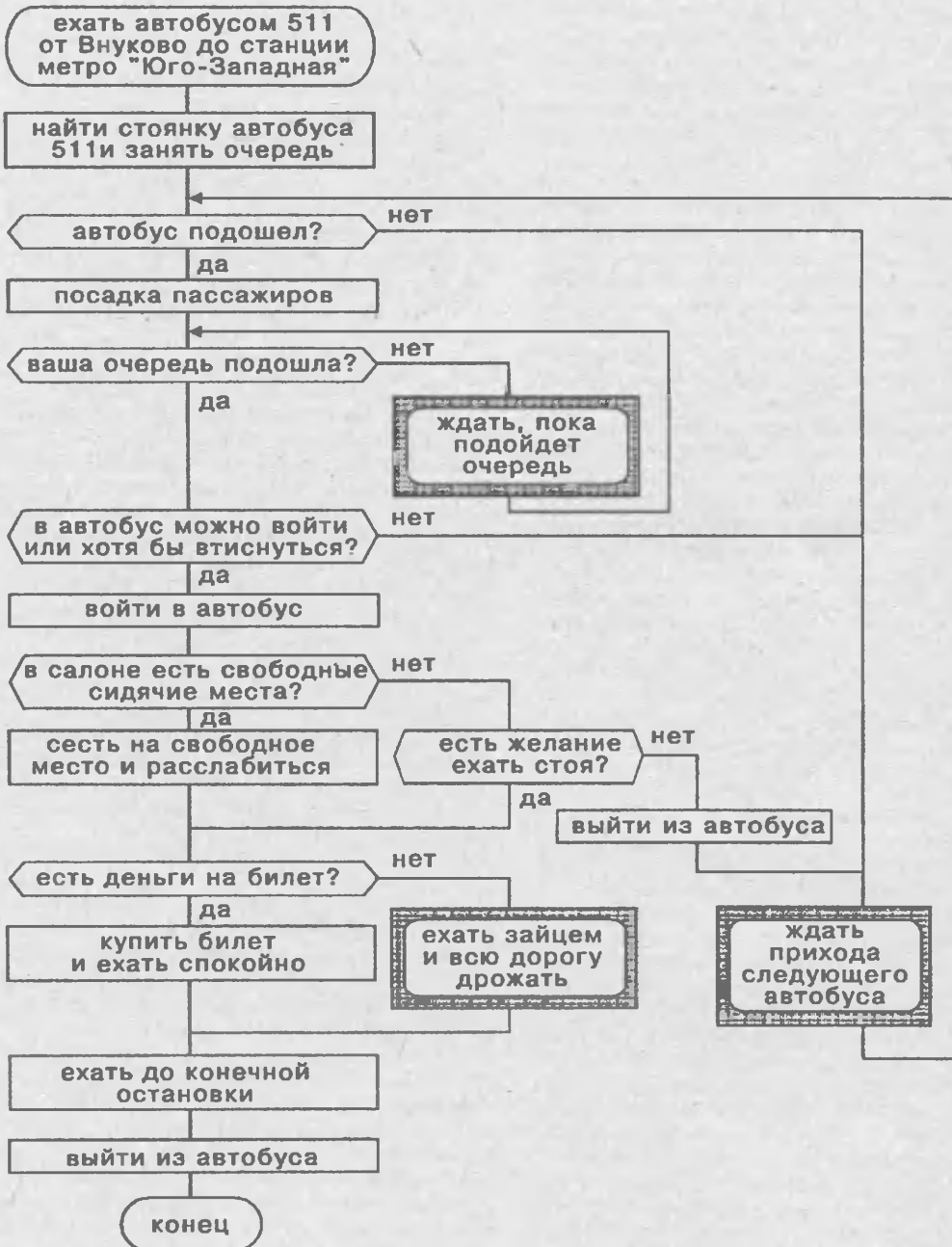


Рис. 6. Алгоритм поездки на автобусе («Примитив»)

которую он отчаянно пытается понять, преодолевая мощное «сопротивление материала». Чтобы упростить дело и облегчить задачу понимания, нужно, чтобы школьник «прозрел» и, расчленив проблему на части, увидел ее смысловую структуру. Причем увидел не в фигуральном смысле слова, не с помощью воображения, не духовным оком, а своими двумя глазами — на бумаге или экране.

Но как это сделать? Трудность в том, что ни один из существующих языков не предоставляет читателю, изучающему сложные алгоритмы и программы, эффективной помощи, позволяющей МОМЕНТАЛЬНО уяснить их визуально-смысловой каркас. Чтобы решить задачу, ДРАКОН-схема «силуэт» расчленяется на несколько веток, которые зрительно полностью изолированы друг от друга. Например, алгоритм на рис. 5 содержит две ветки: «покупки до замужества» и «покупки после замужества». На рис. 7 четыре ветки: «поиск автобуса», «ожидание посадки», «посадка в автобус», «поездка в автобусе». Деление алгоритма на ветки как раз и образует искомый визуально-смысловой каркас.

Каждая ветка имеет один вход и один или несколько выходов. Она состоит из трех частей: графоэлемента «начало ветки», в котором записано ее имя, тела ветки и одного или нескольких графоэлементов «адрес», в которых указывается имя следующей ветки, на которую следует передать управление. Например, вторая ветка на рис. 8 состоит из заголовка «ожидание клева», тела ветки, которое включает блоки «забросить удочку», «клянула?», «подсекай!», «пора домой?», «ждать, пока не клянет» и двух элементов «адрес»: «рыбацкая работа» и «обратная дорога». При положительном ответе на вопрос «клянула?» после исполнения блока «подсекай!» управление передается на графоэлемент «адрес», внутри которого указано имя «рыбацкая работа». Элемент «адрес» выполняет безусловный переход на следующую ветку, именно на ту, у которой в графоэлементе «начало ветки» записано имя, совпадающее с указанным в блоке «адрес». Вход в любую ветку возможен только через ее начало.

Шапкой называется верхняя часть ДРАКОН-схемы «силуэт», которая включает заголовок алгоритма, верхнюю шину и графоэлементы «начало ветки». Назначение шапки — помочь читателю мгновенно (не более, чем за несколько секунд) получить ответ на три вопроса: 1) как называется проблема, 2) из скольких частей она состоит, 3) как называется каждая часть? Вот ответы для рис. 8.

— Как называется проблема? Рыбная ловля.

— Из скольких частей состоит проблема? Из четырех.

— Как называется каждая часть?  
1. Подготовка к ловле. 2. Ожидание клева. 3. Рыбацкая работа. 4. Обратная дорога.

Таким образом, ДРАКОН предоставляет школьнику эффективный трехэтапный метод познания незнакомой программы. На первом этапе, анализируя шапку, школьник познает зрительно-смысловую структуру программы, ее деление на ветки. На втором этапе осуществляется углубленный анализ каждой ветки. На третьем — производится разбор взаимодействия веток.

### Что лучше: примитив или силуэт?

Если программа очень простая (примитивная) и содержит не более 5—15 графоэлементов, ДРАКОН-схему рекомендуется оформлять как примитив, в противном случае — как силуэт.

Нарушение этого правила обычно чревато неприятностями, ибо мешает школьнику выявить сущность решаемой проблемы и, следовательно, затрудняет и замедляет понимание смысла программы.

Например, алгоритм на рис. 6 выглядит громоздким и неоправданно сложным для восприятия; это вызвано тем, что он содержит 19 блоков, однако изображен в виде примитива. Криминал в том, что схема на рис. 6 не позволяет читателю моментально (за несколько секунд) распознать визуально-смысловую структуру алгоритма. В самом деле, из скольких частей состоит решаемая проблема? Глядя на рис. 6, ответить на этот вопрос довольно трудно, а быстро ответить — невозможно. Положение в корне меняется, когда мы смотрим на рис. 7, где тот же самый алгоритм изображен в виде силуэта. Тут, как говорится, и ежу ясно: алгоритм состоит из четырех частей: «поиск автобуса», «ожидание посадки», «посадка в автобус», «поездка на автобусе». Однако это не все: не менее важно и то обстоятельство, что запутанность зрительного рисунка исчезла и схема приобрела новой эстетическое (эргономическое) качество — элегантность, ясность и прозрачность. Таким образом, в сложных случаях силуэт позволяет существенно уменьшить интеллектуальные усилия, затрачиваемые на понимание программы.

Впрочем, иногда полезно использовать силуэт и в более простых ситуациях (рис. 5). Если по каким-то причинам, например при споре о разделе имущества после развода, покупки принцессы Анны нужно раз-



делить на сделанные до и после замужества, выгоднее использовать силуэт (рис. 5). Если же подробное разграничение не играет роли, следует применить примитив (рис. 4).

### Зачем нужен главный маршрут?

Некоторые инженеры НПО автоматизики и приборостроения, склонные к резким выражениям, называют традиционные блок-схемы программ [10] «помоночными блок-схемами», потому что изображенные на них хитросплетения блоков, соединенных хаосом куда угодно гуляющих рваных линий, больше напоминают кучу мусора, нежели регулярную структуру. ДРАКОН выгодно отличается тем, что его графический узор имеет строгое математическое и когнитивно-эргономическое обоснование и подчиняется жестким и тщательно продуманным правилам. Среди них особое место занимает правило под названием «главный маршрут», которое позволяет сделать ДРАКОН-схему обзоримой и зрительно предсказуемой.

Рассмотрим задачу. Предположим, в запутанном лабиринте, соединяющем начало и конец сложной программы, требуется по некоторому критерию выделить один-единственный маршрут — «путеводную нить», с которой можно зрительно сравнивать все прочие маршруты, чтобы легко сориентироваться в алгоритме и не заблудиться в путанице развилки. Эта «путеводная нить» (назовем ее «главный маршрут») должна быть визуально легко различимой. Иными словами, бросив беглый взгляд на ДРАКОН-схему, мы должны обнаружить четкие ориентиры, благодаря которым можно сразу и безошибочно увидеть главный маршрут и упорядоченные относительно него остальные маршруты.

Понятие «главный маршрут» имеет два аспекта: зрительный и логический. С визуальной точки зрения, главный маршрут примитива должен быть вертикальной линией, идущей сверху вниз без каких-либо изломов и уступов (см. рис. 1—4, 6).

Аналогично главный маршрут ветки есть вертикаль, идущая от начала ветки до графоэлемента «адрес», а если у ветки несколько выходов — до левого из них. Главный маршрут силуэта — последовательное соединение главных маршрутов поочередно работающих веток.

Логический аспект заключается в следующем. Меняя места слова «да» и «нет» в развилках и варианты в переключателях (и подключенные к ним гирлянды блоков), следует добиться, чтобы на главном маршруте оказался тот выход развилки или переключателя,

который ведет к наибольшему успеху. При этом каждая ветка должна быть построена по единому правилу: чем правее (чем дальше от главного маршрута) расположена очередная вертикаль, тем менее успешные действия она описывает. Например, на рис. 7 ветка «посадка в автобус» имеет три вертикали. Левая вертикаль (главный маршрут) описывает наибольший успех, так как вы будете ехать в автобусе сидя. Правая вертикаль означает наименьший успех, поскольку вы вышли из автобуса и поездка откладывается. Средняя вертикаль (расположенная выше блока «есть желание ехать стоя?») занимает промежуточное положение, потому что — в зависимости от ответа на заданный вопрос — может иметь место либо частичный успех (вы будете ехать, но не сидя, а стоя), либо неудача, поскольку вы выходите из автобуса, так куда и не уехать.

В тех случаях, когда понятие «успех» не работает, вместо него следует выбрать какой-либо другой осмысленный критерий, чтобы смещение вправо от главного маршрута ветки или примитива всегда было не случайным, а обоснованным.

Таким образом, ДРАКОН позволяет школьнику моментально увидеть главный маршрут любого, сколь угодно сложного и разветвленного алгоритма и, сверх того, делает смещение всех вертикалей относительно главного маршрута не случайным, а осмысленным и предсказуемым, т. е. легким для восприятия.

### Дополнительные пояснения к рисункам

Итак, важнейшая цель языка ДРАКОН состоит в том, чтобы увеличить понимаемость алгоритмов и программ, минимизировать интеллектуальные усилия школьника, необходимые для их восприятия, познания и составления. Для решения этой задачи и используется целый арсенал специальных средств, некоторые из которых (далеко не все) представлены на рис. 1—10.

Просматривая рисунки, следует иметь в виду, что в них порою используются нетрадиционные обозначения. Например, блок со штриховкой — это встроенный комментарий (рис. 1—3, 6—9), внутри которого разрешается помещать не только текст, но и графику, а также видео и средства управления гипертекстом и мультимедиа. Перевернутая трапеция обозначает паузу (задержку), имеющую фиксированную длительность, что существенно облегчает понимание процессов реального времени (рис. 1—3). Развилка (проверка усло-

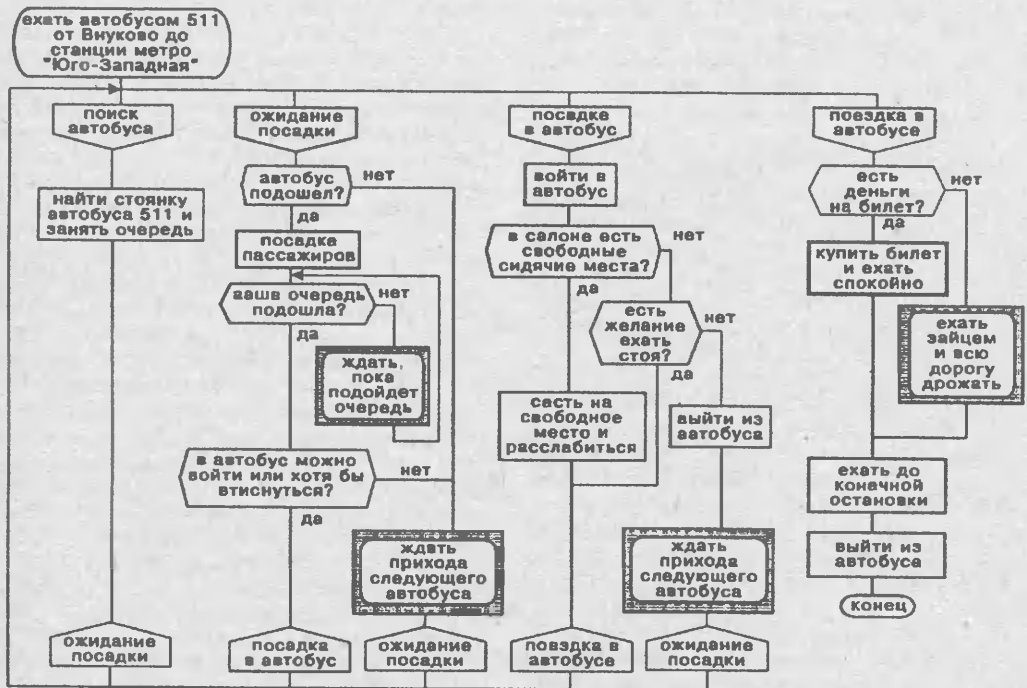


Рис. 7. Алгоритм поездки на автобусе («Силуэт»)

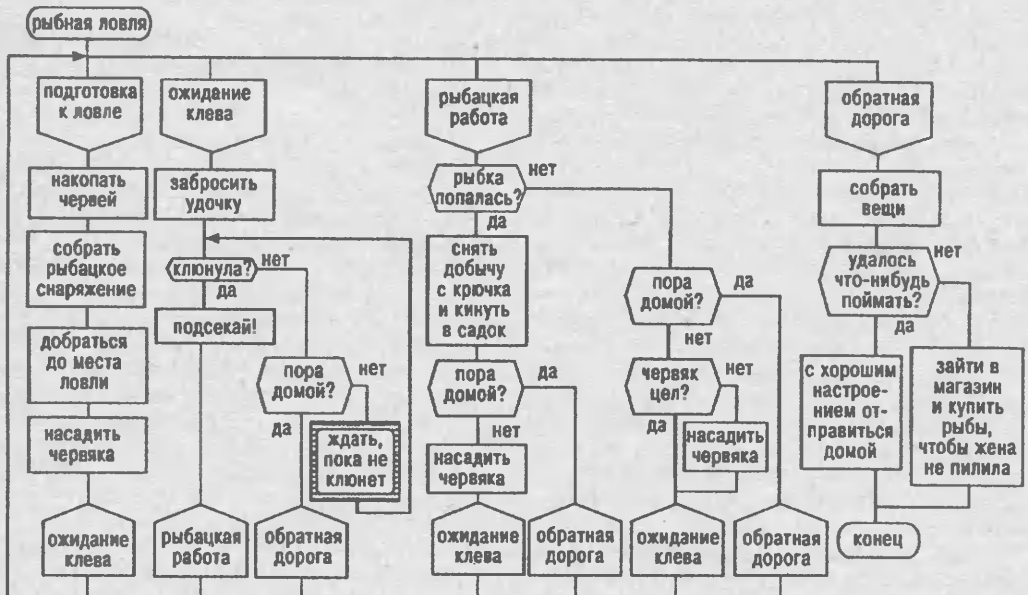


Рис. 8. Алгоритм «Рыбная ловля»

