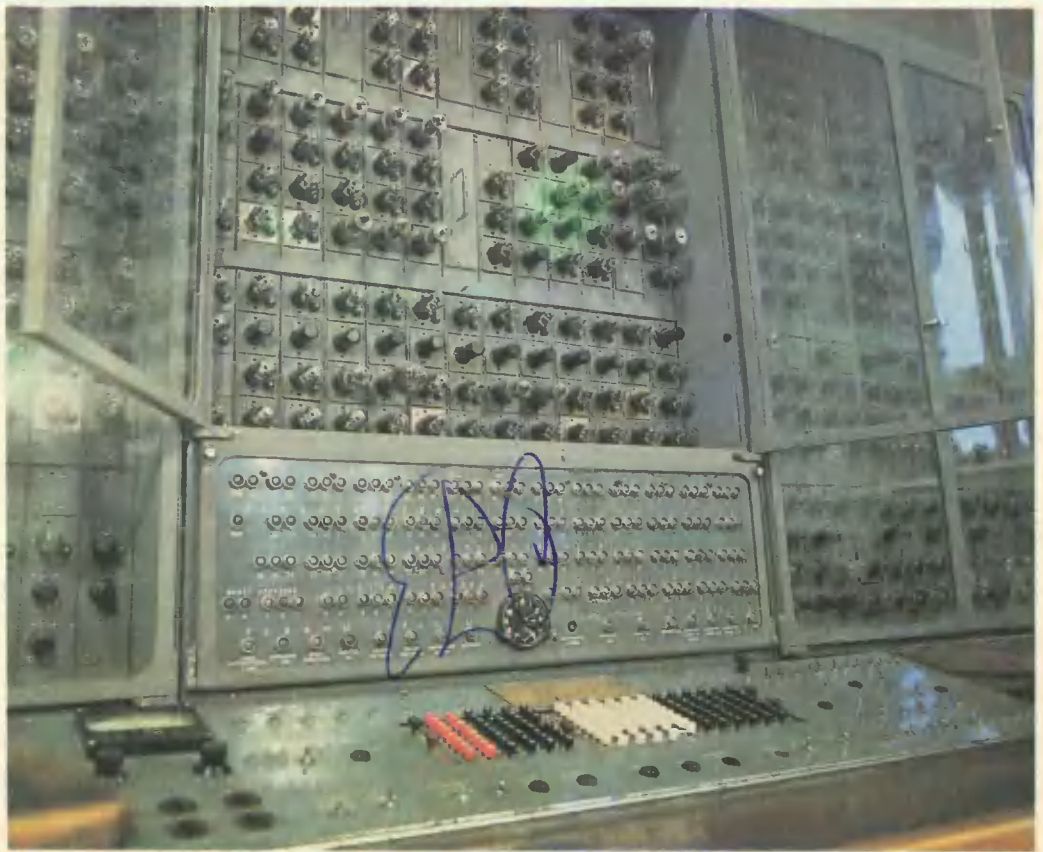


ISSN 0234—0453

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

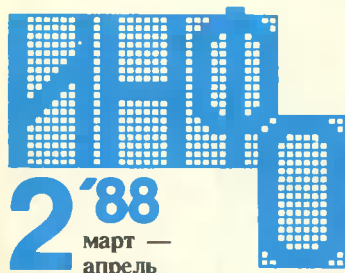
1988





Фотография на первой странице обложки многое скажет даже непосвященному. Огромный пульт, радиолампы... Музейный экспонат!

Действительно, одна из первых советских серийных ЭВМ «Урал» стоит в Московском ордена Трудового Красного Знамени Политехническом музее. Экспозиция вычислительной техники в нем невелика по площади, но предельно насыщена интереснейшими предметами. На второй странице обложки — калькулятор в записной книжке. Он сделан в 1930 году! И пусть с его помощью можно только складывать и вычитать — этот пионер миниатюризации заслуживает уважения.



ИНФОРМАТИКА И ОБРАЗОВАНИЕ

Содержание

Начало сотрудничества 3

Методика

Фукс Л. Алгоритмы поиска и сортировки 6
Смирнов Е. К концепции обучения информатике в младших классах 22

КВТ

Кривцов А. «Агат» в образовании: 1983—1987 26
Пирогов С. Технические средства диалога с персональным компьютером 35
Кузьмин Ю., Гвардина Т., Кузьмина Л. Система «Рига» 42
Тимошенко Г. Лабораторные работы на ДВК-1 49
Карпов В., Карпова О., Новичков В. Графическая информация на алфавитно-цифровом дисплее 63
Алексеев М., Алексеева Т. Реализация Е-практикума на «Искре-1256» 67
Таллинский научно-учебный центр: программное обеспечение для ПЭВМ 69
Антипов И., Колобов С., Ермолаев О., Анисимов В. О методике психологического исследования адаптации школьников к учебной нагрузке при работе на компьютере 74
Проблема ремонта: острота не снимается 75
Почему низка надежность техники? 77

Молодежная инициатива

Точка зрения

Андропов И., Гусев Е., Звягин М. Для кого написано методическое письмо? 87
Чистопольский П. «Как бы есть...» 89
Лях В. О программе курса ОИВТ: глядя из вуза 92
Рубашевский С. Новое содержание — новая форма 94
Леонас В. «Тихое бульканье» 95

Внеклассная работа

Гутман Г., Карпилова О. Азбука программирования 99
Григас Г. Олимпиада юных программистов 104

Зарубежный опыт

- Драганов М. Цели и задачи курса информатики в средних школах Болгарии 107
Асенова П. Состояние и перспективы обучения информатике в Болгарии 108

Педагогические кадры

- Психолого-педагогическая подготовка специалистов народного хозяйства для преподавания в школе 112
От школьника до профессора 119

Нам пишут

- Самодельный модем — нелепость или необходимость? 120
«Хотелось бы обсудить...» 121

Информация

- Баринев А., Семибратов А. Семинар обобщает опыт 122

Репортаж номера

- Школьник здоровье, компьютер... 124

Библиография

126

Веселый урок

Редактор отдела *Ф. Даниловский*
Редактор отдела *К. Шеховцев*
Научный редактор *Т. Драгныш*
Заведующая редакцией *Н. Игнатова*
Художественный редактор *Л. Розанова*
Корректор *Л. Яковлева*

Главный редактор
академик

В. А. МЕЛЬНИКОВ

Редакционная
коллегия

И. М. БОБКО

Б. М. ГЕРАСИМОВ

Г. В. ГОДЖЕЛЛО

А. В. ДЕНИСЕНКО

А. П. ЕРШОВ

С. А. ЖДАНОВ

Б. В. ЛОМОВ

Ю. В. ЛУИЗО

(зам. главного
редактора)

Н. Г. МЕЛЬДИАНОВ

И. С. ОРЕШКОВ

О. К. ПАВЛОВА

А. Ю. УВАРОВ

А. И. ФУРСЕНКО

В. О. ХОРОШИЛОВ

Обложка *Э. Бажилина*

В оформлении номера принимали участие *Э. Бажилин, А. Лобашинский, А. Пономарев, С. Расторгуев.*

Издательство «Педагогика» Академии педагогических наук СССР и Государственного комитета СССР по делам издательств, полиграфии и книжной торговли

Почтовый адрес: 107847, Москва, Лефортовский пер., 8

Телефон редакции: 249-97-77

Сдано в набор 22.01.88. Подписано в печать 23.02.88. А 05644.

Формат 70×100/16. Печать офсетная. Усл. печ. л. 10,40. Уч.-изд. л. 12,47.

Усл. кр.-отт. 42,88. Тираж 95 375 экз. Заказ 146 Цена 60 коп.

Ордена Трудового Красного Знамени Чеховский полиграфический комбинат ВО «Союзполиграфпром» Государственного комитета СССР по делам издательств, полиграфии и книжной торговли.
142300, г. Чехов Московской области.

© Издательство «Педагогика», «Информатика и образование», 1988

Начало сотрудничества

По приглашению руководителя временного научно-технического коллектива (ВНТК) «Школа-1» академика Е. П. Велихова в декабре 1987 г. Советский Союз посетили американские ученые Сеймур Пейперт и Сильвия Вир.

С. Пейперт — один из выдающихся современных теоретиков образования. Математик и психолог, руководитель группы теории познания и теории обучения в Массачусетском технологическом институте, он получил мировую славу как автор наиболее серьезного и продолжительного эксперимента в области компьютеризации школы — проекта Лого.

На международной конференции «Дети в информационный век»*, проходившей в Софии, обсуждение докладов советскими и американскими учеными (А. П. Ершовым, А. Л. Семеновым, с одной стороны, и С. Пейпертом, М. Шаффер — с другой) сопровождалось глубоким и содержательным обменом мнениями по проблемам школьной информатики. Надо отметить, что и в СССР, и в США многие пришли в программирование и информатику из математики. Весьма поучительным оказалось проследить влияние математической культуры, глубокой и разносторонней в обеих странах, на бурно развивающуюся

сегодня информатику. Именно в области математической логики, теории алгоритмов и их приложений в вычислительных науках существовали идеи, положенные в основу «взаимовыгодного обмена». Организационным итогом научных контактов, к которым присоединились и болгарские коллеги, явился замысел создания небольшой исследовательской группы, нашедший поддержку в Академии наук Болгарии и Академии наук СССР. В задачу группы входит изучение фундаментальных психологических и педагогических аспектов компьютеризации образования и анализ долгосрочных тенденций ее развития.

В июле 1987 г. Москву посетила Ш. Тёркл, сотрудница группы С. Пейперта, автор известной книги о психологических проблемах взаимодействия человека и машины «Второе Я». Выступления Ш. Тёркл в ВНТК «Школа-1» вызвали большой интерес у советских коллег.