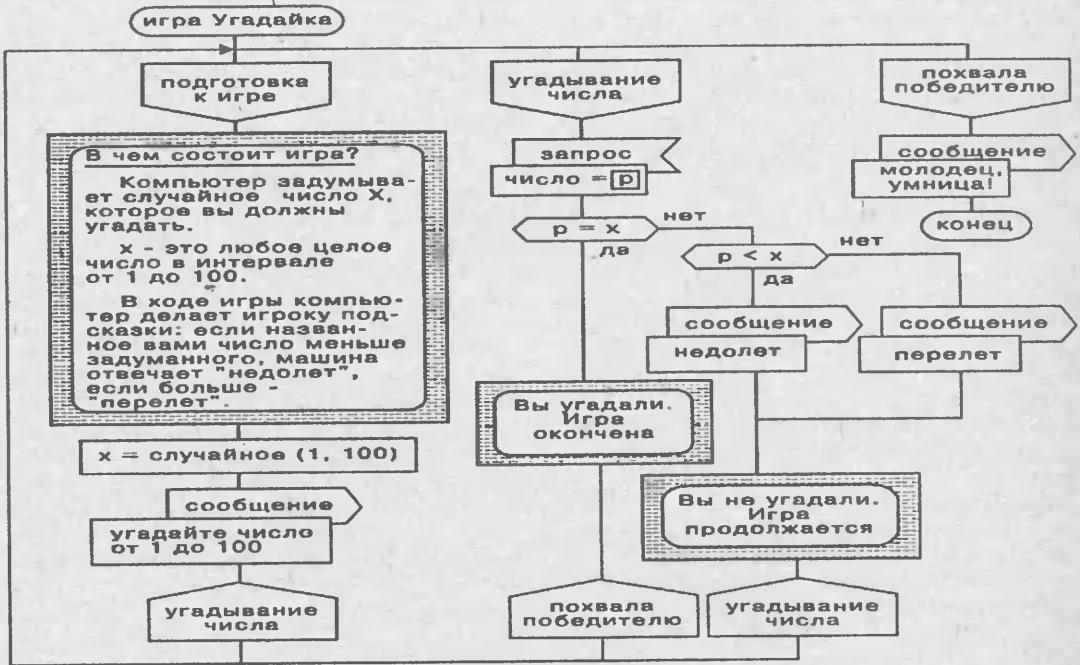


Рис. 9. Алгоритм игры «Угадайка» на языке ДРАКОН

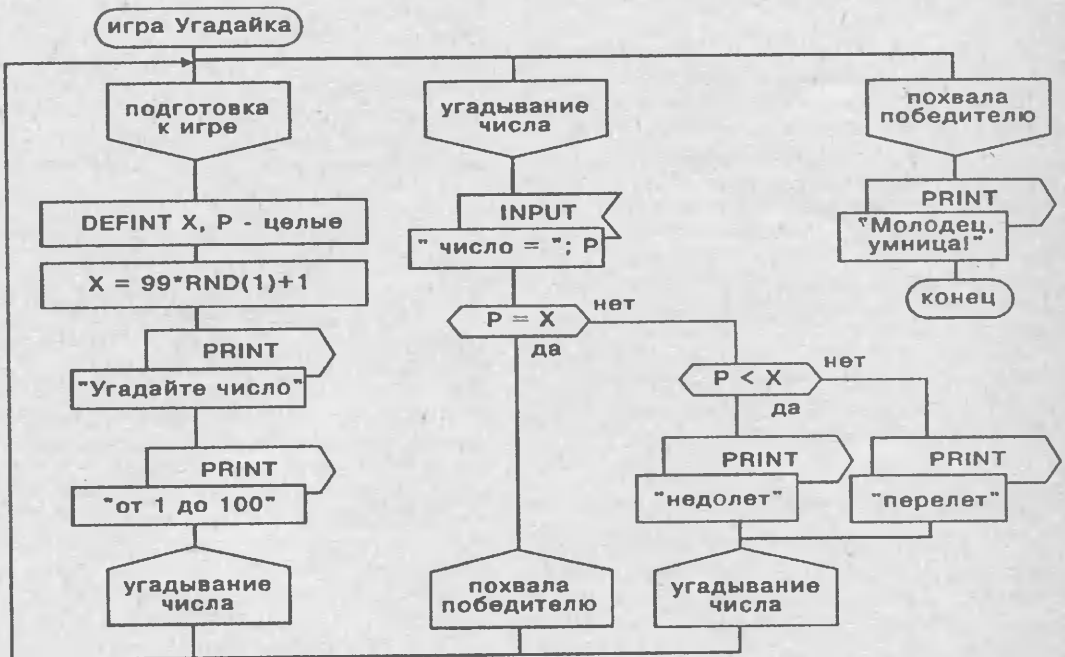


Рис. 10. Программа игры «Угадайка» на языке ДРАКОН-БЕЙСИК

вия) обозначается не ромбом, как обычно, а стилизованным ромбиком, у которого отпилены верхний и нижний углы.

Внимательно анализируя рисунки, читатель может обнаружить целый ряд секретов из серии «тысяча мелочей», направленных на облегчение труда школьника, о которых нет возможности рассказать в рамках этой, по необходимости краткой статьи. Хотя каждый из подобных «секретных приемов», рассматриваемый в отдельности, может показаться не заслуживающим внимания пустяком, однако в целом дружная армия полезных мелочей может стать важным подспорьем для тех, кто стремится создать для школьника условия максимального интеллектуального комфорта в когнитивно-эргономическом смысле слова. Приведем несколько примеров.

1. В ДРАКОН-схеме запрещаются пересечения, разрывы и неоправданные изломы линий.

2. Ромбик имеет один вход сверху и два выхода: вниз и вправо; все остальные графоэлементы имеют один вход сверху и один выход снизу.

3. Запрещается направлять управляющую связь снизу вверх во всех случаях кроме организации цикла.

4. В графоэлементе «заголовок» следует указывать точное название алгоритма; запрещается вместо названия писать слово «начало».

5. Условие задачи рекомендуется помещать во встроенный комментарий, размещенный после заголовка примитива (рис. 3) или после начала левой ветки силуэта (рис. 9).

6. Используйте прием «горизонтальная очная ставка», располагая одинаковые элементы на одном горизонтальном уровне. Применение этого приема на рис. 1 и 2 позволяет читателю легко обнаружить разницу между алгоритмами «яйцо всмятку» и «яйцо вкрутую»: в первом случае яйцо опускают в горячую, во втором — в холодную воду. Тот же прием на рис. 7 (см. два графоэлемента «ждать прихода следующего автобуса») позволяет лучше уяснить визуальную структуру алгоритма. И так далее.

### Семейство ДРАКОН-языков

Термин «ДРАКОН» обозначает не один, а целое семейство языков, внутри которого действует единый графический стандарт изображения ДРАКОН-схем. Семейство включает не только ДРАКОН-1 и ДРАКОН-2, но и ряд других, так называемых гибридных языков: ДРАКОН-БЕЙСИК, ДРАКОН-ПАСКАЛЬ, ДРАКОН-СИ и т. д. Чтобы построить гибридный язык, нужно к графиче-

скому синтаксису ДРАКОНа присоединить по определенным правилам текстовый синтаксис любого из перечисленных языков. Чтобы пояснить эту идею, одна и та же программа, заимствованная с небольшими модификациями из [14, с. 117], представлена на языке ДРАКОН-2 (рис. 9), ДРАКОН-БЕЙСИК (рис. 10) и БЕЙСИК (рис. 11). Для удобства читателя на рис. 11 показано соответствие между операторами БЕЙСИКА и ветками ДРАКОНа и ДРАКОН-БЕЙСИКА.

Во избежание недоразумений добавим, что программа на ДРАКОНе-2 (рис. 9) является неполной: в ней отсутствуют указания о типе переменных X и P. Это объясняется тем, что указанная информация может задаваться как на ДРАКОН-схеме, так и в другом месте.

И последнее. На языке ДРАКОН-2 переменная величина, значение которой человек должен ввести в ЭВМ, записывается в маленьком квадрате в элементе «запрос» (рис. 9). Такая запись намного нагляднее, чем в Бейсике (см. эквивалентный оператор INPUT на рис. 10 и 11).

### Заключение

В статье сделана попытка обсудить вопрос о перспективах развития языковых средств школьной информатики. В настоящее время они включают различные варианты так называемого школьного алгоритмического языка и какой-либо язык программирования, преимущественно БЕЙСИК, причем все используемые языки развиваются в рамках традиции текстового программирования. Главный вывод состоит в том, что критерий сверхвысокой понимаемости алгоритмов и программ в будущем должен стать определяющим при выборе генерального направления развития школьных языков. Отсюда вытекает, что языки визуального программирования могут составить мощную конкуренцию традиционным текстовым языкам и, вероятно, уже в этом десятилетии одержат победу в борьбе за лидерство. В качестве одного из возможных подходов к созданию школьного языка нового поколения предложена концепция визуального языка ДРАКОН, который является универсальным в том смысле, что способен одновременно выполнять функции как школьного алгоритмического языка, так и школьного языка программирования.

В настоящее время разработка ДРАКОНа еще не завершена, однако в НПО автоматизации и приборостроения совместно с Институтом прикладной математики РАН уже создана CASE-система «графит-прол2», являющаяся первым, но пока еще весьма гру-

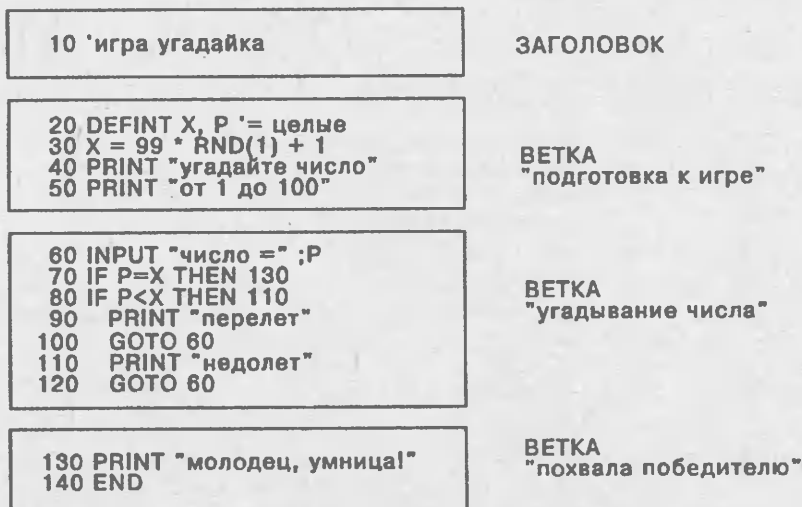


Рис. 11. Программа игры «Угадайка» на языке БЕЙСИК

бым приближением к ДРАКОН-технологии. К сожалению, система «графит-прол2» не удовлетворяет требованиям школьной информатики; поэтому сегодня ДРАКОН может использоваться в школе только как

«бумажный» школьный алгоритмический язык, не имеющий программной поддержки. Думается, однако, что даже в этом ограниченном качестве он, по-видимому, может быть полезен учителям информатики\*.

**Литература**

1. Информационные города и интеллектуальные здания. // В кн. Современные тенденции информатизации и медиатизации общества. М.: ИНИОН, 1991. С. 98—110.
2. Кушниренко А., Эпиктетов М. КуМир — новое семейство учебных программ // Информатика и образование. 1993. № 1. С. 21, 22.
3. Кузнецов В. Е. Язык программирования «К» в изучении основ информатики // Информатика и образование. 1993. № 3. С. 78—82.
4. Хлебцевич Г. Е., Цыганкова С. В. Визуальный стиль программирования: понятия и возможности // Программирование. 1990. № 4. С. 73, 74, 76.
5. Вельбицкий И. В., Ковалев А. А., Лизенко С. А. Графический интерфейс представления алгоритмов и программ // УСИМ. 1988. № 4. С. 42—47.
6. Лалыгин В. С космической орбиты до посадочной полосы безукоризненно вела «Буря» уникальная система автоматического управления // Социалистическая индустрия. 1988. 7 дек.
7. Саркисян А. А. Повышение качества программ на основе автоматизированных методов. М.: Радио и связь. 1991. С. 17, 19.

8. Robson D. J., Bennet K. H., Cornelius B. J., Mulgo M. Approaches to Program Comprehension // J. Systems Software. 1991. № 14. P. 79—84.
9. Поиск. 1993. № 18. С. 5.
10. ГОСТ 19.701—90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.
11. Сбытов Н. Н., Трубицын А. В. Окно в новое измерение // Мир ПК. 1993. № 7. С. 71—75.
12. Гейн А. Г. и др. Основы информатики и вычислительной техники. М.: Просвещение, 1994.
13. Кушниренко А. Г. и др. Основы информатики и вычислительной техники. М.: Просвещение, 1993.
14. Каймин В. А. и др. Основы информатики и вычислительной техники. М.: Просвещение, 1989.
15. Ершов А. П. и др. Основы информатики и вычислительной техники. Ч. 1, 2. М.: Просвещение, 1988.
16. Кетков Ю. А. Толковый словарь языка программирования БЕЙСИК. М.: Наука, 1992.

\* Более подробные сведения о ДРАКОНе можно найти в книге В. Д. Паронджанова «Учись рисовать ясные блок-схемы. ДРАКОН — общедоступный графический язык для систематизации знаний», которая выйдет в издательстве «Радио и связь» в 1995 г.

Вышли из печати приложения к журналу «Информатика и образование».  
Ниже публикуется их содержание.

### «ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР БК-0010 – БК-0011М» №2–1994

	Информационное сообщение
Ю. А. Зальцман	МикроЭВМ БК-0010. Архитектура и программирование на языке ассемблера
	Ремонт на дому
Д. Ю. Усенков	ПЗУ для БК
Борис Ф. Фролкин	Контроллер дискового БК-0010/11 и его доработка
А. Г. Прудковский	Операционная система NORD – плюсы и минусы
В. З. Манвелян	Универсальный программатор микросхем ППЗУ для БК
А. И. Винниченко	DOSB10 v2.0: новая версия ОС для БК
А. А. Саяпин	Драйвер виртуального диска для ОС БК-0011М
Д. Ю. Усенков	Компактное хранение БЕЙСИК-кодовых программ
	Материалы о БК, опубликованные в журналах ...потехе час

Подписной индекс: для индивидуальных подписчиков – 73177  
для предприятий и организаций – 73092

Периодичность – 1 раз в 2 месяца.

### «ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР УКНЦ» № 1–1994

В. В. Дубовик	Солнечногорский электромеханический завод
	Список сокращений
М. Б. Носов	Все о периферийном процессоре «Глюки» и как их обойти
Б. Ф. Фролкин	PAF Commander V1.02 А. Петросяна
	DESS V05.00е Д. Климова
	К вопросу о работе локальной БЕЙСИК-сети в КУВТ УКНЦ
	Справочник пользователя
	Режим адресации процессора K1801BM2
	Система команд процессора K1801BM2
	Система прерываний процессора K1801BM2
	Глобальные переменные системы ввода-вывода (ОС ПМ)
	Обзор
А. В. Тютюнников	Архиваторы
И. И. Быстров	«NET-Rt11» – сетевая система для КУВТ «УКНЦ» с IBM-совместимой головной машиной
О. И. Ховайко	Хитрости работы с периферийным процессором

Подписной индекс: для индивидуальных подписчиков – 73179  
для предприятий и организаций – 73093

Периодичность – 1 раз в 3 месяца.

На все издания можно подписаться с любого месяца в отделении связи.  
Для тех, кто не подписался, справки о приобретении по тел. 208-30-78, 151-19-40.

# ТОЧКА ЗРЕНИЯ

В. О. Байбаков

## ЭВМ В ВУЗЕ

Нет сейчас, пожалуй, устройства более модного, чем ЭВМ. Их появление принято сравнивать по значению с «приручением» огня и изобретением парового двигателя. По количеству компьютеров оценивается престижность института, а в сравнительно недавнем прошлом за процент использования ЭВМ в курсовых и дипломных проектах начислялись очки в соцсоревновании. Возьмите проспект любого вуза, и, если в нем есть хотя бы одна фотография, вы наверняка увидите на ней компьютер.

Но вот идут годы, все больше компьютеров получает институт, и после первой эйфории приходит понимание, что если студент введет в компьютер три цифры, а в ответ получит четвертую, то трудно ожидать от этого существенного повышения качества учебного процесса.

Следующий этап внедрения ЭВМ в учебный процесс связан с разработкой специальных обучающих программ. И здесь разработчик, как правило, натыкается на один из трех «подводных камней». Назовем их «синдром книги», «синдром белочки» и «синдром кино».

На начальных этапах, когда работать приходится на примитивных компьютерах типа «ДВК-2» или «Роботрон 1715», на первый план выходит «синдром книги». И бедный студент вынужден читать огромные тексты на прыгающем, дергающемся и мерцающем экране. Изредка тексты прерываются контрольными вопросами, когда нужно выбрать один ответ из нескольких (точно так же, как много лет назад на различных электромеханических игрушках).

Но вот институт получает IBM PC-совместимые компьютеры с их богатыми графическими возможностями. И сразу на поверхность всплывает «синдром белочки». Здесь, вероятно, уместно пояснить, при чем тут симпатичный грызун. Он, точнее, она пришла к нам из старого анекдота: когда воспитательница детского сада выводит детей в лес, забирается на дерево и говорит: «А вот так, дети,

выглядит белочка...» Так и студенты подчас вместо того, чтобы потрогать руками реальную шестеренку, коровье вымя или человеческий череп, вынуждены разглядывать на экране компьютера очередную электронную «белочку».

И наконец как совокупность электронной книги с электронной «белочкой», выступает электронное кино. Увы! Не так уж редко приходится наблюдать, как прекрасный сценарий учебного фильма с дикторским текстом за кадром, с изображением реальных объектов и только с элементами (подчеркну ЭЛЕМЕНТАМИ) мультипликации превращается на экране компьютера в скучную электронную книгу, по страницам которой обильно прыгают «белочки».

Так что же получается? Компьютер хуже книги — с экрана трудно читать. Компьютер хуже кино — беднее изобразительные средства. И наконец, компьютер при контроле хуже человека и не лучше примитивных контролирующих аппаратов. Вот тут вы со мной, возможно, сразу не согласитесь. «Позвольте! — воскликнете вы. — А как же развитые системы анализа ответов по ключевым словам! В Адонисе, например!» (Адонис — это довольно распространенная система для подготовки обучающих программ.) Да, действительно анализ ответа в произвольной форме по ключевым словам теоретически возможен. Но богат и могуч русский язык! Почти невозможно подобрать ВСЕ ключевые слова, которыми может быть описан правильный ответ, стопроцентно исключив при этом возможность неправильного ответа с теми же ключевыми словами. Добавьте к этому слабое владение клавиатурой и возможные вследствие этого опечатки, и я уверен, вы сами вернетесь к доброму старому методу, когда предлагается выбрать один ответ из нескольких.

Есть и еще один весьма веский довод не в пользу применения компьютеров в учебном процессе — их относительная дороговизна. В самом деле, если «у них» стоимость хорошего компьютера составляет процентов тридцать

месячного оклада преподавателя, то у нас на его приобретение пойдет месячный фонд заработной платы целой кафедры.

Итак, если компьютер хуже книги, хуже кино (видео), хуже преподавателя при контроле и вдобавок сказочно дорог, так стоит ли вообще применять его в учебном процессе? Будем честны — при нынешнем состоянии высшей школы (да и всего государства) такой необходимости практически нет.

Пройдут годы, и, даст Бог, все встанет на свои места. Квалифицированный рабочий станет получать больше грузчика, инженер — больше рабочего, а профессор — больше их всех, вместе взятых. Знания дороги, а компьютеры дешевы и будут, так же как сейчас «у них», столь же обычной вещью, как телефон.

А что пока? На нынешнем этапе очень важно понять самим и объяснить студентам, где та граница, за пределами которой применение компьютера действительно оправдано, а где можно обойтись калькулятором, карточкой, пишущей машинкой и прочими атрибутами прошлого века. Нужно нащупать те участки учебного процесса, где применение вычислительной техники дает действительно новое качество, и сосредоточить на этих участках все те крохи техники, которые все же достаются институтам.

В свете этих общих рассуждений давайте вначале остановимся на таком, самом исконно компьютеризированном предмете, как информатика (вычислительная техника и программирование и т. п.). Не секрет, что в абсолютном большинстве вузов преподавание этого предмета сводится к изучению одного из языков программирования. Чаще всего — БЕЙСИКА. Недавно от очень уважаемого человека мне пришлось услышать, что в его институте сделан большой шаг в преподавании информатики — они переходят с БЕЙСИКа на ТУРБО-ПАСКАЛЬ. В ответ я сказал, что мы подумываем, а не отказаться ли вообще от изучения языка программирования (что, кстати, давным-давно сделано во всех «цивилизованных» странах). Почему? Обратимся к истории.

Появление первых персональных компьютеров принято связывать со стремлением энтузиастов избавиться в общении с ЭВМ от посредничества «касты жрецов» — программистов. Но вот прошло полтора десятка лет — и что же? Количество профессиональных программистов возросло в несколько раз! Но свои посреднические услуги они выполняют уже не «физически», а посредством программ. В результате абсолютное большинство пользователей персональных компьютеров никогда не занимаются программированием. (В одной умной книжке мне встретился любо-

пытный афоризм. Цитирую по памяти: «Если вам в голову пришла мысль написать программу для IBM PC-совместимого компьютера, немедленно откажитесь от нее. Такая программа наверняка уже существует, и дешевле купить ее, чем разрабатывать самому.») Но если так, то стоит ли вообще изучать язык программирования в институте? Со временем, когда информатика будет достаточно хорошо преподаваться в школе, от изучения языка программирования в институте можно будет отказаться. Пока же, думается, это преждевременно. И не потому, что язык пригодится потом (наверняка не пригодится), а потому, что нет лучшего способа развить логику мышления, точность формулировок, аккуратность, чем программирование. Кстати, для изучения того же БЕЙСИКа нет необходимости в дорогих компьютерах. Пойдет и БК, и «Микроша».

Какой язык выбрать для изучения: БЕЙСИК, ПАСКАЛЬ, а может быть, СИ или АС-СЕМБЛЕР? Дело вкуса (и финансовых возможностей — СИ на «Микроше» не пойдет). Но мне кажется, что к БЕЙСИКУ не стоит относиться свысока. Во-первых, этот язык стал своеобразным «эсперанто» программистов. Еще один афоризм: «Если ваша ЭВМ не понимает язык Бейсик, — значит, это не ЭВМ» (Основы информатики и вычислительной техники. Свердловск: Изд-во Уральск. ун-та, 1989. С. 272). Во-вторых, БЕЙСИК позволяет наиболее быстро усвоить минимальный набор операторов и перейти к реальному программированию. И наконец, современные версии БЕЙСИКа не так уж сильно уступают по возможностям более сложным языкам.

Итак, курс «Программирование» пока имеет право на жизнь. Но вот следует ли ограничиваться только им? Нет! Необходим другой курс — «Применение ЭВМ в ...». В рамках этого курса необходимо давать студентам навыки работы на современных IBM PC-совместимых компьютерах (вероятнее всего, именно они в обозримом будущем останутся стандартом делового компьютера). Необходимо дать обзор программного обеспечения — представление о текстовых редакторах, электронных таблицах, базах данных, АРМах и др. Студенты должны получить представление о разновидностях ЭВМ, их марках, цене, тенденциях рынка компьютеров.

Таким образом, если курс «Программирование» направлен на формирование образа мышления будущего специалиста и по этому признаку может быть отнесен к базовым предметам, то курс «Применение ЭВМ в ...» должен быть чисто практическим, т. е. давать конкретные знания, на основании которых



будущий специалист сможет принимать решение о целесообразности использования вычислительной техники и соответствующего программного обеспечения при решении профессиональных задач.

При изучении как курса «Программирование», так и курса «Применение ЭВМ в ...» необходимо использование компьютеров. Преподавание «насухо» — при помощи доски и мела — приносит мало пользы. Так же малоэффективно оказывается проведение занятий, когда студенты сидят по два, три, а то и четыре человека за компьютером. При этом один из студентов постоянно оказывается в роли лидера, практически единолично работая на ЭВМ, в то время как остальные остаются пассивными наблюдателями и не получают необходимых навыков. Качество обучения оказывается тем выше, чем меньше студентов приходится на одного преподавателя. Правда, эта не очень оригинальная мысль может быть отнесена и к любому другому предмету. Однако есть в отношении «компьютерных курсов» и одна особенность. заключается она в том, что 90% ошибок и затруднений у начинающих пользователей компьютера — типовые, повторяющиеся от группы к группе, из года в год. Поэтому огромную помощь преподавателю может оказать работающий в паре с ним инженер, лаборант или даже студент старшего курса, прошедший предварительно минимальную подготовку. Но в любом случае на одного преподавателя должно приходиться не больше 15 студентов. Проведение занятий с целой группой, даже с помощью лаборанта, малоэффективно.

Итак, вводный курс по применению ЭВМ пройден. Как дальше должно строиться общение с компьютером?

Прежде всего необходимо отказаться от стопроцентного охвата студентов компьютеризацией (здесь речь не идет о курсах «Программирование» и «Применение ЭВМ в ...» — там стопроцентный охват необходим). Хорошо это или плохо? Конечно, плохо — современный специалист обязан прилично работать на персональном компьютере. Но при нашей нищете приходится выбирать: либо все студенты научатся «чуть-чуть», либо некоторые из них смогут овладеть компьютером профессионально. Уже на первых этапах компьютерного обучения из группы выделяются один-два студента, активно интересующихся ЭВМ, стремящихся с ними работать. Наша задача — сделать из этих студентов проводников новых компьютерных технологий, а, возможно, в будущем и программистов наиболее ценной «породы», объединяющих в одном лице и специалиста-эксперта, и постановщика задачи, и собственно программиста.

Как это сделать? Во-первых, в институте обязательно должен быть класс, а лучше — несколько классов, оборудованных современной компьютерной техникой и открытых для свободного посещения студентами в вечернее время. Во время работы класса в нем должен присутствовать преподаватель-консультант, причем это дежурство должно входить в учебную нагрузку, поскольку работа такого консультанта не менее напряженная, чем, например, на семинаре.

Однако только желания работать на компьютере и возможности этим заниматься еще мало. Нужна задача, решая которую с помощью компьютера студент получал бы удовлетворение. (Моральное, но было бы совсем хорошо, чтобы и материальное.) Поставить перед студентом такую задачу сможет только преподаватель-специалист, сам знакомый с основами работы на персональных компьютерах. Но вмените в обязанность преподавателю такую работу, и вы получите очередную профанацию. Воистину: лучше меньше, да лучше! Конечно, заинтересовать преподавателя в совместной со студентами работе над программным обеспечением нужно. Хорошая, большая программа, дающая новое качество при обучении, должна цениться не меньше, чем монография, а руководство серией студенческих работ вполне обоснованно можно было бы приравнять к методичке. Стимулом должен быть умеренный «пряник», но ни в коем случае не «кнут».

Перечитав предыдущие строки, я еще раз убедился в том, сколь глубоко сидит в нас, начинавших на «Минсках» и «ЕСах», представление о том, что работа на компьютере непременно связана с программированием. А почему бы не поручить студенту вначале набирать с помощью компьютера тексты отчетов, статей или методичек, научив его предварительно с помощью тренажера слепому десятипальцевому методу работы с клавиатурой? Затем студент смог бы помогать вести базы данных, проводить с помощью электронных таблиц обработку результатов экспериментов, готовить в графическом редакторе иллюстрации. Таким образом, осваивая компьютер, студент помогал бы разгрузить преподавателей и научных сотрудников от рутинной работы, для которой вечно не хватает вспомогательного персонала.

Как завершающий элемент системы внедрения компьютеризации в вузе, необходимо особое подразделение, состоящее из программистов-консультантов. Основной задачей такого подразделения должна быть подготовка преподавателей и научных сотрудников в области применения компьютеров и дальнейшая консультационная помощь им.

Почему это нельзя поручить преподавателям кафедры вычислительной техники (или аналогичной)? Потому что учебная нагрузка у них слишком велика, а для того чтобы стать и оставаться классным специалистом в области применения компьютеров, надо большую часть рабочего дня (а подчас и ночи) проводить за экраном компьютера или за изучением документации.

Тогда, может быть, поручить функции консультантов программистам вычислительного центра? Увы, тоже нельзя. По трем причинам. Во-первых, консультантом должен быть преподаватель, педагог. Ведь научить работе на компьютере пожилого профессора куда сложнее, чем десятка студентов. Во-вторых, над каждым профессиональным программистом дамокловым мечом висит Его Величество План. И чтобы его выполнять и перевыполнять, программист вынужден специализироваться на каком-то одном классе задач. А консультанту нужен широкий кругозор. Нужно иметь возможность, не оглядываясь особо на план, сегодня заниматься базой данных, завтра работать с графическим редактором, послезавтра браться с компьютерным вирусом и т. д. Нужно также иметь возможность изучать литературу, посещать выставки, ездить на стажировки за границу. И ведь только такой скажочной жизнью и можно привлечь в институт действительно хороших профессионалов, тех, которые в любом кооперативе или СП могут получать на порядок больше, чем в состоянии дать им институт. И это третья причина, почему нужно особое подразделение программистов-консультантов.

Однако кроме консультаций, посещений выставок и заграничных поездок сотрудники этого подразделения должны еще чем-то заниматься. Чем? Очевидно, разработкой таких программ, которые ввиду их сложности не могут быть созданы студентами и преподавателями. Это могут быть программы двух типов. Тип первый — программы, выполняемые по заданию научно-исследовательского сектора или для обслуживания организации учебной и хозяйственной работы института, т. е. программы, непосредственно в учебном процессе не используемые. Тип второй — учебные и обучающие программы. Поговорим о них поподробнее.

Какого рода учебные программы могут существенно улучшить учебный процесс и, таким образом, оправдать огромные расходы на их создание и эксплуатацию? Ясно, что эти программы должны быть свободны от трех описанных выше синдромов — «синдрома белочки», «синдрома книги» и «синдрома кино». Добавим сюда еще синдром черного ящика,

когда известны исходные данные, получаются результаты, но вот как — не ясно.

Честно говоря, из всех программ, которые я видел (и создавал), единственной программой, вполне свободной от перечисленных четырех синдромов, был тренажер клавиатуры. Но, может быть, именно здесь зарыта собака? Может, это и есть та самая незаполненная ниша в учебном процессе, заполнить которую призван компьютер? Я имею в виду программы-тренажеры.

Приведем в качестве примера идеи трех сценариев таких программ, построенных на базе математической модели некоторого процесса.

Сценарий 1. Широко известны компьютерные игры, в ходе которых моделируется управление автомобилем. Остается только направить «автомобиль» на улицы «города», смоделировать различные дорожные ситуации и проверять при этом соблюдение правил дорожного движения.

Сценарий 2. Тренажер агронома. Моделируется объект управления — поле или даже целое хозяйство, чтобы тренироваться в организации севооборота. Моделируются внешние воздействия: погода, наличие или отсутствие на складе семян разных сортов, удобрений, величина госзаказа и продналога, технические возможности проведения агромероприятий и прочее. В результате увлекательной игры будущий агроном сможет вывести «электронное хозяйство» в миллионеры или разорить. И в том и в другом случае он приобретет необходимый опыт. Конечно, такой тренажер не заменит реального общения с землей, но ...

Сценарий 3. Тренажер врача. Моделируется больной. Врач может направлять его на анализы, рентген, назначать курс лечения или операцию. Можно смоделировать и случайные явления, типа отсутствия в аптеке нужных лекарств или халатность при проведении анализа. В результате — либо «выздоровление», либо ... Но рискует при этом лишь студент и лишь возможностью не получить зачет.

Вы скажете, что для разработки любого такого тренажера потребуются проведение научно-исследовательских работ, труд многих людей, большое время. Да, но и эффект будет куда больше, чем от сотни электронных «белочек».

Остановимся несколько подробнее на классе программ, наиболее характерном для технических вузов — на расчетных задачах. Эти задачи можно условно разделить на два подкласса: задачи простые и сложные.

К относительно простым задачам можно отнести те, для решения которых достаточно

рассчитать одно или несколько уравнений и ответ получается однозначным. При этом задачи ЭВМ сводятся к функциям калькулятора. Как правило, программы для решения такого вида задач являются собой классический пример «черного ящика». Нужны ли подобные задачи в учебном процессе? Безусловно, однако с одним «но». Пользу они приносят не при эксплуатации, а при их создании. Ибо для того, чтобы разработать программу, необходимо хорошо разобраться в методике расчета. Эффективность же использования «черных ящиков» в учебном процессе, как правило, отрицательная. Увы! Лозунг большинства студентов: «Не знать, а сдать». И в этом им великолепно помогают «черные ящики».

Совсем другое дело «сложные» программы, в которые ряд «простых» программ входят как элементы, позволяющие устранить рутинные этапы расчета, а при принятии решения требуется вмешательство человека.

В качестве примера такой программы можно привести программу выбора подшипников качения, разработанную в МИИСП. Эта программа моделирует работу с каталогом подшипников. Студент шаг за шагом вводит исходные данные, выбирает коэффициенты по таблицам, которые приводятся на

экране компьютера, но они в точности соответствуют аналогичным таблицам в справочнике-каталоге. Студент подбирает схему подшипниковых опор, тип подшипников, с помощью базы данных (не автоматизированной) выбирает модель подшипников и в результате расчета получает их долговечность. Программа ведет студента, подсказывает, контролирует правильность выбора коэффициентов, производит элементарные вычисления, сообщая при этом промежуточные результаты и формулы, по которым ведется расчет. Но основную, творческую часть работы выполняет сам студент. Таким образом, программа служит своеобразным тренажером по расчету подшипников. Конечно, эту программу можно было бы во многом автоматизировать, упростить, ускорить работу с ней. Но обучающий эффект от этого бы уменьшился.

Итак, нужен ли сегодня компьютер в вузе? Из первой части статьи выходит, что вроде бы не нужен, из второй — что нужен. Где истина? А истина, по-видимому, в том, что сейчас вполне можно обойтись и без компьютера, сэкономить на этом уйму денег и ... оказаться завтра у разбитого корыта.

## Электронный переводчик :

- 8 языков: русский, английский, немецкий, испанский, французский, итальянский, португальский, польский
- 50 000 слов
- инструкция на русском языке
- карманный размер
- часы, калькулятор, перевод валют
- автоматическое отключение
- незаменим в библиотеке, зарубежной поездке, для преподавателей и студентов,



Цена 290 000 рублей. Высылается наложенным платежом.

Заявки принимаются по адресу:  
125438, Москва, а/я 51, Маслову А.Н  
Тел.: (095) 454-00-35 (с автоответчиком).

# ЛинТех

## ВЫСОКОСКОРОСТНЫЕ ЛОКАЛЬНЫЕ СЕТИ И УЧЕБНЫЕ КОМПЬЮТЕРНЫЕ КЛАССЫ.

Наши сетевые системы всегда избавят Вас от всех проблем локальных сетей "Корвет" и "УКНЦ" - постоянные "зависания" и низкая скорость перестанут мешать Вашей работе.

ОС "NET-CP/M В-2.0" для КВТ "Корвет" с "IBM"-совместимой головной машиной. Система включает в себя комплект высокоскоростных сетевых адаптеров и программное обеспечение, позволяющее работать на каждом РМУ в MS-DOS-подобной системе.

Двадцатикратное повышение скорости передачи данных, гибкое разграничение доступа, древовидная файловая структура, набор системных команд и организация bat-файлов, Вы изучаете MS-DOS. Работа с нашей системой, Вы изучаете прикладных программ и подробные пакеты установки и использования ОС. Гарантия - 3 года. Цена - \$600, оплата в рублях по курсу ММВБ.