Развитие контактов было продолжено в сентябре 1987 г., когда в составе группы американских ученых в СССР приехала Сильвия Вир, врач по образованию, автор многих работ по педагогической психологии и, в частности, по применению компьютеров в обучении детей с различными дефектами развития. Тогда С. Вир посетила базовые школы-лаборатории и институты ВНТК «Школа-1». Была достигнута договоренность о более продолжительном рабочем визи-

* См.: Информатика и образование. 1987. № 6. С. 124.

те представителей группы Пейперта в Советский Союз.

И вот С. Вир и С. Пейперт в Москве. Они сразу же объявили, что их интересуют не парадные панорамные доклады. Прежде всего необходима продуктивная работа с теми советскими специалистами, которые сейчас реально занимаются созданием педагогических программных средств и осваивают их в школах.

ВНТК «Школа-1» в ряде своих подразделений уже давно ведет экспериментальную работу на базе языка Лого. Незадолго до приезда С. Пейперта был организован «кликбез» по его педагогической концепции и языку Лого для членов ВНТК.



4



Сеймур Пейперт
проводит занятие
с московскими
школьниками

В Москве состоялись лекции С. Пейперта, был организован показ программного обеспечения для системы Logowriter, конструктора на базе Logo для роботов, управляемых ЭВМ. (Об этих учебных средствах мы подробнее расскажем в последующих публикациях.) Американские ученые посетили механико-математический факультет МГУ, где познакомились с разработками в области компьютеризации обучения, ведущимися под руководством А. Г. Кушнirenко; встретились со специалистами базовых институтов ВНТК «Школа-1»: Научного совета АН СССР по комплексной проблеме «Кибернетика», Института психологии АН СССР, НИИ ШОТСО АПН СССР, Института программных

систем АН СССР (г. Переславль-Залесский), Научно-технического объединения АН СССР (Ленинград).

С точки зрения перспектив дальнейшего сотрудничества важную роль сыграло посещение американскими учеными школ-лабораторий ВНТК «Школа-1» в Москве и Переславле-Залесском, а также специализированной московской школы для глухих и слабослышащих детей.

В школе № 57 американские ученые отметили атмосферу взаимопонимания, плодотворное сотрудничество учащихся и преподавателей математики и информатики, многие из которых — вчерашние выпускники самой школы, а сегодня — студенты, аспиранты, научные со-

трудники. Такая преемственность, непрерывный приток в школу научных достижений делает ее, по мнению С. Пейперта, уникальным местом для проведения планируемых учеными СССР, США, НРБ и Японии совместных исследований по изучению фундаментальных проблем компьютеризации образования.

Истинное восхищение у американских коллег вызвала самоотверженность, которую проявляют педагоги-воспитатели школы для глухих, работающие с группой ребят, почти лишенных также и возможности видеть. Но восторги были смешаны с горечью и недоумением: все свободное время учителя-энтузиасты тратят на переписывание аршинными буквами детской классики, а ведь лазерный принтер помог бы в тысячи раз ускорить издание «Чука и Гека», причем варьирования размера шрифта для разных групп детей достичь совсем просто.

Интерес к компьютеризации дефектологии привел наших гостей в НИИ дефектологии АПН СССР, где они познакомились с работами по обучению чтению слабовидящих и другими исследованиями.

В рамках пребывания С. Пейперта в Москве состоялось первое заседание организованной на конференции в Софии группы по изучению фундаментальных вопросов использования новой информационной технологии в образовании. Группа получила название «Школа Омега». В заседании приняли участие С. Пейперт, Б. Пенков (Проблемная группа по образованию Академии наук Болгарии), А. П. Ершов, А. Л. Семенов, А. Ю. Уваров. Были намечены очередные шаги в научной и организационной деятельности группы, подписаны программные документы. Важным для развития научного сотрудничества между двумя странами явились беседы С. Пейперта и С. Вир с Е. П. Велиховым, И. М. Гельфандом, А. П. Ершовым, В. И. Кузнецовым.

В заключение хочется отметить энтузиазм, с которым американские коллеги преодолевали встретившиеся на их пути трудности, начиная с языкового и климатического барьеров и кончая непривычным и несколько запаздывавшим завтраком, и выразить надежду на дальнейшее плодотворное сотрудничество.

В. БЕЛИКОВ, А. СЕМЕНОВ

Издательство «Педагогика» осуществляет перевод книги С. Пейперта «Переворот в сознании: дети, компьютеры и плодотворные идеи», выпуск в свет которой запланирован в 1989 г. Обзор книги см. в следующем номере журнала. А пока приводим высказывание о книге президента международного общества по изучению искусственного интеллекта Тима О'Ши: «Книга Пейперта — наиболее ценная из всех вышедших книг о роли компьютера в образовании».

Л. ФУКС

Томский государственный университет

Алгоритмы поиска и сортировки

Алгоритмы информационного поиска и сортировки настолько тесно связаны друг с другом, что образуют фактически отдельный класс алгоритмов. Он интересен и с точки зрения обучения, и с точки зрения использования при решении многих задач. В то же время он имеет свою ярко выраженную специфику, связанную с тем, что внешне тривиальные задачи *найти* или *упорядочить* допускают разнообразные решения.

Все приводимые алгоритмы разрабатываются с использованием метода пошаговой детализации, основные идеи которого разъясняются в методических указаниях по курсу «Основы информатики и вычислительной техники» — «Алгоритмы, алгоритмы, алгоритмы...».

Найти нужный элемент

*Кто весел — тот смеется,
Кто хочет — тот добьется,
Кто ищет — тот всегда найдет!*

Задача поиска заключается в отыскании в последовательности элемента (или нескольких элементов) с заданными свойствами его значения. Например, найти элемент с наибольшим (наименьшим) значением или найти элемент с заданным значением. Подобные задачи возникают очень часто при составлении алгоритмов разной сложности.

Для более детального анализа алгоритмов поиска сформулируем конкретные задачи.

А. Найти минимальное значение элементов последовательности (все элементы разные).

Б. Найти номер минимального элемента последовательности (все элементы разные).

В. Найти максимальный элемент и его номер в последовательности с совпадающими элементами.

Г. Найти номер элемента с заданным значением (все элементы разные).

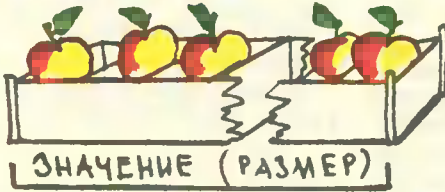
Обозначим исследуемую последовательность x_1, x_2, \dots, x_N и составим алгоритмы решения каждой из предложенных задач.

А. На ЛОС 1 задача А рассматривается как задача определения размера самого маленького яблока из лежащих в ящике, разделенном на ячейки. Сделаем из мягкой проволоки рамку размером в любое произвольное яблоко. Тем самым мы задались исходным размером; назовем рамку эталоном. Берем следующее яблоко и пытаемся пронести его через рамку. Если оно не проходит (больше эталона), то оно нас не интересует. Если же окажется, что очередное яблоко меньше эталона, то это уже претендент на наименьшее. Изменим эталон до размера этого яблока, подкрутив проволоку, и продолжаем сравнение. Когда мы таким образом пропустим все яблоки через рамку, ее размер и будет размером наименьшего.

Аналогично можно найти наибольшее яблоко, но только тогда рамку нужно

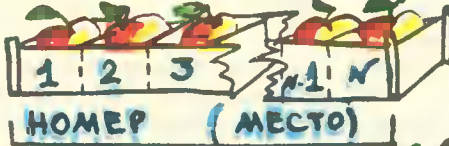
ПОИСК

А



ЗНАЧЕНИЕ (РАЗМЕР)

Б



НОМЕР (МЕСТО)

В



ЭТАЛОН



ПРЕТЕНДЕНТ



ОТКУДА

НОМЕР 1

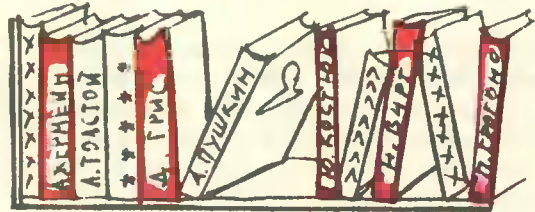
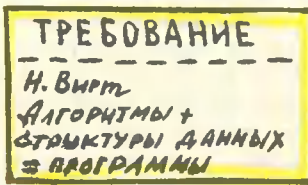
НОМЕР К

ЭТАЛОН ← ПЕРВЫЙ
 → ТЕКУЩИЙ
 → ПОСЛЕДНИЙ

7

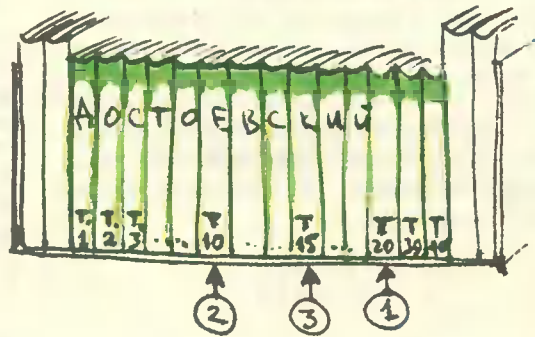
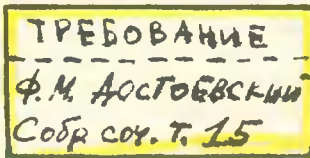
Г

(НЕУПОРЯДОЧЕН)



Д

(ДИХОТОМИЯ)



каждый раз увеличивать до размеров большего яблока.

Распространим эту идею на поиск минимального элемента в последователь-

ности. Поиск будем проводить путем сравнения всех элементов с эталоном — переменной, которой заранее присвоим значение какого-то элемента (о том, ка-

переменной NOM не будет присвоено никакого значения, а ведь она является результатом! Следовательно, при инициализации цикла переменной NOM должно быть присвоено начальное значение 1.

Законченный алгоритм определения номера минимального элемента имеет вид:

алг НОММИН (цел N, вещ таб X[1:N], цел NOM)

апр X, N
рез NOM
нач цел I, вещ MIN
I:=2; MIN:=X[I]; NOM:=1;

пока I ≤ N
нц если MIN > X[I]
то MIN := X[I]; NOM := I
все
I := I + 1
кц

кон

Полученный алгоритм очень легко превратить в алгоритм определения номера максимального элемента. Для этого условие в условном операторе напишем в виде MIN < X[I]. Конечно, эталонную переменную лучше в этом случае назвать MAX.

При составлении подпрограммы на языке Бейсик учтем, что цикл работает ровно N-1 раз, от I=2 с шагом 1 до I=N. Следовательно, можно использовать цикл FOR. Тогда не нужно делать начальное присваивание I:=2, и переход к новому элементу I:=I+1 выполнится автоматически.

```
10 LET MIN = X(1)
15 LET NOM = 1
20 FOR I = 2 TO N
25 IF MIN <= X(I) THEN GOTO 40
30 LET MIN = X(I)
35 LET NOM = I
40 NEXT I
45 RETURN
```

9

```
const m = 10;
type mass = array [1..m] of real;
procedure nommin (x:mass; n:integer; var nom:integer);
  var i : integer; min : real;
begin min := x[1]; nom := 1;
  for i := 2 to n do
    if min > x[i]
      then begin min := x[i]; nom := i
            end
end;
```

```
SUBROUTINE NOMMIN (X,N,NOM)
DIMENSION X(N)
REAL MIN
MIN = X(1)
NOM = 1
DO 1 I = 2,N
IF (MIN .LE. X(I)) GOTO 1
MIN = X(I)
NOM = I
1 CONTINUE
RETURN
END
```

▲ Процедура для определения номера минимального элемента последовательности на Паскале.

◀ Подпрограмма на Фортране.

В. Когда ставится задача нахождения минимального или максимального элемента и его номера в последовательности с совпадающими значениями, нужно оговорить особо, какой именно — первый или последний — из одинаковых элементов считать искомым. На ЛОС 1 эта задача поясняется поиском первого (последнего) из самых больших яблок. Римские цифры I, II, III указывают на порядок следования яблок одинаковой величины.

Структура алгоритма в любой постановке задачи остается неизменной, отличие отражается только на сравнении эталонной переменной с текущим элементом массива. Поясним ситуацию на конкретном численном примере.

10

Номер элемента	1	2	3	4	5	6	7	8
Значение элемента	2 ₁	4	2 ₂	1 ₁	7	2 ₃	9	1 ₂

Индексы показывают порядок следования совпадающих элементов. По алгоритму Б мы присвоим эталонной переменной MIN значение 2, а номеру минимального элемента NOM — значение 1. Затем сравним X[2] с MIN; так как 2₁ не больше 4, то MIN не изменяется; переходим к сравнению X[3] с MIN. Условие 2₁ > 2₂ тоже не выполняется, поэтому MIN остается равным 2, а NOM — 1. После сравнения X[4] и MIN происходит изменение значений: MIN становится равно 1₁, а NOM — 4. Такими они останутся до конца, так как при сравнении MIN с X[8], т. е. при проверке условия 1₁ > 1₂, переписывания MIN и NOM не произойдет.

Отсюда следует вывод: если для последовательности с совпадающими значениями поставлена задача отыскания первого по порядку следования минимального элемента и его номера, то нужно применить алгоритм задачи Б без изменений. Если ищется последний по порядку минимальный элемент и его номер, то в сравнение эталонной переменной и текущего элемента нужно добавить знак равенства (сделать $MIN \geq X[I]$); тогда переписывания MIN и NOM будут происходить и для совпадающих элементов, и NOM получит интересующее нас последнее значение.

Г. Пусть задана последовательность $\{x_j, j=1, N$ и переменная P, которая назы-

вается поисковой. Известно, что в $\{x_j\}$ все элементы имеют разные значения. Требуется определить номер элемента, значение которого равно значению переменной P. Так как сравнение вещественных величин на равенство не имеет смысла, считаем, что массив X и переменная P — целые (они могут быть и литерными).

Самый поверхностный анализ этой задачи выявляет одну из ее особенностей: в последовательности может не оказаться элемента со значением P. Такую ситуацию обязательно нужно предусмотреть при конструировании алгоритма.

Если провести аналогию между поиском нужного элемента в последовательности и поиском нужной книги на полке, что и сделано на ЛОС1, то становится очевидным, что в неупорядоченной последовательности нет иного способа, кроме последовательного просмотра элементов (книг) и сравнения их с поисковым значением (требованием на книгу). В упорядоченной же последовательности поиск можно организовать более оптимальным, с точки зрения быстрейшего действия, способом. Поэтому разграничим эти две ситуации.

Г1 (неупорядоченная последовательность).

Просматриваем последовательно все элементы (книги на полке) и сравниваем их с поисковым значением (автора и название книги — с требованием). Когда обнаружится искомый элемент (книга), запомним его номер (место книги на полке).

Основой алгоритма, таким образом, является цикл.

инициализация цикла	1
<u>пока</u> есть элементы	2
<u>нц</u>	
сравнить очередной элемент	
с поисковой переменной	3
<u>кц</u>	

При инициализации цикла, как обычно, нужно задать исходное значение I индексу I элемента. Затем вспомним, что наш алгоритм должен однозначно реагировать на ситуацию, когда искомого значения в последовательности нет. Результатом K является номер найденного элемента, поэтому резонно в си-

туации, когда элемент не найден, считать результат нулевым. Сделать это присваивание нужно до входа в цикл, так как иначе придется обременять алгоритм дополнительными проверками.

В итоге получаем:

1 $I:=1; K:=0$

Условие 2 — это сравнение текущего номера элемента с длиной последовательности:

2 $I \leq N$

Особый интерес вызывает, конечно, детализация шага 3. Ведь на основании сравнения мы должны будем сделать выбор: продолжать сравнения дальше или закончить цикл. Запишем:

3 если $X[I]=P$
то запомнить номер элемента; 3.1
закончить цикл 3.2
иначе перейти к следующему сравнению 3.3

все

Получим

При составлении процедуры на Паскале нужно учесть, что цикл работает неопределенное число раз, следовательно, нельзя использовать оператор FOR, а можно WHILE.

```
const m = 10;
type mass = array [1..m] of integer;
procedure poisk (x:mass; n,p:integer; var k:integer);
  var i : integer;
begin
  i := 1; k := 0;
  while i <= n do
    if x[i] = p
      then begin k := i; i := n + 1
              end
      else i := i + 1
end;
```

Тот же алгоритм, оформленный в виде подпрограммы на Бейсике. ▼

```
100 LET I = 1
105 LET K = 0
110 IF I > N THEN GOTO 145 1
115 IF X(I) <> P THEN GOTO 135
120 LET K = I
125 LET I = N + 1
130 GOTO 145
135 LET I = I + 1 2
140 GOTO 110
145 RETURN 3
```

3.1 $K:=1$

Для выполнения шага 3.2 нужно вспомнить, что цикл будет работать только при $I \leq N$. Следовательно, достаточно присвоить I любое значение, большее N, и цикл завершится. Пусть

3.2 $I:=N+1$

Шаг 3.3 не сложен:

3.3 $I:=I+1$

В итоге получим следующий алгоритм:

алг ПОИСК (цел N, цел таб X[1:N], цел P, K)

арг X, N, P

рез K

нач цел I

$I:=1; K:=0$

пока $I \leq N$

нц если $X[I]=P$
то $K:=I; I:=N+1$

иначе $I:=I+1$

все

кц

кон

11

Подпрограмма поиска в неупорядоченном массиве на Фортране. ▼

```
SUBROUTINE POISK (X,N,P)
INTEGER X(N),P
I = 1
K = 0
IF (I .GT. N) GOTO 3
IF (X(I) .NE. P) GOTO 2
K = I
I = N + 1
GOTO 3
I = I + 1
GOTO 1
RETURN
END
```

Г2 (упорядоченная последовательность).

Если последовательность упорядочена, т. е. значения элементов возрастают (убывают) с увеличением номера элемента, то к ней можно применить другой алгоритм поиска. Будем рассматривать случай возрастающей последовательности. Обратимся к ЛОС 1. Как найти нужную книгу среди расставленных по порядку томов? Если мы знаем, сколько всего томов (40), то выберем сразу средний (20) и посмотрим, тот ли это, что нам нужен. Если нет, то проверим, справа или слева находится интересующий нас том (15-й). Поскольку он должен находиться слева, то между 1-м и 20-м выбираем средний (10) и снова делаем ту же проверку. 15-й том находится правее. Делим промежуток между 10-м и 20-м томами пополам и сразу находим нужный 15-й том. Запоминаем место, где он стоит.

Ясно видно, что в процессе поиска мы сдвигаем друг к другу границы промежутка, в котором ищется нужная книга, причем после каждого сравнения изменяется только одна граница — либо верхняя, либо нижняя. Алгоритм, использующий последовательное уменьшение промежутка исследуемых данных в два раза, называется алгоритмом дихотомии (деления пополам).