ОС "NET-RT11 В-1.0" для КВТ "УКНЦ" совместимой с IBM-головной машиной. Система обеспечивает работу на РМУ в стандартной RT11 без сбоев и зависаний, характерных для сети УКНЦ. Скорость передачи данных 375-1400 кбит/с, гибкие возможности разграничения доступа при минимальных переделках стандартной сети.

Гарантия - 3 года. Цена - \$600, оплата в рублях по курсу ММВБ.

**Покупая "NET-CP/M В-2.0" или "NET-RT11 В-1.0", Вы не получите "кота в мешке" - если система Вам не понравится, в течение 3-х месяцев покупки и получить деньги обратно.**

Классы PC AT/386.  
Широкий спектр возможных конфигураций на базе PC AT/386 по выбору заказчика. Базовый вариант - 10 РМУ 386SX-40, ОСУ 1Mb и 1 РМП 386DX-40, ОСУ 2 Mb, цветные SVGA мониторы, локальная сеть 2,5 мбит/с. На РМП установлен винчестер 170 Mb.

Гарантия - 1 год. Цена - \$7900, оплата в рублях по курсу ММВБ.

Адрес для  
корреспонденции:

119501, Москва, з/п 942.  
Телефон/факс: 273-50-14  
E-mail: shop@litech.msk.su



# ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

**С. З. Свердлов,**

*Вологодский государственный педагогический институт*

## КАК НАРИСОВАТЬ ПРОГРАММУ

Многим наверняка уже доводилось рисовать на экране компьютера, используя какой-либо графический редактор. Это одно из любимых занятий взрослых и детей и к тому же важный компонент компьютерной грамотности [1]. Нарисованную цветную картинку можно использовать для демонстрации с экрана или включить в свою программу. Это могут быть заставка, фон, персонажи, иллюстрации и многое другое. Именно таким способом создается графическое оформление программ профессионального уровня, в том числе игр [2]. Но для использования рисунка нужно знать структуру формата файла, в котором этот рисунок сохранен графическим редактором, или располагать специальной библиотекой программ для работы с данным форматом. На IBM PC существует большое количество популярных форматов для хранения изображений: PCX, TIFF, GIF и др. Но даже если имеется их подробное описание [3,4], требуется достаточно высокая квалификация, чтобы им воспользоваться. Нужно иметь в виду и то, что обычно изображение сохраняется в растровой форме, занимая много места на диске и в памяти. Причем с ростом разрешающей способности дисплеев объем графических файлов растрового формата увеличивается.

Какую же технологию подготовки рисунков для их последующего включения в прикладные программы выбрать? Нас интересуют те подходы, которые доступны

для понимания и использования начинающими программистами, не требуют значительных ресурсов компьютера (объема памяти, емкости дисков, высокого быстродействия) и не требуют использования нестандартных библиотек программ, по возможности не привязаны к определенному типу дисплея и/или графическому режиму. (При этом речь не идет о разработке программ со сложными видеоэффектами, анимацией профессионального качества и т. п.)

Рассмотрим для начала наиболее известные пользователям IBM PC графические редакторы. Их можно разделить на несколько групп. К первой относится популярное семейство PC PaintBrush, Hewlett-Packard PaintBrush, Photo Finish, Microsoft Windows PaintBrush и др. Их общей чертой является поддержка растровых форматов изображения, в первую очередь PCX (последний был создан фирмой ZSoft специально для PC PaintBrush). К ним примыкает ряд других систем, более простых и поэтому лучше работающих на компьютерах скромных конфигураций. В первую очередь это редактор Dr. Halo, отличающийся исключительно простым и удобным интерфейсом пользователя, благодаря чему он широко известен и даже занимает призовые места в рейтингах популярности графических систем [5].

Однако, несмотря на богатый набор возможностей, создать в этих редакторах хороший рисунок, оказывается, непросто. Сказывается отсутствие «откатки» более

чем на один шаг и сам растровый принцип хранения изображения, делающий невозможными операции с отдельными линиями и фигурами. Работа ведется, как правило, с отдельными точками. Действительной сферой использования редакторов такого класса является доводка сканированных изображений для их последующего использования в издательских системах. Кроме того, формат РСХ обладает сильной аппаратной зависимостью, а объем файлов изображений оказывается довольно большим. Редакторы этого класса удовлетворительно работают на компьютере не слабее IBM PC/AT-286, а многие — лишь под управлением Microsoft Windows. Для наших целей этот тип графических редакторов не очень подходит.

Другая группа программ для рисования на IBM-совместимых компьютерах представлена мощной графической системой Corel Draw [6]. Этот пакет, обладающий большими возможностями и работающий под управлением Microsoft Windows, требует нескольких мегабайт оперативной памяти и десятки мегабайт на винчестере. В нем используется принципиально иной способ представления рисунка, который хранится не как совокупность точек (растр), а как набор графических элементов — линий, фигур, надписей. Это позволяет сделать более удобным его редактирование и гораздо лучше подходит для создания новой иллюстрации, а не ретуширования. Важно и то, что файл рисунка в таком векторном формате может быть гораздо меньшего размера, чем в растровом.

Существенно ограничивает использование многих известных систем отсутствие шрифтов кириллицы и пользовательский интерфейс на английском языке. В последнее время многие фирмы — производители программного обеспечения предприняли усилия по русификации своих графических пакетов, но относится это в основном к программам для Microsoft Windows.

Как можно видеть, все рассмотренные нами графические редакторы не вполне удовлетворяют сформулированным ранее требованиям, а значит, мы приходим к не-

обходимости разработки новой системы. Что же хотелось бы при этом получить? Графический редактор, такой же простой и удобный, как Dr. Halo, но использующий векторный формат изображения и предоставляющий дополнительные возможности редактирования при создании иллюстраций. Используемый формат должен быть простым, а его описание доступно пользователям.

Эти пожелания были положены в основу создания графического редактора Турбограф [7]. Искомый набор графических примитивов и способов описания их параметров оказался буквально под рукой. Никто не будет отрицать, что самой популярной графической библиотекой для IBM PC (по крайней мере, в нашей стране) является разработка фирмы Борланд, входящая в состав систем программирования Турбо Паскаль (начиная с версии 4.0) и Турбо Си, а также Турбо Си++, Борланд Паскаль и Борланд Си++. Именно набор процедур этой библиотеки и был положен в основу формата представления рисунка в графическом редакторе Турбограф. Это позволило очень легко решить проблему воспроизведения рисунка, записанного в таком формате. Более того, стало возможным осуществить самый простой способ включения такого рисунка в прикладную программу. Действительно, поскольку набор графических примитивов совпадает с перечнем процедур графической библиотеки, не составляет труда сохранить созданный рисунок в виде фрагмента программы на Паскале или Си, а затем использовать этот фрагмент в прикладной программе.

Идея графического редактора, который сам пишет текст программы, воспроизводящей нарисованное, вообще говоря, не нова. Возможно, читателям знаком редактор «Чертило» Э. С. Рудакова, с помощью которого на «Ямахе» можно «нарисовать» программу на Бейсике. Первое знакомство с этим редактором поначалу вызвало улыбку: «Остроумно, но наивно — можно ведь просто запоминать растр». Однако в дальнейшем, когда встал вопрос о выборе графической разработки для ис-

пользования в учебном процессе на IBM, мы в качестве прототипа остановились именно на «Чертиле» — просто, понятно, удобно, практично и надежно (он используется на кафедре информатики ВГПИ уже пять лет и весьма популярен).



Рис. 1. Графический редактор Турбограф

автоматическое получение текстов программ на Паскале и Си, воспроизводящих созданный рисунок (рис. 2);

автоматическое создание исполняемой (.EXE) программы, воспроизводящей рисунок;

полная совместимость с компиляторами фирмы Борланд (Турбо Паскаль 4.0-7.0, Турбо Си, Турбо Си++, Borland Pascal 7.0, Borland C++);

минимальные требования к компьютеру (IBM PC/XT, CGA, мышь, 640 кб ОЗУ, 700 кб на диске для полного комплекта или 240 кб в минимальной конфигурации);

поддержка дисплеев CGA, MCGA, EGA, VGA, SVGA, Hercules и всех графических режимов, используемых библиотекой фирмы Борланд (рис. 3);



Рис. 3. Диалоговое окно выбора графического режима

В настоящее время создана и распространяется версия 1.02 (рис. 1). Ее отличительные черты:

реализация технологии визуального программирования на Турбо Паскале, Турбо Си и Турбо Си++;

простой и понятный пользовательский интерфейс на русском языке;

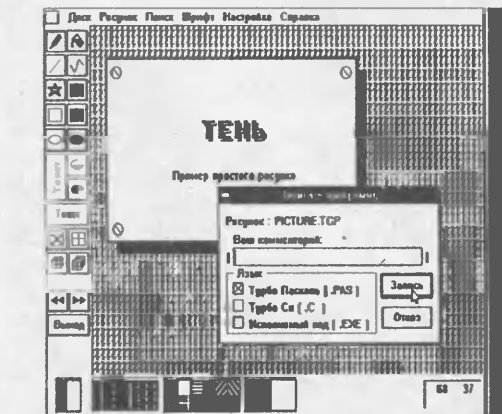


Рис. 2. Запись текста программы

простые и вместе с тем мощные возможности редактирования:

- откатка назад на любое число шагов;
- поиск, вставка и удаление отдельных линий, фигур и текста;
- удаление, перенос, масштабирование, копирование рисунков и их частей;

поддержка русских шрифтов формата VGI;

дополнительная утилита для просмотра и демонстрации рисунков с возможностью создания слайд-фильмов;

полная открытость используемого

формата файлов рисунка, их подробное описание, исходный текст программных модулей поддержки;

преобразование программ на Паскале в рисунки добавлением в их текст одной буквы (рис. 4).

В качестве примера приведем текст программы на Паскале, полученной автоматически с помощью Турбографа (см. рис.2):

```
program Picture;
{ Программа получена с помощью графической системы }
{ Турбограф 1.02 Copyright (c) 1993, 94 С. Свердлов }
```

uses

Graph, Crt;

const

FillPattern1 : FillPatternType = ( 7, 223, 208, 215, 215, 215, 215, 255 );

var

```
grDriver   : integer;
grMode     : integer;
ErrCode    : integer;
Key        : char;
```

procedure DrawPicture;

begin

```
SetFillPattern( FillPattern1, 1 );
Bar( 0, 479, 639, 0 );
SetFillStyle( InterleaveFill, 1 );
Bar( 105, 288, 470, 73 );
SetFillStyle( SolidFill, 1 );
Bar( 91, 274, 456, 59 );
SetColor( 0 );
SetLineStyle( SolidLn, 0, ThickWidth );
Rectangle( 91, 274, 456, 59 );
SetTextStyle( DefaultFont, HorizDir, 3 );
OutTextXY( 232, 136, 'ТЕНЬ' );
SetColor( 1 );
OutTextXY( 230, 134, 'ТЕНЬ' );
SetColor( 0 );
OutTextXY( 229, 133, 'ТЕНЬ' );
SetLineStyle( SolidLn, 0, NormWidth );
FillEllipse( 106, 76, 7, 7 );
```

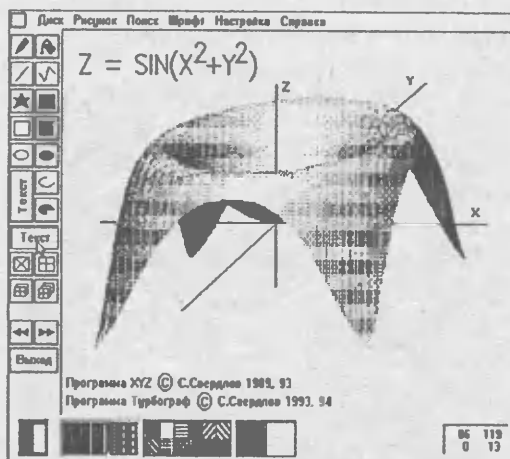


Рис. 4. Изображение поверхности для функции двух переменных в рабочем поле Турбографа



```

Line( 101, 71, 111, 81 );
FillEllipse( 107, 259, 7, 7 );
Line( 102, 254, 112, 264 );
FillEllipse( 440, 259, 7, 7 );
Line( 435, 254, 445, 264 );
FillEllipse( 440, 76, 7, 7 );
Line( 435, 71, 445, 81 );
SetTextStyle( InstallUserFont('SYSV'),
HorizDir, 4 );
'OutTextXY( 189, 187, 'Пример
простого рисунка' );
end;

```

**begin**

```

grDriver := MCGA;
grMode := 5; { 640 x 480, черно-белый }
InitGraph( grDriver, grMode, '' );
ErrCode := GraphResult;
if ErrCode = grOk then begin
    DrawPicture;
    Key := ReadKey;
    CloseGraph;
end
else
    WriteLn('Ошибка графики: ', GraphErrorMsg(ErrCode));

```

**end.**

С помощью редактора и дополняющих его инструментальных средств может быть получен ряд интересных эффектов, а также выполнены преобразования рисунков различных форматов (рис. 4, 5).

### Литература

1. *Каймин В. А. и др.* Основы информатики и вычислительной техники: Проб. учеб. для 10—11 кл. ср.шк. М.: Просвещение, 1990.
2. *Чикин Д.* Пристальный взгляд на компьютерные игры // Монитор. 1993. № 5.
3. *Романов В. Ю.* Популярные форматы файлов для хранения графических изображений на IBM PC. М.: Унитех, 1992.
4. *Тихонов А.* Показать картинку // Монитор. 1992. № 3.
5. *Тайнен Д.* «Мировой класс-1992»: итоги конкурса // Мир ПК. 1993. № 4.



Рис. 5. Изображение графического редактора Dr. Halo в рабочем поле Турбографа

6. Пользователю CorelDRAW 4.0. Санкт-Петербург: Изд-во «Макет» 1993.

7. *Свердлов С. З.* Объектно-ориентированный графический редактор для компиляторов Borland // Материалы Международной конференции «Применение новых компьютерных технологий в образовании». Троицк: ТРИНИТИ, 1993.

*Желающие приобрести программу Турбограф могут обращаться в редакцию по телефону : (095) 208-30-78.*

**В. В. Илларионов,**

*г. Балашов*

## ГРАФИЧЕСКИЕ ТЕХНОЛОГИИ В ТУРБО-ПАСКАЛЕ СИСТЕМЫ MSX

Общеизвестно, что прорисовка изображений для обучающих и игровых программ — трудоемкая задача. Существенно облегчить ее могут графические редакторы. В настоящей работе рассматривается применение известного знакомого графического редактора GRED, работающего на компьютерах систем MSX-1 и MSX-2 («Ямаха»).

Суть его в том, что он рисует не в обычном режиме «карандаша», а шаблонами 8x8 пикселей, причем форма и цвет этих шаблонов задаются и их можно менять. Выигрыш такого метода рисования в том, что записанные в файл картинки имеют весьма небольшие размеры. Чтобы понять принцип работы редактора GRED, кратко рассмотрим организацию изображения в режиме SCREEN 2. Этот режим можно рассматривать как совокупность трех текстовых экранов, каждый из которых имеет размеры 32x24. Это означает разбиение таблицы имен образов PNT, состоящую в SCREEN 2 из 768 байт, на три части по 256 байт. В каждом байте записывается код того символа, который будет отображаться на данном знакоместе. Таких кодов, очевидно, 256, а форма для каждого такого символа записывается в таблице шаблонов образов-PGT, по 8 байт на символ, как в обычном текстовом режиме. Таблица PGT, состоящая из 6144 байт, делится на три независимые части по 2048 байт. Каждому байту таблицы PGT соответствует один байт таблице цветов СТ. Четыре старших бита этого байта кодируют цвет изображения, а четыре младших бита кодируют цвет фона для сегмента из восьми пикселей, который находится в каком-то знакомстве.

А теперь рассмотрим, как кодирует изображение редактор GRED. При записи рисунка нужно записать два основных файла — с расширением .NAM и .PAT. Файл с расширением .NAM имеет размер 768 байт — это записана таблица PNT. Файл с расширением .PAT имеет размер 4096 байт, это так называемая библиотека. Что же записывается в эту библиотеку? Дело в том, что редактор GRED «разре-

шает» работать только с 256 шаблонами, т. е. только с одной третью таблиц PGT и СТ. А остальные две трети он просто дублирует. На этом построена вся игра, и поэтому компактно кодируется изображение, естественно, ценой потери некоторого разнообразия. В файл .PAT записываются одна треть таблицы PGT (2048 байт) и одна треть таблицы СТ (2048 байт). Таким образом и получаются 4096 байт файла .PAT.

Итак, мы нарисовали рисунок и знаем, как устроены таблицы в режиме SCREEN 2. Осталось перенести содержимое файлов .NAM и .PAT в видеопамять, и на экране появится тот же рисунок, который был виден в редакторе.

Очень хорошо подходит для решения такой задачи система программирования Турбо-Паскаль, обладающая развитой файловой системой и порождающая быстрый код, что существенно для многих игровых и обучающих программ.