Те же самые действия можно провести с элементами последовательности. Обозначим A номер наименьшего, а B — номер наибольшего элемента последовательности (а затем и выделяемой ее части). С помощью метода пошаговой детализации начнем разрабатывать алгоритм. Основой его будет цикл, который выполняется неопределенное число раз. В этом цикле мы производим деление интервала номеров $[A, B]$ пополам, получаем номер C и сравниваем элемент, имеющий этот номер, с поисковой переменной P . В зависимости от результата сравнения перевычисляем A или B . Повторяем эти действия до тех пор, пока A и B не совпадут, затем делаем проверку $X[A]=P$ и по ее исходу формируем результат.

инициализация цикла

пока A и B не совпадают

нц вычислить C

1

2

3

сравнить $X[C]$ с поисковой переменной P

4

кц

проверить совпадение выделенного элемента с поисковым значением 5

При инициализации нужно задать начальные значения для A и B :

1 $A:=1; B:=N$

Шаг 2: так как A — номер меньшего элемента, а B — большего, то несовпадение A и B можно записать только так:

2 $A < B$

Шаг 3: для вычисления C нужно воспользоваться либо функцией выделения целой части числа, либо операцией деления нацело двух целых чисел DIV , так как $A+B$ может быть нечетным числом.

3 $C := (A+B) DIV 2$

или

3 $C := INT((A+B)/2)$

Шаг 4: любое сравнение реализуется с помощью условного оператора:

4 если $X[C] < P$

то пересчитать A 4.1

иначе пересчитать B 4.2

Шаги 4.1 и 4.2, несмотря на свою внешнюю простоту, очень ответственны. От того, насколько правильно будет произведен пересчет, зависит своевременное окончание работы цикла. По условию 2 цикл прекратит свою работу при $A=B$. Как достичь этого? Пусть мы уже несколько раз сделали пересчет A и B и дошли до соотношения между ними $A+1=B$. Это означает, что при очередном сравнении мы будем искать нужный элемент среди двух расположенных рядом: $X[A]$ и $X[B]$. Согласно шагу 3, вычисленное C будет равно A (это легко проверить на примере: $(7+8)/2=7,5$; $INT(7,5)=7$). Следовательно, сравнение $X[C] < P$ в этом частном случае выполнится как $X[A] < P$. Если это условие истинно, то искомым элемент, возможно, есть $X[B]$, другого выбора у нас уже нет. Тогда нужно сделать так, чтобы A стало равным B . Напомним, что $A=C$, $B=C+1$, значит,

4.1 $A:=C+1$

Если же $X[A] < P$ — ложно, то искомым элемент, возможно, есть $X[A]$. Тогда для завершения работы цикла нужно положить

4.2 $B:=C$

Легко проверить, что такие присваивания не противоречат выбранному нами алгоритму на любом шаге работы цикла.

И наконец, переходим к шагу 5. Поясним, почему такую проверку нужно сделать. Дело в том, что в цикле мы не делаем проверки на совпадение какого-либо элемента с поисковым, мы только определяем, меньше этот элемент или не меньше P . И вот теперь нам предстоит убедиться в том, что этот элемент, а его индекс A (или B), действительно искомым:

```
5  если  $X[A] = P$ 
    то считать результатом  $K = A$ 
    иначе считать
        результатом  $K = 0$ .
```

Ситуация $X[A] \neq P$ означает, что нужного элемента в последовательности нет; тогда, как и в случае неупорядоченной последовательности, мы считаем, что искомым номер равен 0.

```
5  если  $X[A] = P$ 
```

```
100 LET A=1
110 LET B=N
120 IF A>=B THEN 180
130 LET C=INT((A+B)/2)
140 IF X(C)<P THEN 160
150 LET B=C : GOTO 170
160 LET A=C+1
170 GOTO 120
180 IF X(A)=P THEN 200
190 LET K=0 : GOTO 210
200 LET K=A
210 RETURN
```

```
const m = 10;
type mass = array [1..m] of integer;
procedure dpoisk (x:mass; n,p:integer; var k:integer);
var c,a,b : integer;
begin
  a := 1; b := n;
  while a < b do
    begin c := (a + b) div 2;
          if x[c] < p
            then a := c + 1
            else b := c
          end;
    if x[a] = p
      then k := a
      else k := 0
    end;
end;
```

```
то  $K := A$ 
иначе  $K := 0$ 
```

Итак, получаем алгоритм дихотомического поиска в упорядоченном массиве.

```
алг ДИХ (цел  $N$ , цел табл  $X[1:N]$ ,
```

```
цел  $P, K$ )
```

```
арг  $X, N, P$ 
```

```
рез  $K$ 
```

```
нач цел  $A, B, C$ 
```

```
 $A := 1; B := N$ 
```

```
пока  $A < B$ 
```

```
нц  $C := \text{INT}((A+B)/2)$ 
```

```
если  $X[C] < P$ 
```

```
то  $A := C + 1$ 
```

```
иначе  $B := C$ 
```

```
все
```

```
кц
```

```
если  $X[A] = P$ 
```

```
то  $K := A$ 
```

```
иначе  $K := 0$ 
```

```
все
```

```
кон
```

13

Подпрограмма на Бейсике.

Процедура на Паскале. ▼

```

SUBROUTINE DPOISK (X,N,P,K)
  INTEGER X(N),P,A,B,C      2   A=C+1
  A=1                       3   GOTO 1
  B=N                       4   IF (X(A) .EQ. P) GOTO 5
1  IF (A .GE. B) GOTO 4     K=Ø
  C=INT((A+B)/2)           GOTO 6
  IF (X(C) .LT. P) GOTO 2  5   K=A
  B=C                       6   RETURN
  GOTO 3                   END
    
```

Упорядочить последовательность

14 Сортировка относится к алгоритмам обработки таблиц (массивов) любого типа. Смысл этой задачи заключается в перестановке элементов таблицы в определенном порядке. Отсортировать числовую таблицу — это значит переставить в ней элементы так, чтобы они расположились в порядке убывания (возрастания) значений с возрастанием номера элемента.

Пример 1.

Номер элемента	1	2	3	4	5
Исходная таблица	7	12	1	49	3
Упорядоченная по возрастанию таблица	1	3	7	12	49

Пример 2.

Номер элемента	1	2	3	4	5
Исходная таблица	4	2 ₁	17 ₁	2 ₂	17 ₂
Упорядоченная по возрастанию таблица	2 ₁	2 ₂	4	17 ₁	17 ₂

Здесь индексы показывают относительный порядок элементов с равными значениями.

Сортировка таблицы литерного типа обычно заключается в упорядочении значений по алфавиту.

Пример 3.

Номер элемента	1	2	3	4	5
Исходная таблица	МИР	СОН	ТУР	КОЛ	ЕЛЬ
Упорядоченная по возрастанию таблица	ЕЛЬ	КОЛ	МИР	СОН	ТУР

Метод сортировки называется устойчивым, если относительный порядок эле-

ментов с равными значениями не меняется при упорядочении.

Сортировку можно рассматривать и как самостоятельную задачу (например, для получения упорядоченного по алфавиту списка сотрудников какого-либо учреждения), и как вспомогательную — для облегчения последующего поиска элементов в упорядоченной таблице.

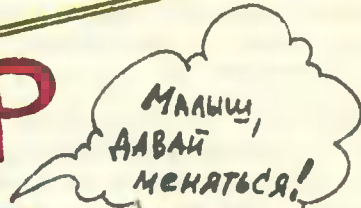
Известные алгоритмы сортировки данных, расположенных в оперативной памяти, чрезвычайно разнообразны. Их анализ очень полезен с точки зрения обучения, так как в них используются практически все универсальные приемы конструирования алгоритмов любой сложности. По словам Н. Вирта, «создается впечатление, что можно построить целый курс программирования, выбирая примеры только из задач сортировки». Интересен этот класс алгоритмов еще и тем, что на нем наглядно демонстрируются богатейшие возможности программирования: насколько разными путями можно прийти к одной цели — получению упорядоченной таблицы! На множестве алгоритмов сортировки становится понятной необходимость оценки качества алгоритма, критериями которого являются экономия памяти и увеличение быстродействия.

В настоящей статье рассматриваются три наиболее простых способа сортировки. На ЛОС 2 их поясняют: последовательность разновысоких треугольников — метод вставок; прямоугольников — метод выбора; «пузырьки, всплывающие в резервуаре» — метод обмена. Несмотря на свою внешнюю несложность и очевидность, они интересны тем, что практически все другие возможности упорядочения являют-

СОРТИРОВКА

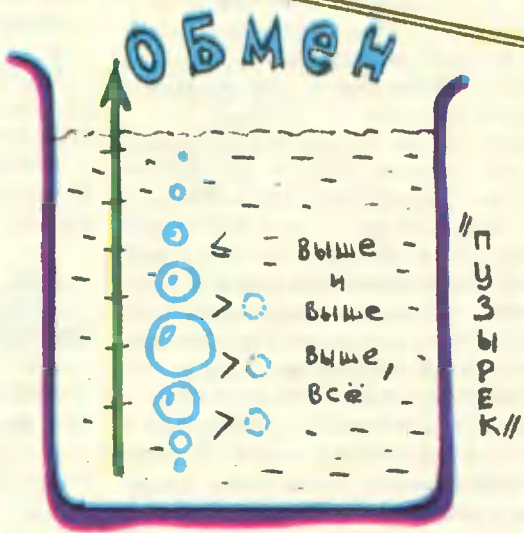


ВЫБОР



15

Эт, раз,
Еще раз,
Еще много-
много
раз!