Общая последовательность наших действий:

- 1) грузим файлы .NAM и .PAT в массивы;
- 2) включаем режим SCREEN 2;

3) пересылаем содержимое загруженных массивов в видеопамять;

Очевидно, нам необходимо иметь хотя бы основные процедуры для работы с графикой: включение режима SCREEN 2, включение режима SCREEN 0, пересылка из памяти в видеопамять, установка цвета. Обычно все такие процедуры имеются в графических пакетах. Однако для случая, если читатель не имеет возможности воспользоваться готовым графическим пакетом, ниже приводится минимальный графический пакет, который автор называет SMALL GRAPH. Этот пакет записывается в файл с именем SGRAPH.PAS и под таким именем подключается к основной программе.

```
{ Графический пакет SMALL GRAPH }
{$U- }
{ Процедура установки цвета }
Procedure Color (x,y,z:byte);
Begin
  mem[$F3E9]:=x; mem[$F3EA]:=y;
  mem[$F3EB]:=z;
  inline ($F7/0/$62/0)
End; { color }
```

```

{ Процедура установки графического
SCREEN 2 }
Procedure GraphMode;
Begin
  Inline ($F7/0/$72/0)
End; { graphmode }
{ Процедура установки текстового режима
SCREEN 0 }
Procedure TextMode;
Begin
  Inline ($F7/0/$6C/0)
End; { textmode }
{ Процедура пересылки блока данных из ОЗУ
в видеопамять }
{ ad - адрес блока в ОЗУ, vad - адрес в
видеопамяти }
{ dlina - длина блока }
procedure vwrite(ad,vad,dlina:integer);
begin
  inline($2A/ad/$ED/$5B/vad/$ED/$4B/
dlina/$F7/0/$5C/0);
end; { vwrite }

```

Процедура vwrite написана на базе функции BIOS для компактности. Однако она не самая быстрая. В графических пакетах такая функция обычно напрямую работает с портами видеопроцессора, что повышает скорость работы.

А теперь приведем программу, которая воспринимает два рисунка и выводит поочередно их на экран, соответственно нажатой клавише Ввод или Пробел. Первый рисунок должен иметь имя PRIM.NAM, второй — PRIM2.NAM. Их общая библиотека — PRIM.PAT. Комментарии к процедурам поясняют смысл того, что они осуществляют. Отметим, что файлы мы должны описывать как бестиповые. Для работы с ними применяются процедуры blockread, blockwrite, работающие с блоками по 128 байт.

```

{ Программа вывода изображения }
{ подготовленного редактором GRED }
{$I SGRAPH } { Подключаем графический
пакет }
{$U- } { отключаем проверку CTRL+STOP }
const dlnt=768; { длина таблицы PNT }
      dlpgt=2048; { длина одной трети таблиц
PGT, CT }
      vadpnt=$1800; { начальный адрес таб-
лицы PNT }
      vadct=$2000; { начальный адрес таб-
лицы CT }
      countgr=24; dlst=32; { 32*24=768 }
type tpfile=file;

```

```

type buffer_pnt=array[1..countgr,1..dlst] of
byte;
var bufpnt1: buffer_pnt; { буфер для 1-го ри-
сунка }
      bufpnt2:buffer_pnt; { буфер для 2-го рисун-
ка }
      bufpgt: array[1..dlpgt] of byte; { буфер для
трети таблицы PGT }
      bufct: array[1..dlpgt] of byte; { буфер для
трети CT }
{ процедура загрузки из файла .NAM в буфер }
procedure loadbufnam(var f:tpfile;var
bufpnt:buffer_pnt);
const countid=6; { число считываний }
begin
  blockread(f,bufPNT,countid);
end; { loadbufnam }
procedure loadbufpat(var f:tpfile); { загрузка
PGT и CT }
const countid=16;
begin
  blockread(f,bufPGT,countid);
  blockread(f,bufCT,countid);
end; { loadbufpat }
{ Пересылка из буфера bufpnt в таблицу PNT }
procedure wrScrPNT(var bufpnt:buffer_pnt);
const vadpnt=$1800; dlnt=768;
begin
  vwrite(Addr(bufpnt),vadpnt,dlnt);
end; { wrScrPNT }
{ Заполнение таблицы PGT из буфера bufpgt }
procedure wrScrPGT;
const vad: array[1..3] of integer=(0,2048,
4096);
      dl=2048;
var i:byte;
begin
  for i:=1 to 3 do
    vwrite(Addr(bufpgt),vad[i],dl);
  end; { wrscrpgt }
{ Заполнение таблицы CT из буфера bufCT }
procedure wrScrCT;
const vad: array[1..3] of integer=($2000,
$2800,$3000);
      dl=2048;
var i:byte;
begin
  for i:=1 to 3 do
    vwrite(Addr(bufCT),vad[i],dl);
  end; { wrscrct }
var f1,f2,f3:tpfile;
      ch:char;
      i:integer;
begin
  { Открываем файлы }
  assign(f1,'PRIM.NAM'); reset(f1);
  assign(f2,'PRIM2.NAM'); reset(f2);
  assign(f3,'PRIM.PAT'); reset(f3);

```

```

{ Грузим картинку }
loadbufnam(f1,bufpnt1);
loadbufnam(f2,bufpnt2);
{ Грузим библиотеку }
loadbufpat(f3);
graphmode; { Включаем режим SCREEN 2 }
color(1,1,1); { фоновый цвет }
{ Пересылаем библиотеку в таблицы PGT
и СТ }
wrScrPgt; wrScrCt;
{ Пересылаем первую картинку на экран }
wrScrPNT(bufpnt1);
ch:=#0;
while ch #27 do { пока не нажата клавиша, цикл }
begin
read(kbd,ch); { читаем клавишу }
case ch of
#13: wrScrPNT(bufpnt1); { выводим
первую картинку}
#32: wrScrPNT(bufpnt2); { выводим
вторую картинку}
end; { case }
end;
textmode; { включаем SCREEN 0 }
color(15,12,4);
end.

```

Запустив вышеприведенную программу, убедимся, что смена кадров происходит практически мгновенно. Что эквивалентно экраным страницам, которых здесь может быть не четыре, как в SCREEN 5, а значительно больше, ведь одна страница при неизменной библиотеке занимает всего 768 байт. Отметим, что таблица PNT в режиме SCREEN 5 занимает 27136 байт.

В реальной игровой или обучающей программе может быть довольно много картинок. Хранить их в разных файлах и загружать оттуда весьма неудобно. В этом случае можно рекомендовать склеить их в один, заодно подключив туда и библиотеку. Склеюку двоичных файлов можно осуществить с помощью команды COPY операционной системы MSX-DOS, добавляя ключ /b после имени склеиваемого файла, например: COPY имя1/b+имя2/b имя\_рез. При загрузке картинок их можно размещать не только в массивах, но и перебросить их в область видеопамати за 16 Кбайт начальной видеопамати. В этом случае мы теряем совместимость с MSX-1, но экономим обычную память. При пересылке из области хранения соответствующей картинке теперь будет нужна процедура пересылки из видеопамати в память, чтобы затем переслать в видеотаблицу. Приведем текст

соответствующей процедуры на тот случай, если ее нет у читателя:

```

{ vad - видеоадрес, ad - адрес, dlina - длина
блока }
procedure vread(vad,ad,dlina:integer);
begin
inline($2A/vad/$ED/$5B/ad/$ED/$4B/dl
ina/$F7/0/$59/0)
end; { vread } .

```

Еще лучше использовать высокоскоростную пересылку VRAM-VRAM, если она есть.

Отметим, что после того, как картинка выведена на экран, теперь уже нельзя произвольно применять обычные процедуры типа рисования линии, окружности и т. д., так как это может разрушить таблицу образов PGT. Нельзя также выводить текст с помощью базовых процедур. Все тексты должны быть набраны в редакторе вместе с рисунком. Если все же требуется заняться рисованием, то нужно выделить для этой области экрана окно и очистить соответствующую часть таблицы шаблонов.

Однако можно свободно пользоваться спрайтами, так как они не нарушают таблиц PNT, PGT и СТ. На общем фоне нарисованной картинке можно эффективно использовать возможности динамической графики. Все выше приведенные процедуры загрузки и отображения будут правильно работать и в режиме SCREEN 4. В этом случае расширятся возможности работы со спрайтами. Процедуру включения режима SCREEN 4 можно взять из любого графического пакета.

Во время пересылок таблиц возможны мелькания на экране. При желании на это время экран можно отключить, а затем включить. Приведем соответствующие процедуры:

```

procedure DisScr; { выключение экрана }
begin inline($F7/0/$41/0) end;
procedure EnaScr; { включение экрана }
begin inline($F7/0/$44/0) end; .

```

В заключение отметим, что в настоящее время имеются знакоместные графические редакторы и для старших графических режимов, в частности для SCREEN 5. Фактически они эмулируют режим SCREEN 2, программно выполняя то, что в режиме SCREEN 2 осуществляется аппаратно. Процедуры работы с такими изображениями будут значительно сложнее.

Н. Л. Демкина,

г. Москва

## КОНКУРС «ПОЛЬЗОВАТЕЛЬ ЭВМ-94»



ми его проведения стали Учебно-методический кабинет по профессиональному образованию Московского департамента образования (директор К. А. Тюркиова) и постоянно работающее городское объединение преподавателей информатики и вычислительной техники (руководитель — старший методист Н. Л. Демкина).

Обратим внимание на весьма важную особенность тематики конкурса. Среди программистов различного рода первенства сегодня не новость. Но программистом становится далеко не каждый, тогда как умение работать с компьютером и имеющимся на нем прикладным программным обеспечением, использовать ЭВМ в повседневной деятельности необходимо всем, без исключения, учащимся независимо от сферы их будущей деятельности, поручкой тому все большее проникновение вычислительной техники даже в, казалось бы, далекие от нее области: пищевую, швейную промышленность и т. п. Нужно учить студентов не только основам алгоритмизации и программирования, но и уметь работать с текстовыми и графическими редакторами, базами данных, подготавливать и печатать документы, обрабатывать другую информацию. К сожалению, всему этому в учебной программе уделяется слишком мало внимания.

Но вернемся к конкурсу «Пользователь ЭВМ — 94». Его основными задачами стали выявление творчески одаренной молодежи, в совершенстве владеющей навыками пользования ЭВМ, организация обмена опытом подготовки учащихся в области компьютеризации и пропаганда ее идей.

В конкурсе приняли участие 52 студента из 31 учебного заведения среднего профессионального образования. Участникам предлагалось выполнить ряд заданий, проявив свое умение использовать технические возможности ЭВМ, работать с сервисной оболочкой Norton Commander и с текстовым редактором (по выбору — Лексикон, Фотон или MS-Word), в течение полутора часов создать в графическом редакторе картинку на предло-

женную тему (пейзаж, портрет, натюрморт, иные миры) с максимальным использованием предоставляемых графических возможностей. Дополнительно предлагалось продемонстрировать заранее подготовленное домашнее задание — разработку на языке программирования, в какой-либо среде (например TeachCAD) или картинку, созданную в графическом редакторе.

Призерами конкурса «Пользователь ЭВМ — 94» стали:

**I место:** Шишкин Павел (московский пищевой колледж) и Торкунов Антон (колледж работников гостиниц и ресторанов, специалистов малого бизнеса «Царицыно»);

**II место:** Пилогин Виктор (московский коммерческо-банковский колледж) и Смирнов Кирилл (московский колледж городской инфраструктуры, экономики и права);

**III место:** Наумов Андрей (московский колледж архитектуры и менеджмента в строительстве), Блохин Игорь (московский экономико-технологический колледж) и Курбатов Олег (московское театральное художественно-техническое училище).

Помимо официальных наград редакцией журнала «Информатика и образование» были учреждены поощрительные призы за домашнее задание. При его оценке во внимание принимались прежде всего оригинальность заложенных в разработку идей и умение самостоятельно создавать полноценный программный продукт. Призерами конкурса домашних заданий стали следующие участники:

**I приз** (микрокалькулятор и годовой комплект журнала ИНФО за 1994 г.) за программу на ассемблере для анализа конфигурации ПЭВМ и автоматической генерации файла CONFIG.SYS: Наумов Андрей (московский колледж архитектуры и менеджмента в строительстве);

**II приз** (годовой комплект журнала ИНФО за 1994 г.) за систему управления базами данных на Паскале с возможностью хранения, кроме текстовых сведений, графических изображений: Торкунов Антон (колледж работников гостиниц и ресторанов, специалистов малого бизнеса «Царицыно»);

**III приз** (поощрительный) за обучающую программу на TeachCAD по искусству оригинали\*: Масина Вероника (московское педагогическое училище № 5).

В заключение отметим, что конкурс «Пользователь ЭВМ — 94» вызвал большой интерес у преподавателей и учащихся, стал стимулом к созданию новых оригинальных программных разработок. На наш взгляд, имеет смысл проводить подобные конкурсы ежегодно.

\* Изготовление фигурок из бумаги без использования ножниц и клея. — Прим. ред.

**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 — 50 000 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.

Для классов БК:

Программный пакет АЛЬТЕРНАТИВА-2 (БК-0011 всех типов). Состав: сетевой монитор, текстовые редакторы, электронные таблицы, базы данных, Турбо-Паскаль, игры, техническая и методическая документация.

Условия поставки:

Высылка по почте. Форма оплаты — любая (безналичная предоплата по счету, наложенный платеж и т.д.).

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.

# КЛУБ «АГАТ»

Н. А. Костарева,

г. Москва

## СИСТЕМЫ СЧИСЛЕНИЯ

Предлагаемая простая программа на Бейсике демонстрирует почти все операции над символьными переменными, выполняя перевод чисел из одной системы счисления в другую. Эта программа была составлена вместе со школьниками при изучении темы «Символьная переменная» и применяется в качестве контролирующей при сдаче зачета по теме «Системы счисления». Системы счисления могут быть заданы от двоичной до двадцатеричной. Для представления чисел используются цифры от 0 до 9 и заглавные буквы латинского алфавита.

### Назначение строк программы:

- 20 — ввод исходного числа в N\$;
- 30 — ввод основания входной системы счисления в SI;
- 40 — ввод основания выходной системы счисления в SO;
- 50 — обнуление переменных D и K\$:  
D — число в десятичной системе счисления,  
K\$ — результат;
- 60 — если заданная входная система счисления — десятичная (SI=10), то переход на строку 130;
- 70—120 — перевод числа из входной системы счисления (SI) в десятичную:  
Q — код одного символа числа,  
R — цифра числа;
- 130 — если выходная система счисления (SO) — десятичная, то вычисление K\$ и переход на 180 строку;
- 140—170 — перевод числа из десятичной системы счисления в выходную (SO):  
R — остаток при делении,  
Q — код буквы;
- 180 — вывод результата.

### Листинг:

```
10 REM PERSS20
20 INPUT "Введите число";N$
30 INPUT "Введите вх. систему счисления-";SI
40 INPUT "Введите вых. систему счисления-";SO
50 IF SI = 10 THEN D = VAL ( N$ ) : GOTO 130
60 D = 0 : K$ = ""
70 FOR I = 1 TO LEN ( N$ )
80 Q = ASC ( MID$ ( N$,I,1 ) )
90 IF Q > 64 THEN R = Q - 55 : GOTO 110
100 R = VAL ( MID$ ( N$,I,1 ) )
110 D = D + R * SI ^ ( LEN ( N$ ) - I )
120 NEXT I
125 IF SO = 10 THEN K$ = STR$ ( D ) : GOTO 180
130 IF D = 0 THEN PRINT " ОШИБКА ": GOTO 20
140 R = D - INT ( D / SO ) * SO + 0.5
150 IF INT ( R ) > 9 THEN Q = INT ( R ) + 55 : K$ = CHR$ ( Q ) + K$ : GOTO 170
160 K$ = STR$ ( INT ( R ) ) + K$
170 IF D >= SO THEN D = INT ( D / SO ) : GOTO 140
180 PRINT " РЕЗУЛЬТАТ-"; K$
190 END
```

Примечание. В программе используются операторы, имеющиеся в большинстве версий Бейсика, поэтому она может работать практически на любой ПЭВМ.

Н. В. Петрова,

г. Москва

## КОНТРОЛЛЕР «CARD-93» ДЛЯ ПЭВМ «АГАТ»

Эта разработка НТК «СПРАЙТ» позволяет вам работать на «АГАТе» с дискетами объемом 360, 720 и 800 кБ в формате IBM, обеспечивая полную совместимость данных (но, увы, не программ) между «АГАТом» и IBM PC.

Контроллер представляет собой плату, устанавливаемую в разъем магистрали ПЭВМ «АГАТ» (7 или 9) для управления накопителем типа EC5323.01 или аналогичных ему. Он обеспечивает совместимость на физическом уровне с форматами записи гибких магнитных дисков, применяемых в ПЭВМ IBM PC, YAMAHA, SINCLAIR, КВАНТ и им подобных.

Коммерческое программное обеспечение предоставляет возможность работы в среде, весьма напоминающей Norton Commander. В вашем распоряжении: оконный интерфейс, работа с подкаталогами, копирование, просмотр, редактирование, удаление, ряд утилит, в число которых входят декодировки текстовых файлов для различных таблиц символов.

### Справочный листок

Жгут подключения  
цветного монитора БЦ 32 (Винни-  
ца) к ПЭВМ «АГАТ-9»

БЦ 32 ВГ - 76	ОП	«АГАТ-9» ОП
1	В	1
2	6	6
3	Я	3
4, 5	земля	2
6	КСН	4
7	ОСН	5
9		7

Распайка жгута подключения  
принтера МС 6313  
к ПЭВМ «АГАТ-9»

МС 6313 ОП51-30		«АГАТ-9»	
		ОП 34	П 15-32
3	В0	A1 - C4	1 - 24
5	В1	A2	2
7	В2	A3	3
9	В3	A4	4
11	В4	A5	5
13	В5	A6	6
15	В6	A7	7
17	В7	A8	8
23	Stroke	A9	9
21	В8	B9	19
1-20	земля	C9	29
четн.		C2,3,5,6, 7,8,10	22,23,25,26, 27,28,30

Распайка жгута подключения  
принтера МС 6312  
к ПЭВМ «АГАТ-9»

МС 6312 ОП 101		«АГАТ-9»	
		ОП 34	П 15-32
2	В0	A2	2
3	В1	A3	3
4	В2	A4	4
5	В3	A5	5
6	В4	A6	6
7	В5	A7	7
8	В6	A8	8
9	В7	A9	9
1	Stroke	B9	19
11	В8	C9	29
10,21,25	земля	C2,3,4,5, 6,7,8,10	22-28,30

Распайка жгута подключения  
принтера МС 6337  
к ПЭВМ «АГАТ-9»

МС 6337 П 15	МРН 32	«АГАТ-9»		
		ОП34	ОП 34	П 15-32
3	3	В0	A1-C4	1-24
5	5	В1	A2	2
7	7	В2	A3	3
9	9	В3	A4	4
11	11	В4	A5	5
13	13	В5	A6	6
15	15	В6	A7	7
17	17	В7	A8	8
1	1	Stroke	A9	9
21	21	В8	B9	19
2-22	2-24	земля	C9	29
четн.	четн.		C2,3,5,6, 7,8,10	22,23,25, 26,27,28,30

Распайка жгута подключения  
монитора МС 6105  
к ПЭВМ «АГАТ» (Загорского 3-да)

	Монитор П 15-15	ВП ОП	Видео ОП
• 126	1,2	5	
земля	3,4	1	
земля	5,7	2	
земля	8		2
земля	11		2
Видео	12		3

Жгут вольтского исполнения

Распайка жгута подключения  
принтера «ROBOTRON»  
к ПЭВМ «АГАТ-9»

ROBOTRON		«АГАТ-9»	
		ОП-34	П 15-32
2	В0	A1 - C4	1 - 24
3	В1	A2	2
4	В2	A3	3
5	В3	A4	4
6	В4	A5	5
7	В5	A6	6
8	В6	A7	7
9	В7	A8	8
1	Stroke	A9	9
11	В8	B9	19
земля	земля	C9	29
		C2,3,5,6, 7,8,10	22-28,30

Внимание пользователей ПЭВМ «АГАТ».

Компьютерный центр ЮСН предлагает гарантийный и послегарантийный ремонт вашего «АГАТа», поэтапную его модернизацию. Консультации по телефону: (095) 267-32-47.



# КЛУБ «ЯМАХА»

В. А. Попов,

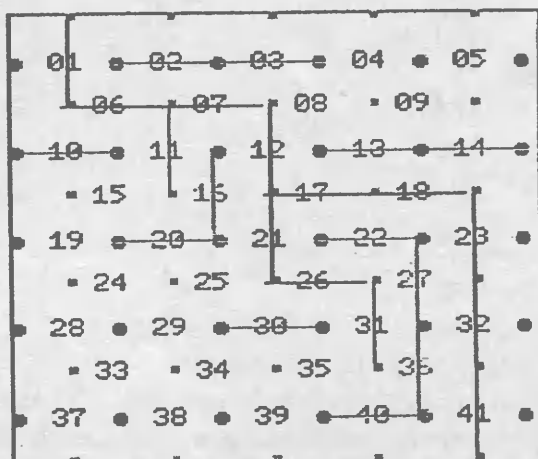
г. Сыктывкар

## «БРИДЖ-ИТ» НА «ЯМАХЕ»

Среди многочисленных разработок для программируемых микрокалькуляторов (ПМК) известна популярная игра «БРИДЖ-ИТ» — «Перекинь мостик». Однако в этой игре, кроме умения отыскивать «умные ходы», ПМК больше ничего не может. Человеку приходится самому не только рисовать игровое поле и изображать ходы, как свои, так и ПМК, но и следить за тем, кто же первый сумел перекинуть свой мостик. Кроме того, все ранее опубликованные программы ([1—3]) составлены так, что ПМК постоянно делает ход первым и (за счет алгоритма О. Гросса) всегда выигрывает. Как только дети узнают об этом, интерес к игре с ПМК мгновенно теряется.

В предлагаемой реализации игры «БРИДЖ-ИТ» на компьютере «Ямаха MSX-2» указанные «минусы» преодолены. Теперь права на первый ход у человека и у ЭВМ равны (в каждой нечетной партии первым начинает игру человек). При этом у компьютера можно выиграть (!), но, если допущен промах, «Ямаха» перехватит инициативу.

В программе предусмотрен вывод на экран заставки (строки 10 и 20), правил игры «БРИДЖ-ИТ» (70—190) и игрового поля (60 и 230—350). Казавшаяся вначале трудной задача выявления победителя получила легкое и «чисто компьютерное» решение: на вспомогательном поле (оно появляется в правом верхнем углу экрана) с помощью оператора PAINT и функции POINT после каждого совершенного хода определяется наличие замкнутой ломаной линии, соединяющей противоположные стороны игрового поля (строки 420 и 960—1030).



№ 2

СЧЕТ 1 : 1 МОЛОДЕЦ!

```

10 SCREEN 3:COLOR 7,1,1:CLS:OPEN "GRP:"AS#1
20 PSET(40,20),1:PRINT#1,"игра":PSET(0,60),1:PRINT#1,"БРИДЖ-ИТ"
30 GOSUB 1050:L=RND(-TIME)
40 DIMX(41),Y(41),X1(21),Y1(21),X2(21),Y2(21),X3(21),Y3(21)
50 DIMX4(21),Y4(21),M(21),N(21)
60 FOR N=1 TO 41:READ X(N),Y(N):NEXT
70 SCREEN 0:COLOR 13,4,4:CLS:WIDTH 80
80 PRINT"    ЦЕЛЬ ИГРЫ: СОВЕРШАЯ ХОДЫ, ТО ЕСТЬ СОЕДИНЯЯ ГОРИ
ЗОНТАЛЬНЫМИ ИЛИ ВЕРТИКАЛЬНЫМИ"
90 PRINT"ОТРЕЗКАМИ УЗЛОВЫЕ ТОЧКИ ИГРОВОГО ПОЛЯ, НЕОБХОДИМО
ПРОЛОЖИТЬ"
100 PRINT"ЗАМКНУТЫЙ ПУТЬ ('МОСТИК') ОТ ОДНОГО КРАЯ ПОЛЯ К ПРО
ТИВПОЛОЖНОМУ."
110 PRINT"    КОМПЬЮТЕР СОЕДИНЯЕТ ВЕРХНИЙ КРАЙ С НИЖНИМ, А
ИГРОК - БОКОВЫЕ."
120 PRINT"УЗЛОВЫЕ ТОЧКИ КОМПЬЮТЕРА ИМЕЮТ ВИД КВАДРАТИКОВ 2X2,"
130 PRINT"УЗЛОВЫЕ ТОЧКИ ИГРОКА ИМЕЮТ ВИД КРУЖОЧКОВ."
140 PRINT"    ДЛЯ СОВЕРШЕНИЯ ХОДА ИГРОК ДОЛЖЕН ВВЕСТИ ДВУЗНАЧ
НУЮ ЧИСЛОВУЮ"
150 PRINT"КОД, РАСПОЛОЖЕННЫЙ НА ИГРОВОМ ПОЛЕ МЕЖДУ СОЕДИНЯЕМЫ
МИ ТОЧКАМИ."
160 PRINT"    МЕЖДУ КОМПЬЮТЕРОМ И ИГРОКОМ ОРГАНИЗУЕТСЯ МАТЧ ИЗ
ЧЕТНОГО"
170 PRINT"КОЛИЧЕСТВА ПАРТИЙ, РАВНОГО 2XВ, ГДЕ В ОПРЕДЕЛЯЕТ ИГ
РОК."
180 PRINT"В КАЖДОЙ ЧЕТНОЙ ПАРТИИ НАЧИНАЕТ ИГРУ КОМПЬЮТЕР."
190 INPUT"    ВВЕДИТЕ НАТУРАЛЬНОЕ ЗНАЧЕНИЕ В":В
200 IF INT(В)=В AND В>0 THEN 210 ELSE 190
210 L=0:N#=""
220 FOR S=1 TO 2XВ:P=2-(SMOD2):K=1
230 *****ИГРОВОЕ ПОЛЕ*****
240 SCREEN 5:LINE (0,0)-(183,178),3,В
250 FOR N=20 TO 160 STEP 35
260 FOR M=0 TO 175 STEP 35
270 LINE (N,M)-(N+2,M+2),13,BF:NEXT:NEXT
280 FOR N=0 TO 200 STEP 35
290 FOR M=20 TO 160 STEP 35
300 CIRCLE (N+2,M),2,7:PAINT(N+2,M),7:NEXT:NEXT
310 FOR N=1TO9:PRESET (X(N)-6,Y(N)-4):PRINT#1,"0"
320 PRESET (X(N),Y(N)-4):PRINT#1,RIGHT$(STR$(N),1):NEXT
330 FORN=10TO41:PRESET (X(N)-6,Y(N)-4):PRINT#1,MID$(STR$(N),2,1)

```

```

340 PRESET (X(N), Y(N)-4):PRINT#1, RIGHT$(STR$(N), 1):NEXT
350 PRESET (200, 56), 4:PRINT#1, "N":S
360 ON P GOTO 370, 520
370 "''''''''''НАЧИНАЕТ ЧЕЛОВЕК''''''''''
380 GOSUB 690:IF I<>1 THEN 400
390 I=INT(RND(1)*4)+2:GOSUB 880:GOTO 410
400 I=1:GOSUB 880
410 K=K+1:GOSUB 690:GOSUB 960
420 IF POINT (202, 34)<>2 THEN 570
430 GOSUB 820
440 FOR M=1 TO K-1
450 IF I=N(M) OR I=M(M) THEN I=0
460 NEXT M
470 IF I<>0 THEN 490
480 I=INT(RND(1)*4)+2:GOTO 430
490 GOSUB 880:GOSUB 960
500 IF POINT (202, 34)=2 THEN 410 ELSE GOSUB 670
510 GOTO 590''''''''''
520 "''''''''''НАЧИНАЕТ КОМПЬЮТЕР''''''''''
530 I=1:GOSUB 880
540 GOSUB 690:K=K+1:GOSUB 820:GOSUB 880:GOSUB 960
550 IF POINT(202, 34)=2 THEN 540 ELSE GOSUB 670
560 GOTO 590''''''''''
570 GOSUB 1040
580 PRINT#1, "ВЫ ВЫИГРАЛИ";S; "-Ю ПАРТИЮ":L=L+1
590 GOSUB 1050:GOSUB 1050:NEXT S
600 GOSUB 1040:PRINT#1, "ОБЩИЙ СЧЕТ: ЧЕЛ.-";L; " КОМП.-";2*В-L
610 IF L<>В THEN 630
620 PRESET (0, 190):PRINT#1, "МОЛОДЕЦ!"
630 PRESET (0, 200):PRINT#1, "БУДЕТЕ ИГРАТЬ ЕЩЕ (Д/Н)?"
640 А$=INKEY$:IF А$="" THEN 630
650 IF А$="D"ORА$="d"ORА$="Д"ORА$="д" THEN 70 ELSE END
660 "''''''''''ПОДПРОГРАММЫ''''''''''
670 GOSUB 1040:PRINT#1, "Вы проиграли";S; "-Ю ПАРТИЮ"
680 RETURN''''''''''
690 С=0:Н=1''''''''''ХОД ЧЕЛОВЕКА''''''''''
700 GOSUB 1040:PRINT #1, "Введите свой";К; "-й ход";
710 А$=INKEY$:IF А$="" THEN 710
720 В$=INKEY$:IF В$="" THEN 720

```

```

730 I=VAL(A$+B$):GOSUB 1040
740 PRINT #1," БАМ";K;"-Я ход";I:N(K)=I
750 GOSUB 1050:T=IMOD9
760 IF T=6 OR T=7 OR T=8 OR T=0 THEN 790
770 X3(K)=X(I)/5+197;Y3(K)=Y(I)/5;X4(K)=X(I)/5+203;Y4(K)=Y3(K)
780 LINE(X(I)-17,Y(I))-(X(I)+17,Y(I)),2:GOTO 810
790 X3(K)=X(I)/5+200;Y3(K)=Y(I)/5-3;X4(K)=X3(K);Y4(K)=Y(I)/5+3
800 LINE(X(I),Y(I)-17)-(X(I),Y(I)+17),2
810 RETURN
820 C=1;H=0"*****ХОД КОМПЬЮТЕРА*****"
830 M=I MOD 10
840 IF M=0 THEN I=I+1;GOTO 880
850 IF M=1 THEN I=I-1;GOTO 880
860 IF M<6 THEN I=I+4
870 IF M>=6 THEN I=I-4;RETURN
880 GOSUB 1040;PRINT #1,"Моя";K;"-Я ход";I:M(K)=I
890 GOSUB 1050:T=IMOD9
900 IF T=6 OR T=7 OR T=8 OR T=0 THEN 930
910 X1(K)=X(I)/5+200;Y1(K)=Y(I)/5-3;X2(K)=X1(K);Y2(K)=Y(I)/5+3
920 LINE(X(I),Y(I)-17)-(X(I),Y(I)+17);GOTO 950
930 X1(K)=X(I)/5+197;Y1(K)=Y(I)/5;X2(K)=X(I)/5+203;Y2(K)=Y1(K)
940 LINE(X(I)-17,Y(I))-(X(I)+17,Y(I))
950 RETURN
960 "*****КОНТРОЛЬ КОНЦА ПАРТИИ*****"
970 LINE(200,0)-(236,36),4,BF
980 LINE(200,0)-(236,36),2,B
990 FOR J=1 TO K
1000 X1=X1(J)XC+X3(J)XН;Y1=Y1(J)XC+Y3(J)XН
1010 X2=X2(J)XC+X4(J)XН;Y2=Y2(J)XC+Y4(J)XН
1020 LINE(X1,Y1)-(X2,Y2),2;NEXT
1030 PAINT(235,1),2;RETURN"*****"
1040 LINE(0,180)-(255,191),4,BF;PRESET(0,180);RETURN
1050 FOR T=1 TO 500;NEXT;RETURN
1060 DATA 20,20,55,20,90,20,125,20,160,20,35,37,70,37,105
1070 DATA 37,140,37,20,55,55,55,90,55,125,55,160,55,35,72,70
1080 DATA 72,105,72,140,72,20,90,55,90,90,90,125,90,160,90
1090 DATA 35,107,70,107,105,107,140,107,20,125,55,125,90,125
1100 DATA 125,125,160,125,35,142,70,142,105,142
1110 DATA 140,142,20,160,55,160,90,160,125,160,160,160

```

## Литература

1. Шмухлер П. Перекинь мостик // Наука и жизнь. 1988. № 3.
2. Располов В. Б., Кузнецов С. Т. Комп'ютерні ігри. Сер. 7. «Педагогічна», № 11. К.: Т-во «Знання» УРСР, 1990.
3. Очков В. Ф., Хмедюк В. А. От микрокалькулятора к персональному компьютеру. М.: Изд-во МЭИ, 1990.

А. А. Саяпин,  
г. Москва

## СЕТЕВАЯ СИСТЕМА SPRUT2

### Краткое описание

#### Назначение и условия применения

Сетевая система SPRUT2 разработана с целью максимального использования широких возможностей локальной сети 75 Кбод на базе компьютеров семейства БК. Система обеспечивает выполнение на периферийных компьютерах БК-0011М программ для БК-0011М (расширение .SAV) и программ для БК-0010 в машинных кодах (.BIN), на языках FOCAL (.FOC) и BASIC (.ASK, .COD). Программы могут читать и писать файлы данных на диски центральной ПЭВМ.

Система работоспособна при наличии сетевого оборудования и с операционной системой V4.0 и старше. Операционная система может находиться в любом режиме (предпочтительнее виртуальный). Файлы сетевой системы SPRUT2 должны располагаться на системном диске.

#### Запуск системы

Если операционная система находится в стандартном режиме, рекомендуется перейти в виртуальный командой .Setswp Memory.

Перед запуском системы могут быть заданы дополнительные однобуквенные имена устройств, доступных периферийным машинам в процессе работы. Эти устройства задаются командой операционной системы .Assign. В поставляемых ИНТЕРСЕРВЕРом операционных системах в стартовом командном файле задаются следующие имена устройств:

.Assign BY0 A .Assign BY1 B .Assign SY C .Assign LD7 D

Запуск системы осуществляется командой SPRUT2. После этого периферийные ПЭВМ можно включать в сеть двумя способами: либо включением питания, либо перезапуском ПЭВМ путем нажатия клавиш [AP2]/[ЛАТ],[AP2]/[СБР]. В обоих случаях вхождение в сеть происходит автоматически.

#### Работа в системе

Пользовательский интерфейс системы выполнен по принципу двухпанельного управления в стиле Norton Commander для IBM PC. Панели содержат каталоги выбранных устройств в центральной ПЭВМ. Левая панель носит название системной, каталог в ней выводится в порядке расположения файлов на устройстве. Правая панель называется рабочей, ее каталог отсортирован по алфавиту.

В нижней строке экрана отображаются функциональные клавиши. Их ввод осуществляется одновременным нажатием [AP2] и соответствующей цифровой клавиши. Приведем их список и назначение:

- 1 (Help) — справочник;
- 2 (Left) — переход в левую панель и чтение каталога;
- 3 (Right) — переход в правую панель и чтение каталога;
- 5 (BK0010) — вход в режим эмуляции БК-0010;
- 0 (Quit) — выход из системы.

Для запуска программ на периферийной ПЭВМ нужно подвести курсор на требуемый файл и нажать [ВВОД]. Переход между панелями осуществляется клавишей [ТАБ]. Файлы данных для исполняемых программ записываются на устройство, каталог которого приведен в рабочей панели.

Следует отметить, что таким образом запускаются программы с расширением .SAV и .BIN (т. е. программы в кодах для БК-0011М и БК-0010). На файлы с другими расширениями система не реагирует. Для исполнения программ для БК-0010 на языках FOCAL и BASIC служит функциональная клавиша [5]. Нажимая одновременно [AP2]/[B], можно выбрать текущий режим БК-0010 — FOCAL или BASIC (при этом меняется текст в нижней строке экрана). Войдя в этот режим, можно работать с языками FOCAL или BASIC как на отдельной ПЭВМ БК-0010.