ся производными этих трех. Эти методы выбраны также и по соображениям эффективности алгоритмов в смысле экономии памяти. Они позволяют перепорядочить таблицу, не прибегая к

вспомогательной таблице, правда, первоначальный вид таблицы после обработки ее такими алгоритмами теряется. Что касается быстродействия этих алгоритмов, то для небольших учебных

таблиц оно примерно одинаково.

Все методы сортировки рассчитаны на обработку одномерных таблиц. При необходимости двумерную таблицу всегда можно представить в виде одномерной путем пересчета индексов.

Для сохранения общности изложения рассматриваются алгоритмы упорядочения исходной последовательности A_1, A_2, \dots, A_N по возрастанию. Переход к алгоритмам упорядочения по убыванию не представляет труда.

Простой выбор. Это очень естественный способ сортировки, обычно он первым приходит на ум человеку, столкнувшемуся с необходимостью упорядочения таблицы. Идея его такова. Найдем в таблице элемент с наименьшим значением и поменяем его местами с первым элементом. После этого те же действия проделаем с оставшимися (без первого) $N-1$ элементами таблицы, затем с $N-2$ элементами и т. д., пока не останется один элемент — последний. Он будет наибольшим.

На ЛОС 2 этот метод отображен для последовательности прямоугольников разной высоты. Два первых элемента уже упорядочены. Затем отыскивается минимальный среди остальных. Если элементы в последовательности могут быть равными, то отыскивается первый (I) среди минимальных (таким образом достигается устойчивость сортировки). Он меняется местами с третьим.

Для составления алгоритма решения этой задачи воспользуемся, как обычно, методом пошаговой детализации. Обратим внимание на то, что для получения результата нам нужно $N-1$ раз находить минимальное значение в таблице, длина которой уменьшается и определяется тем, в который раз мы приступили к поиску минимума. Так, в 1-й раз мы ищем минимум среди элементов A_1, A_2, \dots, A_N , во 2-й — среди A_2, \dots, A_N , в I-й — среди A_1, \dots, A_N . И еще один важный момент. Поскольку после нахождения минимального элемента его нужно будет поменять местами с другим элементом таблицы, то нужно запомнить номер минимального. Следовательно, первый шаг детализации можно представить таким оператором цикла:

I:=1

пока $I \leq N-1$

нц найти минимальный элемент и его номер в таблице A_1, \dots, A_N 1
поменять местами A_1 и минимальный элемент 2
I:=I+1

кц

Еще раз приведем разработанный ранее алгоритм поиска минимума.

1 K:=I; X:=A[I]; J:=I+1

пока $J \leq N$

нц если $A[J] < X$

то $X:=A[J]$; K:=J

все

кц

После выполнения этих действий значением переменной X будет значение минимального среди элементов A_1, \dots, A_N , а значением K — номер этого элемента. Следовательно, пункт 2 можно записать конкретнее:

поменять местами элементы A_1 и A_K 2

Известно, что в общем случае для взаимной перестановки в памяти двух переменных нужна третья, вспомогательная:

2 C:=A[I]; A[I]:=A[K];

A[K]:=C.

Заметим, однако, что в нашей конкретной ситуации эта третья переменная не нужна, так как ее роль играет переменная X из пункта 1:

2 A[K]:=A[I]; A[I]:=X.

Мы сэкономили одно присваивание, но это немало, так как действия выполняются в цикле многократно.

Теперь можно записать полностью алгоритм сортировки простым выбором. алг ВЫБОР (цел N, вещ таб A[1:N])

арг A, N

рез A

нач вещ X, цел I, J, K

I:=1

пока $I \leq N-1$

нц K:=I; X:=A[I]; J:=I+1

пока $J \leq N$

нц если $A[J] < X$

то $X:=A[J]$; K:=J

все

J:=J+1

кц

A[K]:=A[I]; A[I]:=X;

I:=I+1

кц

кон

На языке Паскаль оформим этот алгоритм в виде процедуры. Параметрами ее будут массив и его длина.

```

const m = 10;
type mass = array [1..m] of real;
procedure selec (n:integer; var a:mass);
  var i,j,k : integer; x : real;
begin
  for i := 1 to n-1 do
    begin
      k := i; x := a[i];
      for j := i + 1 to n do
        if a[j] < x
          then begin x := a[j]; k := j
                end;
      a[k] := a[i]; a[i] := x
    end
  end;
end;

```

```

100 FOR I=1 TO N-1
110 LET K=I : LET X=A(I)
120 FOR J=I+1 TO N
130 IF A(J)>=X THEN 150
140 LET X=A(J) : LET K=J
150 NEXT J
160 LET A(K)=A(I) : LET A(I)=X
170 NEXT I
180 RETURN

```

```

SUBROUTINE SELEC (A,N)
DIMENSION A(N)
NI=N-1
DO 1 I=1,NI
K=I
X=A(I)
I1=I+1
DO 2 J=I1,N
IF (A(J) .GE. X) GOTO 2
X=A(J)
K=J
CONTINUE
A(K)=A(I)
A(I)=X
CONTINUE
RETURN
END

```

17

На языке Бейсик подпрограмма ▲ 2 будет иметь такой вид.

Она же на Фортране. ► 1

Простой обмен. Любой метод сортировки так или иначе связан с обменом, т. е. с перестановкой двух элементов в памяти. Но если для других методов это действие является вспомогательным, то для обменной сортировки это основная характеристика процесса.

Идея сортировки простым обменом заключается в многократных попарных сравнениях рядом стоящих элементов таблицы и перестановке этих элементов в заданном порядке. Пусть нам задана таблица:

Номер элемента	1	2	3	4	5
Значение элемента	7	49	1	12	3

Будем просматривать ее от конца к началу (в принципе это не обязательно, можно организовать и обычный просмотр от начала к концу). Сравним 4-й и 5-й элементы. Они расположены не в порядке возрастания, поменяем их местами. Теперь значением 4-го элемента будет 3, а 5-го — 12. Сравним далее 3-й и 4-й элементы. Их оставим на месте. Сравнение 2-го и 3-го элементов показывает, что их нужно переставить. Теперь значение 2-го элемента — 1. И наконец, сравним 1-й и 2-й элементы. Их тоже нужно поменять местами. Таким образом, получим:

Номер элемента	1	2	3	4	5
Значение элемента	1	7	49	3	12

элементы $A_N, A_{N-1}, \dots, A_1, A_{1-1}$
 $I := I + 1$

Мы видим, что в результате наших действий минимальный элемент переместился на первое место в таблице. Очевидно, в дальнейшие просмотры таблицы его уже не нужно включать. Очевидно также, что каждый следующий просмотр будет приводить к постановке очередного конца рассматриваемой части таблицы. В результате $N-1$ просмотра мы получим упорядоченную таблицу.

Если вообразить себе, что элементы таблицы — это пузырьки в резервуаре с водой, каждый весом, равным своему значению (что и изображено на ЛОС 2), то каждый проход с обментами по таблице приводит к «всплыванию пузырька» на соответствующий его весу уровень. Благодаря такой аналогии сортировка простым обменом получила название сортировки методом «пузырька». В примере на ЛОС 2 первые два элемента последовательности уже упорядочены и «всплывает» третий элемент. Знак \leq (а не $<$) показывает условие прекращения сравнений, именно этим достигается устойчивость сортировки «пузырьком».

Составим алгоритм решения данной задачи. Ясно, что основой алгоритма будет цикл, выполняющийся $N-1$ раз. Выбор границ (1 и $N-1$ или 2 и N) повлияет затем на задание индексов сравниваемых элементов, и только. Остановимся на второй паре границ.
 $I := 2$

пока $I \leq N$
нц сравнить попарно

кц
Разберем более детально пункт 1. Для попарного сравнения элементов нужен оператор цикла с границей, зависящей от I , так как при каждом новом проходе по таблице длина ее будет уменьшаться.

$J := N$
пока $J \geq 1$
нц сравнить A_J и A_{J-1} и при необходимости поменять их местами 1.1
 $J := J - 1$

кц
Пункт 1.1. уже легко записать в виде условного оператора:

1.1 если $A[J-1] > A[J]$
то $X := A[J-1]; A[J-1] :=$
 $A[J]; A[J] := X$

все
Объединим теперь все шаги детализации в законченный алгоритм.

алг ОБМЕН (цел N , вещ таб $A[1:N]$)
арг N, A
рез A

нач цел I, J , вещ X
 $I := 2$

пока $I \leq N$
нц $J := N$
пока $J \geq 1$
нц если $A[J-1] > A[J]$
то $X := A[J-1]$
 $A[J-1] := A[J]$
 $A[J] := X$

все
 $J := J - 1$

кц
 $I := I + 1$

кц

кон

На языке Паскаль оформим этот алгоритм в виде процедуры.

```
const m = 10;
type mass = array [1..m] of real;
procedure bubble (var a:mass; n:integer);
  var i, j : integer; x : real;
begin for i := 2 to n do
  for j := n downto 1 do
    if a[j - 1] > a[j]
    then begin
      x := a[j - 1]; a[j - 1] := a[j];
      a[j] := x
    end
  end;
end;
```

```

100 FOR I=2 TO N
110 FOR J=N TO I STEP -1
120 IF A(J-1)<=A(J) THEN 140
130 LET X=A(J-1) : LET A(J-1)=A(J) : LET A(J)=X
140 NEXT J
150 NEXT I
160 RETURN

```

```

SUBROUTINE BUBBLE (A,N)
DIMENSION A(N)
DO 1 I=2,N
DO 2 J=I,N
K=N-J+I
IF (A(K-1) .LE. A(K)) GOTO 2
X=A(K-1)
A(K-1)=A(K)
A(K)=X
2 CONTINUE
1 CONTINUE
RETURN
END

```

На языке Бейсик подпрограмма сортировки методом «пузырька» будет иметь такой вид. ▲

Тот же алгоритм в виде подпрограммы на Фортране. ►

Простые вставки. Пусть в заданной последовательности A_1, A_2, \dots, A_N первые $I-1$ элементов уже отсортированы. Найдем в этой части последовательности место для I -го элемента. Для этого будем сравнивать его по порядку со всеми элементами, стоящими левее, до тех пор пока не окажется, что некоторый $A_K \leq A_I$. Затем сдвинем часть последовательности $A_{K+1}, A_{K+2}, \dots, A_{I-1}$ на один элемент вправо и освободим таким образом место A_{K+1} для элемента A_I , куда его и поставим. Эта последовательность действий отображена на ЛОС 2 при упорядочивании последовательности треугольников разной высоты. Считая, что первые три элемента уже упорядочены, ищем место для четвертого по вышеизложенному правилу. Знак \leq (а не $<$) показывает, когда нужно прекратить сравнения, т. е. движение влево по последовательности. При этом достигается устойчивость сортировки вставками.