Функциональные клавиши [2] и [3] предназначены для чтения каталогов других устройств в соответствующие панели. При их нажатии выдается запрос «... Drive letter [ABCD.]». Следует ввести одну из букв в квадратных скобках. Точка соответствует рабочему диску центральной ПЭВМ. Другие имена устройств задаются командой .Assign операционной системы.

Функциональная клавиша [1] предназначена для получения справочника по сетевой системе SPRUT2. Выйти из SPRUT2 в операционную систему можно клавишами [AP2]/[0].

#### Сообщения системы SPRUT2

- Hardware error — ошибка сетевой аппаратуры;
- Insufficient memory — недостаточное памяти;
- ROM error — ошибка ПЗУ в странице #12;
- No module — на системном диске нет модулей системы или ни один не считан.

С. А. Разбитной,

г. Москва

## ГЕОМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ НА БК

Существует, по крайней мере, три типа программ, основная функция которых — построение изображения на экране дисплея. К первому типу можно отнести графические редакторы — программы, позволяющие сформировать картинку на экране и сохранить ее, как правило, в растровой форме. Ко второму типу относятся плоскостные системы геометрического моделирования. Они отличаются тем, что в памяти машины хранится не изображение на экране, а параметры геометрических объектов — точек, прямых, окружностей и т. д. Параметры этих объектов соответствуют реальным координатам, а на экран выводится полное изображение или его фрагменты в произвольном масштабе. И наконец, третий тип — пространственные системы геометрического моделирования. В памяти хранятся трехмерные координаты, а изображение выводится на экран в произвольном виде и масштабе.

Системы геометрического моделирования (СГМ) используются во всем мире уже более 20 лет и составляют основу пакетов автоматизации конструирования и производства. (Простейший пример — выпуск чертежей.) Ресурсы памяти и быстродействия, необходимые для работы такой системы, значительно превышают возможности бытового компьютера. Тем не менее, автором разработана плоскостная система геометрического моделирования для БК-0010, в которой реализованы основные функции, используемые в современных СГМ.

Эта программа предназначена для точных геометрических построений на плоскости. Создаваемая геометрическая модель может состоять из объектов трех типов: точек, отрезков прямых и дуг окружностей. Для точки в памяти машины хранятся ее координаты, для отрезка — координаты его концов, для дуги — координаты центра, радиус и углы, соответствующие начальной и конечной точкам. Координаты измеряются в любых единицах.

Работа с программой сводится к построению точек, прямых и окружностей. Для задания точки можно указать ее позицию на экране, ввести координаты, смещение от существующей точки по координатным осям или вдоль прямой. Можно построить точку на конце дуги или отрезка, в центре окружности, а также использовать точку пересечения двух линий.

Для построения отрезка прямой можно задать координаты его конца, указав их на экране или введя числовые значения. Можно также построить отрезок под заданным углом к оси X, параллельно или перпендикулярно другому отрезку, построить касательную к окружности или к двум окружностям.

Окружность задается положением центра и длиной радиуса либо строится по трем точкам. Реализованные в системе функции обеспечивают любые построения, которые могут быть выполнены с использованием линейки и циркуля.

Для управления программы используется система меню. Меню, представляющее собой список функций, выводится в нижней части экрана под графическим изображением. Функция выбирается нажатием соответствующей цифровой клавиши.

Изображение выводится на экран в произвольном масштабе. Программа позволяет увеличить или уменьшить текущий масштаб вдвое с указанием нового центра рисунка, выделить на экране прямоугольную область, указав границы курсором, и вывести содержащееся в ней изображение во весь экран, а также рассчитать максимальный масштаб, в котором рисунок полностью помещается на экране.

На созданном изображении программа позволяет определять координаты, измерять расстояние и угол, вычислять площадь треугольника, заданного координатами вершин.

Любой объект, который в данный момент не требуется для работы, может быть временно погашен, а потом, если необхо-

димо, вновь отображен на экране. Линии промежуточных построений могут быть удалены.

С каждым объектом связан параметр, называемый номером уровня. Все объекты, имеющие один и тот же номер уровня, образуют группу, которая может быть вызвана по этому номеру.

Программа написана на языке Фокал и занимает почти всю память БК-0010. Свободного пространства хватает для размещения 30—40 объектов, что позволяет использовать программу, как правило, только в учебных целях. Та же программа, переписанная на БЕЙСИКе, к сожалению, не помещается в памяти БК-0010, но ее реализация на Спектруме с памятью 48К позволяет создать около 80 графических объектов. Объем исходного текста программы на Бейсике для IBM PC составляет 60—80К, в зависимости от включаемых в нее модулей. В отличие от версии для БК, это уже «полномасштабная» программа, предназначенная для выпуска черте-

жей. Она является составной частью программного комплекса ГЕМОС (геометрическое моделирование обводов самолета) для IBM-совместимых ПЭВМ, разработанного специалистами ОКБ им. А. С. Яковлева и использовавшегося при проектировании и производстве самолетов Як-54, Як-112, Як-58 и др.

Характеристики данной программы и опыт работы с ней показывают, что на БК-0011 возможна реализация рабочей версии плоскостной системы геометрического моделирования. Несмотря на слабые возможности этой машины по сравнению с IBM PC, ее можно будет использовать для конструкторских работ, а возможность файловой совместимости БК с IBM позволит выводить чертежи на графопостроитель.

*Желающие приобрести систему ГЕМОС и ознакомиться с ее демоверсией могут обращаться в редакцию по телефону: (095) 208-30-78.*

\* \* \*

### Уважаемые читатели!

В письмах, приходящих в редакцию, пользователи БК задают множество вопросов. Мы попытались выделить наиболее интересующие наших читателей темы:

- организация буферизованного доступа к файлам в ОС ANDOS (операции типа «открытие/закрытие файла», «ввод/вывод байта», «перемещение указателя текущей позиции» для чтения/записи байта в произвольном месте файла — прямой доступ — и т. д.) и обеспечение возможности работы одновременно с несколькими файлами произвольной длины с добавлением информации к уже существующему файлу;
- звуковые эффекты (в том числе синтез речи и многоголосные мелодии);
- распознавание спектра звукового сигнала, поступающего с магнитофона (в том числе с подключенного к нему микрофона), и выделение основной частоты;
- реализация систем печати текста в графическом режиме с загружаемым по выбору пользователя драйвером принтера и подгружаемыми в процессе работы шрифтами (в том числе разрабатываемыми пользователем);
- самодельные периферийные устройства («световое перо», дигитайзер, чувствительный к давлению планшет, сканер и др.);
- спрайтовая графика и видеоэффекты;
- листинги игр;
- реализация языков программирования (Паскаль, Си).

Приглашаем всех, кто располагает какой-либо информацией по перечисленным выше темам, поделиться своими знаниями и опытом через наш журнал. Список тем в дальнейшем будет дополнен по вашим предложениям.

# EYELIBRIS



Вышла в свет книга: Котов Ю. В., Павлова А. А. Основы машинной графики. М.: Просвещение, 1993 (тираж 7000 экз., 16 печ. л.). Первоначально она была задумана как учебное пособие для студентов педагогических вузов, однако ее окончательный вариант более предназначен для самостоятельного освоения машинной графики. И потому эта книга окажется полезной многим — школьным учителям и для занятий в кружках и факультативах, а также индивидуальным пользователям домашних компьютеров. Приводятся примеры программ на распространенных версиях языка Бейсик для популярных ПЭВМ типа БК-0010, БК-0011, «Ямаха». Листинги легко могут быть перенесены на «Корвет», УК НЦ или IBM-совместимые компьютеры.

В первых главах книги дается обзор технического обеспечения и программных средств машинной графики, поясняются принципы действия распространенных типов графических устройств, описаны типовые приемы работы с диалоговыми графическими программами. Далее следует первичный практикум использования графических операторов Бейсика с примерами программ и иллюстрациями. Эти материалы могут использоваться как дополнение к школьному курсу информатики или для самостоятельного изучения. В последующих главах делается попытка разработки вместе с читателями небольшой программной системы двумерной, а затем трехмерной графики. Приводятся тексты типовых графических подпрограмм, раскрываются «секреты» построения аксонометрических и перспективных проекций, координатных преобразований, даются алгоритмы решения распространенных задач геометриче-

ского конструирования и некоторых операций растровой графики.

К сожалению, в текст книги, в том числе и в листинги программ, кое-где вкрались опечатки. Вот некоторые из них:

- стр. 130, строка 25 — пропущена десятичная точка (правильная запись: .01745);
- стр. 142, десятая строка снизу — следует читать:  $W3=BT*WU$ ;
- стр. 143, строка 14 — ...GOTO 5730;
- стр. 145, в последней строке —  $R=CG* \text{COS}(V)$ ;
- стр. 148, строка 23 —  $J4=J3+J2-J1$ .

Кроме того, в строке 19 на стр. 158 перепутаны индексы, а на стр. 218 (седьмая строка снизу) пропущена часть формулы.

Полный перечень замеченных ошибок и дополнительных замечаний вы можете бесплатно получить в редакции журнала лично или по почте (в последнем случае необходимо прислать вместе с заявкой неклеенный конверт с маркой и надписанным обратным адресом). В редакции же можно приобрести описываемые в книге программы, в том числе программную систему Алграф, ссылки на которую приведены в приложении, на магнитном носителе (кассете или дискете).

Несмотря на отмеченные недостатки, авторы и редакция предполагают значительный интерес к этой публикации. В издательстве рассматривается возможность второго издания книги «Основы машинной графики», исправленного и дополненного. Тираж и сроки будут определяться по вашим заявкам на ее приобретение, присланным в адрес редакции.



**Движение «МОЛОДОСТЬ РОССИИ»  
и АССОЦИАЦИЯ СУПЕРКЛУБОВ РОССИИ «СУПЕРНИКО»  
проводят Первый Всероссийский фестиваль**

## **«МОЛОДОСТЬ РОССИИ»**

*Москва, 5-9 января 1995 г.*

Инициатором проведения фестиваля является правительство России, общественные и благотворительные организации.

Издательство «Информатика и образование» (член оргкомитета по конкурсной теме «Компьютерные программы») приглашает всех желающих принять участие в конкурсе программистов. По тематике (от игр до системных программ), типу компьютера, языку программирования, месту жительства (Россия, СНГ, дальнее зарубежье), по возрасту — никаких ограничений не предусмотрено.

Авторы пяти лучших разработок будут награждены профессиональными компьютерами IBM, для занявших последующие десять призовых мест предусмотрены ценные подарки.

Для участия в конкурсе необходимо предоставить следующие материалы:

- законченная программная разработка с ее подробным описанием (последнее — на магнитном носителе и в виде распечатки);
- девиз работы (оценка анонимная);
- подробные сведения об авторе (паспортные данные, почтовый адрес с индексом, телефон) и соавторах;
- сопроводительный лист (краткая аннотация разработки, точное указание типа компьютера, операционной системы, языка программирования и формата записи на магнитном носителе, перечень предоставленных файлов с их назначением).

Кроме того, желательно предоставить также исходный текст на языке программирования (в файле и в виде распечатки), а также демонстрационный файл для IBM (если разработка реализована на компьютере другого типа).

**СРОК ПОДАЧИ МАТЕРИАЛОВ — НЕ ПОЗДНЕЕ 10 НОЯБРЯ 1994 г.**

Материалы на конкурс (дискеты и распечатки) необходимо доставить лично или по почте (можно также переслать файлы по электронной почте).

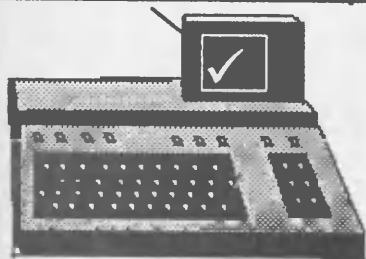
<b>Телефон для справок:</b>	<i>(095) 151-19-40</i>
<b>Адрес:</b>	<i>103051, Россия, Москва, ул. Садовая Сухаревская, д. 16, комн. 9. Васильеву Борису Михайловичу</i>
<b>E-MAIL:</b>	<i>mail@infoobr.msk.su</i>
<b>Факс:</b>	<i>(095) 208-67-37</i>

**ОРГКОМИТЕТ КОНКУРСНОЙ ТЕМЫ  
«КОМПЬЮТЕРНЫЕ ПРОГРАММЫ»**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
**ЦЕНТР ИНТЕРАКТИВНЫХ СРЕДСТВ ОБУЧЕНИЯ**

*Мини-винчестер и быстрая сеть для УКНЦ!*

Мини-винчестер!



*Мини - винчестер*

- Размером как контроллер НГМД
- Высокая надежность
- Не требует использования дискет
- Полная программная совместимость

*Быстрая сеть*

- Подготовка к уроку за полминуты
- Полная защита данных
- Ускорение работы в 50 - 60 раз
- Взаимонезависимая работа учеников

Возможна бесплатная установка ПМК



**УКНЦ**



*базовый курс информатики*

**Цены снижены с 1.08.94 по 31.12.94**

Цены (включая НДС)	старые	новые
комплекс "IBM на УКНЦ" .....		\$35
мини-винчестер 20 Мбайт .....	\$250	\$210
мини-винчестер 30 Мбайт .....		\$270
мини-винчестер 40 Мбайт .....		\$315
быстрая сеть "NET128" .....	\$510	\$375

Каждый первый понедельник каждого месяца - бесплатные семинары

**РЕКВИЗИТЫ ЦЕНТРА**

Адрес: 125315, Москва, ул. Часовая, 21-б.

Банковские реквизиты для Москвы и Московской обл.: Расчетный 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 37, (095) 155 87 30.

Телефакс: (095) 155 87 27.

E-mail: rcime@aha.msk.su

# КЛУБ УКНЦ

---

**В. А. Брусенцев,**

*г. Москва*

## НОВАЯ ЖИЗНЬ УКНЦ

Стандартная конфигурация класса УКНЦ накладывает ряд ограничений на использование многих учебных программ. Они связаны в основном с двумя факторами. Первый из них — использование накопителя на гибких магнитных дисках (НГМД), который обладает малой емкостью и является источником ошибок чтения/записи.

В последнее время стало возможным комплектовать класс УКНЦ винчестером — малогабаритным накопителем на жестком магнитном диске. Винчестер (НЖМД) подключается к ЭВМ МС 0511 (УКНЦ) с помощью стандартного по размерам адаптера, который вставляется во второй слот (верхний разъем на клавиатуре). Емкость НЖМД может быть от 10 до 72 Мб. Это пространство можно разделить на несколько (до восьми) логических устройств, на каждом из которых может быть размещена операционная система и (или) другие программы. Использование винчестера избавляет от постоянной переустановки и перезагрузки дискет и устраняет системные сбои по вине НГМД. Высокая надежность и быстродействие винчестера превращают УКНЦ в персональную ЭВМ, сравнимую по производительности с IBM PC/XT.

Вторым фактором, ограничивающим возможности УКНЦ для работы с учебными программами, широко использующими графику и мультипликацию (такими как

программно-методический комплекс «IBM на УКНЦ», — см. «Информатика и образование». 1993. № 6), становится сетевое обеспечение, которое при проектировании класса было ориентировано исключительно на рассылку специализированных программ (сетевой BASIC, игры и т. п.). В дальнейшем оно было доработано, но низкая скорость передачи данных в канале ЛВС (57,6 кБод) и некоторые другие особенности аппаратуры сетевого обеспечения не дают возможности эффективно использовать класс УКНЦ даже при наличии винчестера. Скорость работы класса с винчестером возрастает по сравнению с использованием НГМД всего в 1,5 — 2 раза (см. таблицу). При этом время подготовки класса к уроку превышает 20 минут, что весьма обременительно для учителя.

В Центре интерактивных средств обучения Министерства образования РФ проверены и предлагаются к продаже три новые разработки, позволяющие повысить быстродействие и надежность класса УКНЦ.

**Использование нового программного сетевого средства BiNet v.1.5 без модернизации класса.** Система виртуального доступа к винчестеру BiNet v.1.5 использует новый протокол обмена в ЛВС, работает практически без ошибок и дает экономию времени еще в 1,5 — 2 раза. Интерфейс учителя и ученика делает работу наглядной и удобной.

На мониторе учителя отображаются:

- рабочий каталог и перечень доступных устройств;
- выбранный для загрузки слепок операционной системы (копия памяти ЭВМ, записанная в файл);
- программа, выбранная для одновременной загрузки на все рабочие места учеников;
- номера машин учеников, с которыми производится обмен;
- процессы загрузки слепков операционной системы и программ;
- текущий и доступные режимы работы.

На мониторах учеников отображаются:

- каталог выбранного в данный момент устройства;
- перечень доступных устройств;
- режимы работы;
- содержание электронного квазидиска, сформированного в памяти периферийного процессора.

В новой сети предоставляются возможности:

- генерации слепков операционной системы;
- одновременной загрузки программ на все рабочие места учеников;
- независимой работы каждого рабочего места.

**Замена сетевых адаптеров локальной вычислительной сети класса на новые (CA 02) и использование нового программного обеспечения.** Новые сетевые адаптеры вставляются на место старых и используются без переделки кабельную систему кольцевой сети КУВТ МС 0202. Скорость передачи данных в канале ЛВС равна 1 Мб в секунду, поэтому реальная скорость работы класса при использовании винчестера и новой сети возрастает в 45—50 раз по сравнению со старой локальной вычислительной сетью и НГМД (см. таблицу).

Интерфейс и возможности новой сети соответствуют указанным в описании BiNet v. 1.5.

**Программно-аппаратный комплекс NET128** состоит из новых сетевых адаптеров (NEW), соединенных параллельно между собой витой парой проводов и включаемых в первый слот (верхний разъем) на клавиатуре УКНЦ. Одновременно сохраняется возможность использования класса со старой ЛВС. При работе с винчестером скорость возрастает в 45 — 50 раз по сравнению со старой локальной сетью и НГМД (см. таблицу).

Комплекс NET128 предоставляет учителю следующие возможности:

1. Винчестер может быть разделен на устройства (от 1 до 8), содержащие операционную систему, общедоступные программы и (или) файлы отдельных учащихся.

2. Раздел, содержащий общие программы, недоступен с рабочих мест по записи, что исключает их умышленную или случайную порчу.

3. Каждый индивидуальный раздел доступен по записи только с одного рабочего места, что исключает порчу файлов одного ученика другими.

4. Индивидуальные разделы недоступны по чтению со всех рабочих мест, кроме одного, что исключает списывание данных одного ученика другими.

5. Индивидуальные разделы могут быть сделаны доступными по чтению с учительского рабочего места во время урока в произвольном порядке, что позволяет контролировать выполнение заданий отдельными учениками или выводить на печать результаты.

6. При смене группы учащихся в классе замена индивидуальных разделов производится с помощью одной команды на сервере.

7. Индивидуальные разделы, неактивные на данном уроке, могут быть сделаны

доступными по чтению с рабочего места учителя, что дает ему возможность проверять выполнение заданий в других классах во время текущего урока.

8. Запуск программ на ученических рабочих местах может выполняться автоматически (из стартового командного файла) или набором на клавиатуре (либо указанием в строке меню файлового монитора) имени программы учеником, что полностью освобождает учителя от манипулирования процессом загрузки программ в течение урока.

9. Учитель или несколько учителей могут иметь свои индивидуальные разделы, один из которых доступен по записи в течение урока, что дает возможность подготавливать

ливать файлы документов и выводить их на печать, создавать командные файлы или программы, в то время как ученики работают со своими разделами, выполняя задание.

10. Интерфейс сервера отображает процессы обмена в классе и успешность их завершения.

11. Файловый монитор на местах учеников отображает каталоги доступных устройств и доступные режимы работы.

**Программно-аппаратный комплекс NET128** организует в классе УКНЦ сетевую систему с полной защитой данных учителя и учеников. Работа РМУ под управлением этого комплекса в режиме реального времени доступа к винчестеру полностью решает все ранее перечисленные проблемы.

Таблица сравнительных характеристик ЛВС УКНЦ

Тип накопителя	Тип адаптера	Тип программно-аппаратного комплекса	Время, необходимое для			Стоимость комплекса ЛВС в долларах без учета стоимости НЖМД
			перезагрузки РМП	одновременной загрузки операционной системы на 12 РМУ (RT11 v. 5.4 размером 78 блоков с заставкой)	одновременной загрузки учебной компьютерной среды на 12 РМУ («Кенгуренок-чертежник» размером 169 блоков)	
НГМД	типовой	NET SVD	1,5 мин	11 мин	20 мин (1 сбой — задержка на 5 мин)	35
НЖМД	типовой	NET SVD	25 с	8 мин	18 мин (1 сбой — задержка на 5 мин)	35
НЖМД	типовой	BiNet v.1.5	13 с	45 с	8 мин (БЕЗ СБОЕВ)	80
НЖМД	CA 02	BiNet v.2.2	13 с	30 с	1,0 мин (БЕЗ СБОЕВ)	510
НЖМД	NEW	NET 128	13 с	1,5 мин	2 мин (БЕЗ СБОЕВ)	510

Телефоны Центра интерактивных средств обучения: 155-87-30, 155-87-37.

Факс: 155-87-27.

# МЕДИАТЕКА

**Уважаемые читатели — учителя информатики, методисты!**

Издательство «Информатика и образование» в октябре 1994 г. выпускает брошюру Е. Н. Ястребцевой «Медиатека». Автор занимается этой проблемой несколько лет, работая в Российской Академии образования. На экспериментальных площадках (в школах различных регионов) уже накоплен опыт создания медиатек по методике, описанной в брошюре. Ниже приводится ее содержание.

Актуальность создания медиатеки

Назначение медиатеки

Деятельность учителя

Деятельность учащегося

Деятельность сотрудника медиатеки

Требования к материально-технической базе медиатеки

Состав информационных и технических средств

Мебель и приспособления

Помещения и их планировка

Финансирование медиатеки

Как создать медиатеку

*Приложение 1. Примерный перечень оборудования и мебели медиатеки*

*Приложение 2. Положение о школьной медиатеке (разработано*

*совместно с Н. М. Стадником)*

I. Общие положения

II. Задачи школьной медиатеки

III. Направления деятельности школьной медиатеки

IV. Финансовая и хозяйственная деятельность школьной медиатеки

V. Структура и штаты

VI. Должностные обязанности

VII. Права и обязанности сотрудников медиатеки

**«МЕДИАТЕКА» РЕКОМЕНДОВАНА К ИЗДАНИЮ ГЛАВНЫМ УПРАВЛЕНИЕМ  
РАЗВИТИЯ ОБЩЕГО СРЕДНЕГО ОБРАЗОВАНИЯ  
МИНИСТЕРСТВА ОБРАЗОВАНИЯ РОССИИ**

Брошюра может поставляться отдельно или вместе с дискетой, на которой записана программа «Каталог». Программа защищена от копирования.

При покупке от 1 до 50 экземпляров:

цена брошюры (с пересылкой) 9 000 руб.

цена брошюры с программой (с пересылкой) 35 000 руб.

При покупке от 50 до 100 экземпляров скидка 2%.

При покупке от 100 до 500 экземпляров скидка 7%.

При покупке свыше 500 экз. скидка 10%.

Оплата может быть произведена как по безналичному расчету, так и за наличный расчет.

Для справок:

телефон 208-30-78, факс 208-67-37, E-Mail info@tit-bit.msk.su.

Если вы решили приобрести брошюру «Медиатека» по безналичному расчету, заполните счет и перечислите сумму вашей покупки на наш расчетный счет.

ГП «Редакция журнала «Информатика и образование». Адрес: 103051, Москва, Садовая Сухаревская, 16. Телефон: 208-30-78. Факс: 208-67-37.  
 Расчетный счет для Москвы и Московской области: 609602 в ММКБ филиал «Интеллект», МФО 212199, уч.1Е,  
 Расчетный счет для России и СНГ: 609602 в ММКБ филиал «Интеллект», корп.счет 216161800 в ЦРКЦ ГУ ЦБ РФ, МФО 211004, уч.1Е.

### СЧЕТ

Плательщик и адрес

Предмет счета	Количество, шт.	Цена, руб.	Скидка	Сумма, руб.
Брошюра «Медиатека»		9 000		
Брошюра «Медиатека» с программой		35 000		
Итого к оплате (с учетом НДС и спецналога)				
Сумма прописью:				

Директор

А.Ю.Кравцова

Гл.бухгалтер

З.В.Бородаева



После этого заполните бланк-заказ и вышлите его по адресу: 103051, Москва, Садовая Сухаревская, 16, комн.9.

### БЛАНК-ЗАКАЗ

Мы хотели бы приобрести следующую литературу:

Наименование издания	Количество экземпляров	Стоимость одного экземпляра, руб.	Общая сумма покупки (со скидкой)
Брошюра «Медиатека»		9 000	
Брошюра «Медиатека» с программой		35 000	

Наш адрес:

Телефон:

E-mail:

Номер платежного поручения

Сумма:

За наличный расчет брошюру и программу можно приобрести в редакции.

## БЕЙСИК КУВТ-86: РАБОТА С КЛАВИАТУРОЙ

Прочитав в третьем номере журнала ИНФО за 1991 г. статью А. Ланеева «Кирпичики ваших программ», я опробовал одну из программ на БК-0010Ш, входящем в КУВТ-86. Предложенная в качестве иллюстрации работы с клавиатурой вторая программа на самом деле рассчитана на джойстик. Чтобы все-таки получить возможность общения с клавиатурой, я проанализировал эту программу и внес изменения. Она стала следующей:

```
10 X=122
20 Y=122
30 CLS
40 LINE (X,Y)-(X+6,Y+6),1,B
50 IF PEEK(&O177716)=-32608 THEN
    P=3 ELSE P=0
60 IF PEEK(&O177662)=8 THEN R=-P
70 IF PEEK(&O177662)=26 THEN U=-P
80 IF PEEK(&O177662)=27 THEN U=P
90 IF PEEK(&O177662)=25 THEN R=P
100 IF P=0 THEN 50
110 LINE (X,Y)-(X+6,Y+6),4,B
120 Y=Y+U
130 X=X+R
140 IF X THEN X=240
150 IF X240 THEN X=10
160 IF Y THEN Y=230
170 IF Y230 THEN Y=10
180 LINE (X,Y)-(X+6,Y+6),1,B
190 R=0
200 U=0
210 GOTO 50
```

*Ученик X класса СШ № 4 г. Грязи Липецкой области*

**От редакции**

Данная программа иллюстрирует возможность работы непосредственно с регистром данных клавиатуры. Строка 50 проверяет, удерживается ли в данный момент какая-либо из клавиш нажатой, а строки 60—90 «выясняют», какая это из клавиш-«стрелок» (и «стрелка» ли это вообще). Оставшаяся часть программы обеспечивает вывод прямоугольника, смещая его по экрану в соответствии с нажатой пользователем клавишей. В программе используются стандартные для вильнюсского Бейсика операторы, и она может работать как на БК-0010.01, так и на БК-0010Ш.



## ВСЕМ! ВСЕМ! ВСЕМ!

Популярный учебник по информатике В.Перепелкина «Персональный компьютер в школе»

Начинающий без проблем освоит азы программирования, бывалый упорядочит свои знания, учитель ознакомится с оригинальной методикой преподавания предмета ОИВТ.

Стоимость книги 0,7 \$ USA без учета почтовых расходов. Оплата в рублях по текущему курсу. По России наложенный платеж.

346839, Ростовская область, Неклиновский р-н, п.Новоприморский.  
«Компьютер – Сервис»

## ДОСКА ОБЪЯВЛЕНИЙ

**БАЛАПАН** – 20 программ для «Корвета» и учебные планы (1 – 4 класс).  
Тел. 145-75-05.

Программы для компьютеров БК-0010.01, 11М (на кассетах и дискетах 5,25); ZX Sinclair, Spektrum и совместимых с ним (на кассетах); Atari XL/XE (на кассетах и дискетах 5,25); Atari ST (на дискетах 3,5); Commodore 64/128 (на кассетах и дискетах 5,25); Amiga (на дискетах 3,5); IBM PC и совместимых с ним (на дискетах 5,25 и 3,5) по адресу: 127349, Москва, а/я 9.

Тел.: (095) 908-22-12 ежедневно с 10 до 21 часа. Юров В.П.



В компьютерный Центр обучения "ЛУЧ"  
125438, Москва, а/я 51, КЦО "Луч",  
Маслову А.Н.

Просим выслать:

- каталог "Книга-почтой" по информатике - \_\_ экз.
- полный каталог "Книга-почтой"
- программы подготовки преподавателей информатики и условия обучения
- каталог средств мультимедиа и другого компьютерного оборудования
- каталог обучающих и системных программ

Наш адрес: \_\_\_\_\_

Телефон/факс/Е-Mail: \_\_\_\_\_

Дата

Подпись

расписание занятий см.на обороте

Для ускорения ответа вложите  
конверт со своим адресом



**С 1 СЕНТЯБРЯ НАЧИНАЕТСЯ ПОДПИСКА НА ГАЗЕТЫ И ЖУРНАЛЫ  
НА 1 ПОЛУГОДИЕ 1995 ГОДА**

**Совт** **МАРКЕТ**

Еженедельная газета для  
пользователей, бизнесменов и  
программистов. Старейшая  
компьютерная газета на русском  
языке (выходит с 1990 г.).

**Каждую неделю  
самая оперативная информация о важнейших событиях  
в компьютерном мире России и СНГ**

- *Информация о новых программных продуктах и технических средствах.*
- *Обзорные материалы по проблемам компьютерной техники и ее применения.*
- *Аналитические материалы о состоянии российского компьютерного рынка.*
- *И многое другое... интересное и полезное.*

**Наш подписной индекс:**

32015 для индивидуальных подписчиков  
50138 для предприятий и организаций

**ВОЗМОЖНА ПОДПИСКА ЧЕРЕЗ РЕДАКЦИЮ**

**Наш адрес:**

Москва, Центр, ул. Медведева, 13, офис 4, 4-й этаж

**Адрес для писем:**

127562, Москва, а/я 120

**Телефоны:**

(095) 209-18-97, 209-49-04 (факс)

**E-mail:**

root@smarket.msk.s (via Relkom)

root@sovt.zgrad.glas.aps.org (via GlasNet)

**Начало занятий в КЦО "ЛУЧ" в 1994/95 уч. году:**

- для преподавателей информатики,  
только получивших компьютеры,  
19 сентября, 10 октября, 14 ноября,  
12 декабря 1994 г.  
и 16 января, 27 февраля, 3 апреля, 15 мая, 26  
июня, 26 июля 1995 г.
- для преподавателей информатики со стажем  
работы на IBM-совместимых компьютерах  
5 сентября, 24 октября, 28 ноября 1994 г. и  
3 января, 13 февраля, 20 марта, 17 апреля,  
12 июня, 10 июля 1995 г.;
- для преподавателей английского языка  
5 сентября 1994 г. и 27 февраля 1995 г.;
- для работников органов образования,  
руководителей учебных заведений  
4 ноября 1994 г. и 15 мая 1995 г.;
- для секретарей-референтов  
12 декабря 1994 г. и 26 июня 1995 г.;
- для бухгалтеров учебных заведений  
10 октября 1994 г. и 15 мая 1995 г.;
- для преподавателей ВУЗов и ИПК  
28 ноября 1994 г. и 30 января 1995 г.;
- по компьютерной верстке и графике  
(MS Word, Ventura, Page Maker, CorelDraw)  
24 октября и 12 декабря 1994 г. и  
17 апреля 1995 г.;
- по системам Мультимедиа  
24 октября и 12 декабря 1994 г.  
и 17 апреля 1995 г.;
- для преподавателей информатики,  
получивших Macintosh  
12 декабря 1994 г.
- по операционной системе UNIX  
24 октября и 12 декабря 1994 г.  
и 17 апреля 1995 г.

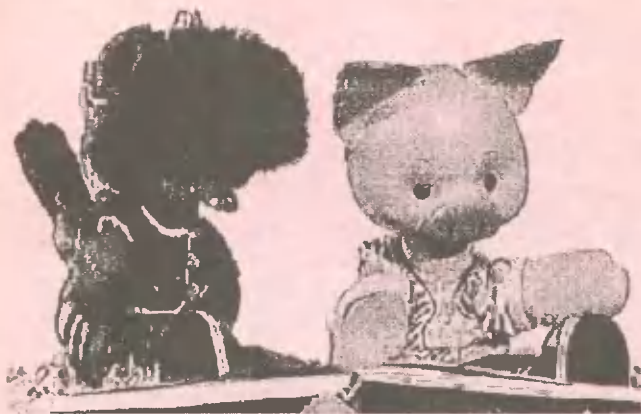
**Продолжительность каждого курса 10 дней (80 часов).**

Место проведения: Высшая школа по туризму - Москва, Кронштадтский бульвар 43а.  
(проезд от ст. м. "Водный стадион", автобусы 65 и 72 до остановки "улица Лавочкина").

**Заявки на обучение направлять по адресу:**

**125438, Москва, а/я 51, КЦО "Луч".тел: (095)-454-00-35**

**Бланк на обороте.**



- Ты решил изучить информатику?
- Да, по учебнику Кушниренко. Жаль, что нет программной поддержки...
- Как это нет! А система "КуМир"?!
- Не слышал о такой...
- Предприятие "ИнфоМир" давно ее разработало!
- Но где-же ее купить?
- Напиши в журнал "Информатика и образование"!

Мой адрес \_\_\_\_\_

Я работаю по учебнику \_\_\_\_\_ Вид ЭВМ \_\_\_\_\_

Я хочу приобрести программное обеспечение \_\_\_\_\_

Я уже приобрел программное обеспечение \_\_\_\_\_

Я хочу разработать свой гипертекстовый курс \_\_\_\_\_

Я хочу получить каталог "ИнфоМир"  Да  Нет

Я считаю, что закупки программного обеспечения должны быть централизованными  Да  Нет

Необходимо разработать для УКНЦ пакеты:

КуМир-Гипертекст  Да  Нет

ПаскальМир  Да  Нет

Планиметрия  Да  Нет

Анкеты и заявки направляйте по адресу: 103051, Москва, Садовая-Сухаревская, 16, комп.9 "Информатика и образование" Для получения каталога вложите конверт с обратным адресом

## Итоги I тиража лотереи зарегистрированных пользователей предприятия ИнфоМир

Комплект программного обеспечения со скидкой 100% 683

Комплект программного обеспечения со скидкой 50% 745 982 388 928

Комплект программного обеспечения со скидкой 25% 588 713 848 704 168 791 835 254 061 215

Все выигравшие получают систему "МикроМир" бесплатно.

У Вас должны совпасть три последние цифры регистрационного номера. Следующий тираж осенью 1994 года.

# Компьютерный центр обучения "Луч"



## Компьютерный центр обучения "Луч"

Высылает по почте книги по информатике; средства мультимедиа, модемы, дискеты, ленты для принтеров и другие компьютерные принадлежности, обучающие и системные программы.

Устанавливает компьютерные классы "под ключ".

Проводит обучение преподавателей информатики, использующих IBM-совместимые компьютеры.

Наш адрес: 125438, Москва, а/я 51  
тел.: (095) 454-00-35. Факс: (095) 454-31-66.  
E-mail: maslov@luch.msk.su

Образец заявки и расписание занятий смотрите на последнем листе журнала.