После того как мы сделаем подобные действия для всех элементов от 2-го до N -го, мы получим отсортированную последовательность. Отметим очень важную деталь в наших действиях. Когда мы проводим сравнение I -го элемента со всеми, стоящими левее него, может оказаться, что не найдется такого A_K , что $A_K \leq A_I$; это произойдет, если A_I меньше всех левых элементов. На ЛОС 2 эта ситуация отмечена дорожным зна-

ком «Прочие опасности». В этом случае нужно закончить просмотр (по достижении первого элемента последовательности) и осуществить сдвиг A_1, \dots, A_{I-1} вправо, а элемент A_I поставить на первое место.

Составим алгоритм решения этой задачи. Как и в предыдущих случаях, основным оператором здесь будет цикл, определяющий, для какого элемента последовательности мы ищем место в упорядоченной левой части.

```

I:=2;
пока I≤N
нц найти место A_K для A_I в упорядоченной части последовательности 1
сдвинуть элементы A_{K+1}, A_{K+2}, ..., A_{I-1} вправо 2
поставить элемент A_I на нужное место 3
I:=I+1
кц

```

В результате действия 1 мы должны получить номер K , индекс элемента, рядом с которым справа нужно поставить A_I .

Чтобы найти место для A_I , нужно сравнивать его последовательно со всеми левыми соседями до элемента $A_K \leq A_I$ и, если такового не окажется, остановиться на первом элементе. Действия эти — циклические, причем удобнее

оформить цикл по текущему номеру элемента. Получим:

```
1   инициализация цикла           1.1
   пока J > 0
     нц сравнить элементы Ai и Aj
                                     1.2
   кц
```

Обдумывая пункт 1.1 — инициализацию цикла, мы должны предусмотреть три момента. Во-первых, значение элемента A_i нужно запомнить, так как иначе при сдвиге оно может потеряться. Во-вторых, нужно зафиксировать номер I—1 — с этого элемента начнется сравнение. В-третьих, нужно позаботиться о том, чтобы в ситуации, когда A_i меньше A₁, A₂, ..., A_{i-1}, он смог встать на первое место. Следовательно, получаем:

```
1.1 X:=A[I]; J:=I-1; K:=1
```

Далее, по результатам сравнения A_i и A_j мы должны сделать вывод о том, продолжать сравнение или нужный элемент уже найден и пора закончить цикл. Закончить цикл можно присваиванием переменной J нулевого значения. Имеем:

```
1.2 если A[I] ≥ A[J]
     то K:=J+1; J:=0
     иначе J:=J-1
```

все

Перейдем к детализации пункта 2. Для сдвига последовательности вправо нужно просматривать ее из конца в начало и последовательно сдвигать элементы.

```
2   J:=I
   пока J > K
     нц A[J]:=A[J-1]
       J:=J-1
   кц
```

Пункт 3 — это один оператор присваивания:

```
3   A[K]:=X
```

И наконец, законченный алгоритм сортировки простыми вставками.

алг ВСТАВКА (цел N, вещ таб A[1:N])

арг A, N

рез A

нач цел I, J, K, вещ X

I:=2

пока I ≤ N

нц X:=A[I]; J:=I-1; K:=1

пока J > 0

нц если A[I] ≥ A[J]

то K:=J+1; J:=0

иначе J:=J-1

все

кц

J:=I

пока J > K

нц A[J]:=A[J-1]

J:=J-1

кц

K[K]:=X;

I:=I+1

кц

кон

Соответствующая подпрограмма на Бейсике.

```
100 FOR I=2 TO N
110 LET X=A(I) : LET J=I-1 : LET K=1
120 IF J=0 THEN 170
130 IF A(I)<A(J) THEN 150
140 LET K=J+1 : LET J=0 : GOTO 160
150 LET J=J-1
160 GOTO 120
170 FOR J=I TO K+1 STEP -1
180 LET A(J)=A(J-1)
190 NEXT J
200 LET A(K)=X
210 NEXT I
220 RETURN
```

```

const m = 10;
type mass = array [1..m] of real;
procedure vstav (var a:mass; n:integer);
  var i,j,k : integer; x : real;
begin for i := 2 to n do
  begin x := a[i]; j := i - 1; k := 1;
    while j > 0 do
      if a[i] >= a[j]
        then begin k := j + 1; j := 0
              end
            else j := j - 1;
      for j := i downto k + 1 do
        a[j] := a[j - 1];
      a[k] := x
    end
  end;
end;

```

SUBROUTINE VSTAV (A,N)

DIMENSION A(N)

DO 1 I=2,N

X=A(I)

J=I-1

K=1

5 IF (J .LE. 0) GOTO 2

IF (A(I) .LT. A(J)) GOTO 3

K=J+1

J=0

GOTO 4

3 J=J-1

4 GOTO 5

2 KI=K+1

DO 6 J=KI,I

M=I-J+KI

6 A(M)=A(M-1)

A(K)=X

1 CONTINUE

RETURN

END

▲ Алгоритм сортировки простыми вставками, оформленный как процедура на языке Паскаль. 21

◀ Подпрограмма на Фортране.

высокого уровня позволяют свободно работать с информацией такого типа. Так, например, в Бейсике для ДВК-1 такие задачи вообще не решаются, а в Паскале длина литерной (символьной) переменной всегда равна 1, в Фортране нельзя использовать литерные (символьные) константы в выполняемых операторах. Наиболее полно возможности обработки данных литерного типа реализованы в языке ПЛ/1.

Вспомним: что значит сравнить две литеры? Это означает, что сравниваются их внутренние коды, т. е. внешнее сравнение литер практически выполняется как внутреннее сравнение чисел. Отсюда следует вывод: непосредственному упорядочению поддаются только те литеры, коды которых упорядочены. Упорядочены, например, буквы латинского алфавита. А вот буквы русского алфавита не упорядочены. Поэтому, прежде чем приступить к сортировке русских слов, нужно перекодировать буквы русского алфавита, а это весьма нетривиальная задача.

Все изученные способы сортировки объединяет то, что мы достигаем цели многократным повторением выбранного алгоритма для определенной части последовательности. Об этом напоминает опорный сигнал «Эх, раз, еще раз, еще много-много раз!».

Мы рассмотрели примеры программ для сортировки числовых таблиц. Не менее интересно использование этих методов для упорядочения литерной информации. К сожалению, не все языки

К концепции обучения информатике в младших классах

Глубокое влияние компьютеризации на учебно-воспитательный процесс может быть достигнуто при комплексном использовании компьютера как средства обучения начиная с младших классов.

22 Определенный опыт уже накоплен. В СССР широко известна методика обучения программированию школьников V—VIII классов, разработанная в Новосибирском Академгородке (1) на базе языка Робик, входящего в систему программирования «Школьника». Большой интерес представляет работа Проблемной группы по образованию при Болгарской академии наук и Министерстве народного просвещения Болгарии, основанная на обучении информатике в рамках новой учебной дисциплины «Язык и математика» начиная с V класса (2). В ЧССР разработано оригинальное методическое и программное обеспечение для обучения школьников основам алгоритмизации на базе исполнителя «Карел» (3). Вероятно, наиболее разработанной остается концепция С. Пейперта, основанная на языке программирования Лого (4).

Тем не менее работа в этой области носит исследовательский, экспериментальный характер и отличается как в постановке целей, так и в методических подходах. В этой статье мы остановимся на концепции применения компьютера в младших классах, содержании экспериментального курса информатики, требованиях к программному и техническому обеспечению.

В основе нашего подхода лежат следующие принципы:

1. Главные цели применения компьютеров в младших классах должны носить мировоззренческий, общекультур-

ный характер, быть направлены на развитие личности и мышления ребенка.

В частности, важнейшей задачей является формирование основ информационной культуры учащихся, т. е. овладение основными понятиями информатики (знания, данные, модель, структура, процесс и др.), методами представления знаний и умением их использовать для решения практических задач. Благодаря подходу, когда информатика рассматривается как наука о формализации знаний, межпредметные связи становятся естественными, как бы встроенными в концепцию, можно говорить об интегрированном обучении основам наук в рамках одной дисциплины — информатики.

С другой стороны, возможности современных компьютеров позволяют ставить и более широкие задачи, в частности, музыкального и эстетического образования и воспитания.

2. Содержание и методика обучения должны быть направлены на развитие теоретического мышления учащихся. Традиционная методика направлена в основном на развитие эмпирического мышления (5).

3. Процесс обучения должен сочетать использование двух взаимосвязанных и взаимодополняющих функций компьютера: компьютера как инструмента формализации знаний о предметном мире и компьютера как активного элемента предметного мира, инструмента измерения, отображения и воздействия на предметный мир. (Традиционный подход ограничивается, как правило, работой с символьными объектами.)

В частности, реализация этого принципа опирается на введение в качестве средств обучения разнообразного учебного оборудования, измерительных приборов, манипуляторов и пр., работающих под управлением ЭВМ.

4. Процесс обучения должен сочетать развитие логического и образного мышления, что возможно благодаря широ-

* Доклад на совещании специалистов стран — членов СЭВ по теме 4.2.1 Расширенной программы сотрудничества по развитию и широкому использованию в народном хозяйстве стран — членов СЭВ микропроцессорной техники на 1986—1990 гг. Рацкеве, ВНР. 23—27 сентября 1986 г.

кому использованию графических и звуковых средств.

5. Одним из ключевых принципов является применение ЭВМ для решения творческих задач проектирования и конструирования.

6. Обучение информатике должно являться составной частью предметной, в частности совместной учебной и трудовой, деятельности.

7. Для эффективного обучения решающее значение имеет коллективный характер деятельности учащихся.

Последовательное применение указанных принципов приводит к необходимости пересмотра традиционной системы образования как в плане содержания и методов, так и организации и материального обеспечения учебно-воспитательного процесса. Фактически речь должна идти о создании системы опережающего образования на основе современных психолого-педагогических концепций и новейших информационных технологий, а не о внедрении компьютера в традиционный учебный процесс. В то же время представляется, что отработка элементов нового подхода возможна уже в нынешних условиях, и в первую очередь в рамках трудового обучения младших школьников. Введение информатики в курс трудового обучения в младших классах, на наш взгляд, отвечает особенностям нашей концепции, в частности, в силу межпредметного характера трудового обучения и имеет ряд организационных достоинств.

Указанные принципы легли в основу разработки содержания экспериментального курса информатики для школьников V—VIII классов. Экспериментальная отработка элементов курса начата в двух школах Москвы.

Кратко остановимся на содержании экспериментального курса, который, реализуя принцип интеграции учебных дисциплин, ставит основную задачу развития творческих возможностей школьников в области проектирования и конструирования с использованием ЭВМ. Проектирование здесь понимается в широком смысле как описание мысленных образов еще не реализованных процессов и конструкций, что по смыслу близко

к понятию моделирования. Заметим, что проектирование в таком понимании является одним из основных видов конструктивной человеческой деятельности, а организация учебной деятельности как процессов описания и проектирования позволит сформировать у учащихся обобщенные способы решения широкого класса задач, что имеет важное мировоззренческое и прикладное значение.

Курс основан на следующем теоретическом материале, который вводится постепенно на протяжении всего обучения:

основные понятия математической логики и теории отношений;

введение в моделирование с помощью компьютера;

обучение процедурам, лежащим в основе описания процессов и конструкций: классификация, декомпозиция, синтез, алгоритмизация.

Приведем примерное разбиение содержания обучения по годам.

V класс. Первоначальное знакомство с персональным компьютером, освоение навыков работы. Развивающие игры с компьютером по математике, языку, логике, музыке. Введение в программирование на языке Лого. Решение простейших учебных и творческих задач, в

23



частности конструирование простейших изделий. Эксперименты с электронными игрушками, управляемыми ЭВМ.

VI класс. Продолжение изучения программирования на языке Лого. Использование компьютера для обработки текстов. Углубление изучения машинной графики (плоские изображения). Формирование первоначальных представлений об анализе, синтезе, классификации, алгоритмизации, лежащих в основе проектирования и описания объектов и процессов. Решение простейших учебных задач в области математики, языка, проектирования, конструирования и технологии. Разработка простейших программ управления реальными объектами: электронными игрушками, учебными роботами и манипуляторами и др.

VII класс. Продолжение знакомства с использованием компьютеров для управления простейшими технологическими процессами и техническими объектами. Применение элементов систем обработки текстов и машинной графики при составлении технологической документации на изделия. Продолжение решения творческих задач в области проектирования, конструирования и тех-

нологии. Первоначальное знакомство с использованием ЭВМ для поиска, хранения и обработки информации.

VIII класс. Введение в логическое программирование на языке Микро-Пролог. Знакомство с информационными системами общего назначения (базы данных, электронные таблицы, простейшие базы знаний и экспертные системы) и работа с ними для решения задач проектирования, конструирования и управления. Простейшие деловые игры. Коллективная разработка учебно-производственных проектов.

Требования к программному и техническому обеспечению школьной информатики рассмотрены в работах (6, 7, 8). Здесь мы остановимся только на некоторых особенностях, связанных с возрастом обучаемых. Общим принципом в выборе программного и технического обеспечения является необходимость создания среды, максимально благоприятной для решения разнообразных творческих задач. В частности, если обратиться к классификации учебного программного обеспечения, предложенной в работе (7), этому принципу удовлетворяют, с одной стороны, инст-



рументальные средства (например, обеспечивающие работу с графической информацией, интерпретаторы языков программирования, базы данных и базы знаний и т. д., т. е. средства, обеспечивающие самостоятельную творческую работу учащихся), а с другой стороны, автоматизированные системы управления учебной деятельностью, формирования познавательных процессов. Заметим, что построение учебной деятельности как процессов описания и проектирования позволяет ставить вопрос о создании не только предметных, но и межпредметных программно-аппаратных учебных сред.

С точки зрения изобразительных возможностей программное обеспечение для младших школьников должно в максимальной степени использовать принцип непосредственного взаимодействия с обрабатываемыми объектами, возможности цветной динамической графики, звуковые эффекты и музыку.

На наш взгляд, существует довольно распространенное заблуждение, что компьютер для младших классов должен по своим техническим характеристикам уступать компьютеру для старших. Этот тезис может быть справедливым лишь при постановке очень ограниченных целей обучения. Наоборот, для создания отвечающей целям обучения среды компьютер должен обладать достаточно мощными ресурсами, в первую очередь иметь необходимый для работы учебного программного обеспечения объем оперативной и внешней памяти. Важнейшим программно-аппаратным компо-

нентом является быстрая локальная сеть с разнообразным набором пользовательских функций, обеспечивающих коллективную работу учащихся. Наконец, учебный процесс в младших классах должен быть «подкреплен» широким набором учебного оборудования, в частности электронными игрушками, учебными роботами и манипуляторами, измерительной аппаратурой, средствами для управления графическими объектами, музыкальными синтезаторами и т. д.

Л и т е р а т у р а

1. Звенигородский Г. А. Первые уроки программирования. М.: Наука, 1985.
2. Николов П., Сендова Е. Език и математика. София: ПГО, 1984.
3. Microelektronika SSM. Praha, 1987.
4. Papert S. Mindstorms. Children, Computers and Powerful Ideas//Basic Books, 1980.
5. Давыдов В. В. Проблемы развивающего обучения. М.: Педагогика, 1986.
6. Смирнов Е. П. Требования к техническому и программному обеспечению преподавания информатики и вычислительной техники в общеобразовательной школе: Доклады советской делегации на 5-й Международной конференции по проблемам школьного оборудования. 9—13 сентября 1985 г. Галле, 1985.
7. Смирнов Е. П. Функции компьютера в учебной деятельности и учебное программное обеспечение//Вопросы психологии. 1987. № 1.
8. Архангельский А. Г., Рязанский М. В., Смирнов Е. П. Исходные требования к перспективному комплексу учебной вычислительной техники. М.: НИИ ШОТСО АПН СССР, 1987.

Расследование поручено ЭВМ

В течении многих лет криминалисты пытаются достоверно описать как психологические, так и физические особенности правонарушителя по его действиям. Для этого, в частности, необходимо на месте преступления тщательно отбирать материалы для составления портрета преступника. По

ЧТО?
МОЖЕТ
ЭВМ

сообщениям зарубежной печати, в ближайшем будущем в США отобранные материалы следователь через персональный терминал сможет ввести в ЭВМ. Дан-

ные поступят в информационно-справочную систему, ориентированную на анализ аналогичных случаев в стране, и послужат основой для возможной идентификации преступника. Ответ на запрос может быть получен следователем оперативно. По прогнозам специалистов, умение использовать в работе микро-ЭВМ будет обычным в профессиональной деятельности юриста.

А. КРИВЦОВ
Канд. техн. наук

«Агат» в образовании: 1983—1987

В мае 1983 г. на урок математики в IX классе школы № 166 г. Новосибирска пришли Геннадий Анатольевич Звенигородский, ученик VI класса Леня Рабинович и автор этих строк. Мы принесли с собой «Агат» — это был пятый макетный образец, на котором прошедшей ночью были завершены работы по переносу Apple-реализации системы программирования Рапира (по современной нумерации версия А1.—1). Машину установили, включили, Леня на ней поработал, и урок не был сорван, в чем и заключался главный результат этого эксперимента. В августе того же года макет эксплуатировался на летней школе юных программистов в интересах отладки Рапиры.

В 1984 г. ЛЭМЗ (Лианозовский электромеханический завод) выпустил первую опытную партию «Агатов». К летней школе удалось привезти в Новосибирск 5 машин и подготовить версию А1.0 Рапиры, что позволило провести на школе первые систематические занятия по программированию. Здесь же появился первый опыт ремонта в условиях эксплуатации и родился термин “пятый «Агат»”, точно характеризующий метод восстановления отказавшей машины путем разорения соседней: на церемонии закрытия об-

наженный остов этого “пятого «Агата»” был торжественно вынесен на раскладушке.

В ноябре 1984 г. в школе № 166 г. Новосибирска был введен в действие первый вычислительный кабинет в составе 20 ПЭВМ «Агат» и начались регулярные практические занятия по программированию с учениками VI—X классов. Геннадий Анатольевич Звенигородский не дождался месяца до этого события.

Серийный выпуск «Агатов» был организован в 1985 г., и сразу начались широкие поставки вычислительных кабинетов на базе этой машины в учебные заведения. В результате по тиражу установок в школах, пединститутах и институтах усовершенствования учителей «Агат» уступает лишь «Электронике БК-0010», оставляя позади все отечественные ЭВМ и «Ямахи».

Стоимость одного «Агата» сейчас — 3900 руб., в учреждения просвещения поставляются комплекты из 12 и 16 ПЭВМ с одним печатающим устройством на комплект. Кабинеты поставляются по нарядам Управления снабжения школ Минпроса СССР. Заметная доля закупок оплачивается из средств шефствующих предприятий (для многих предприятий такой путь оказывается единственной возможностью получить

доступ к персональным ЭВМ, а решение вопроса оплаты облегчает получение награды в Минпросе).

Каждая ПЭВМ поставляется в комплекте с цветным телевизором (со второго полугодия 1987 г. по санитарно-гигиеническим показаниям предусмотрено комплектование черно-белыми мониторами), накопителем на гибком магнитном диске и 10 дисками (2 записанных и 8 чистых; итого с классом «Агат» поставляется 120 или 160 дисков, а с классом ДВК-БК-0010 — 5).

Записанные диски содержат проверочные тесты и систему программирования «Бейсик». Такой комплект поставки обеспечивает возможность эксплуатации каждой ЭВМ класса в качестве инструментального комплекса для разработки программного обеспечения с привлечением, как минимум, Бейсика и ассемблера (средства ассемблирования имеются в составе реализации Бейсика). Это свойство обнаружено и широко используется в большинстве учебных заведений, располагающих классами «Агатов».

Технические характеристики ПЭВМ, используемых в отрасли просвещения

Традиционными характеристиками, по которым сравниваются ЭВМ вообще и ПЭВМ в частности, являются производительность процессора и объем оперативной памяти — эти параметры обычно задаются в требованиях к ЭВМ того или иного класса, откуда они переходят в техническое задание на разработку, а после попадают в эксплуатационную и сдаточную документацию, в проспекты выставок и журнальные публикации. Спутниками «Агата» на этом пути были 64К байт ОЗУ (расширяемое до 256К) и 300 000 операций в секунду.

Программное обеспечение ЭВМ характеризуется, как правило, составом, характеристиками программных систем до последних лет широко не объявлялись. Однако с развитием рынка персональных ЭВМ у пользователя появилась острая потребность знать для выбора модели ПЭВМ такие ее характеристики, которые соответствовали бы его

уровню общения с машиной. Традиционные параметры лишь частично удовлетворяют этим требованиям, поскольку подавляющее большинство пользователей ПЭВМ работают не на машине, а на той или иной программной системе. Быстродействие процессора и объем памяти они воспринимают через призму этой системы — и вот возникли оценки интегральных характеристик ЭВМ с программными системами.

Распространено мнение (автор разделяет эту точку зрения), что интегральные характеристики более адекватно описывают быстродействие процессора, поскольку они дают единственную возможность объективного сравнения ЭВМ с различными архитектурами, в то время как «операции в секунду» требуют подробного объяснения того, каких операций и в каких условиях машина может сделать в секунду так много (еще на БЭСМ-6 удалось за секунду сложить миллион нулей).

Для сравнения производительности ПЭВМ по интегральным характеристикам сейчас наиболее широко используются тесты Benchmark [1]. Это набор из 8 коротких программ на Бейсике (их тексты есть в [1]), тестирующих различные операции языка: циклы, условные переходы, арифметические операции, работу с массивами, вызов подпрограмм, расчет тригонометрических и показательных функций. Работа по вводу этих программ в машину и измерению длительности их выполнения занимает даже на незнакомой машине не более часа. Бейсик реализуется на всех ПЭВМ; его простота и логическая бедность позволяют предполагать незначительность разброса характеристик, связанного с «плохим» программированием системы. Все это и обуславливает популярность набора Benchmark: уже принято для каждой из вновь появляющихся на рынке ПЭВМ публиковать измеренные на этих тестах параметры, как правило, в сравнении с данными распространенных ПЭВМ того же класса.

В таблице приведено время выполнения программ Benchmark для ПЭВМ, получивших наибольшее распространение в просвещении, — «Электрони-

Время выполнения тестовых программ «Benchmark» BM1—BM8 (в секундах) и другие характеристики ПЭВМ, используемых в просвещении

	«Корвет»	УКНЦ	«Ямаха»	«Агат»	БК-0010	IBM PC
BM1	1.6	0.9	2.1	1.2	2,6	1.5
BM2	6.0	0.9	5.9	8.6	40	5.2
BM3	16.2	2.6	16.9	16.3	59	12.1
BM4	16.0	3.9	18.4	18.0	68	12.6
BM5	17.0	4.1	19.5	19.8	76	13.6
BM6	31.0	7.9	32.0	30.0	101	23.5
BM7	49.0	10.8	45.3	46	148	37.5
BM8	84	25	219.8	108	136	35.1
Память, К байт: объявлено	256		128	64	32	
предоставляется Бейсиком	21		22	42	16	
Процессор: разрядность	8	16	8	8	16	
быстродействие (тыс. оп./с)	625	600	625	300	600	

Примечание. На БК-0010 проверялся интерпретатор Фокала; по оценкам разработчиков, быстродействие Бейсика на этом компьютере будет в 2—3 раза выше.

28

ка БК-0010» (КУВТ-86), «Ямаха» и «Агат», а также для планирующейся к массовой поставке в учебные заведения ПЭВМ ПК 8020 (КУВТ «Корвет») и МС 0511 (КУВТ УКНЦ) в сравнении с опубликованными [1] данными по профессиональной ЭВМ IBM PC — прототипу ЕС 1840*. Сравнение характеристик позволяет сделать следующие выводы:

несмотря на то что по объявленному значению быстродействия процессора «Агат» в 2 раза (а с учетом разрядности — и более) уступает другим ПЭВМ, пользовательская производительность его приблизительно равна производительности «Корвета» и «Ямахи», в 3—4 раза выше, чем у Фокала БК-0010 и лишь на 30—35 % ниже производительности IBM PC;

пользовательская емкость ПЭВМ «Агат» существенно больше, чем у остальных школьных ПЭВМ, что имеет большое значение при реализации учебных пакетов прикладных программ.

По максимальному разрешению графических режимов «Агат» уступает «Корвету» и «Ямахе» (соответственно 512 и 256 цветных точек по горизон-

тали против 256 черно-белых или 128 цветных точек у «Агата») и примерно соответствует БК-0010, однако опыт разработки учебных пакетов на «Агате» не выявил принципиальных ограничений, не допускающих реализации требуемых в том или ином применении изображений. Параметры раstra «Агата» удовлетворяют техническим требованиям к разработке и производству комплектов технических средств и оборудования кабинетов вычислительной техники (КУВТ) для всех типов учебных заведений, утвержденным в мае 1985 г., а непревышение этих требований позволяет увеличить пользовательскую емкость машины за счет экономии памяти, используемой для представления экрана.

Основной недостаток серийно выпускаемого «Агата» по отношению к упомянутому исходным требованиям (отсутствие возможности объединения ЭВМ кабинета в локальную сеть) преодолевается для уже существующих установок внедрением комплекта модуля межмашинного обмена (см. ниже).

Программное обеспечение

Основу программного обеспечения ПЭВМ «Агат» составляют система прог-

* Автор выражает признательность Ю. А. Шорину, выполнившему эти измерения.

раммирования «Бейсик» и пакет прикладных программ автоматизации школьного учебного процесса «Школьница», содержащий систему программирования на учебно-производственном языке Рапира и языке начального обучения Робик. Кроме них приняты к серийному выпуску система подготовки текстов «Агат-автор» и система численного моделирования (электронная таблица). В составе программного обеспечения 36-часового курса подготовки руководителей и педагогических кадров по информатике Минпросом СССР распространен также набор учебных пакетов прикладных программ по отдельным разделам различных школьных курсов. На местах разработано большое количество программ, среди которых наибольшую известность приобрели различные игры, учебные пакеты, психологические и медицинские тесты.

Языки программирования. Реализация Бейсика появилась на «Агате» в 1982 г. и предназначалась в первую очередь для разработки тестового обеспечения производства ПЭВМ и ее приемки. Для решения этих задач в реализации Applesoft ПЭВМ «Apple II+», взятой за основу, были заменены блоки текстового и графического вывода, добавлены средства ассемблирования и отладки программ на машинном языке, в интересах мнемоничности ассемблерного текста введены идентификация имен переменных по неограниченному количеству символов и шестнадцатеричные константы. В течение 1983—1984 гг. принципиальных изменений в версию не вносилось, и с началом серийного производства ПЭВМ она получила широкое распространение. Сейчас Бейсик-60 (объем файла, содержащего интерпретатор этой версии, — 60 секторов) продолжает поставляться. Некоторые программы (в том числе «Шахматы», «Шашки» и ряд других игр) содержат ссылки на адреса интерпретатора и не могут работать с другими версиями. В 1985 г. при подготовке к передаче в Государственный фонд алгоритмов и программ версия была переработана с учетом замечаний, выявленных при эксплуатации машин опытной партии, и, естественно, в соответствии с аппети-

тами разработчиков. Был заменен блок редактирования программы (появилась возможность контекстного поиска и замены участков программы по директиве LIST), введен блок хранения данных на внешнем запоминающем устройстве (директивы STORE, RECALL, CHAIN), серьезно переработана документация. Результат этих усовершенствований называется СП «Бейсик» Фг. 00012-01, имеет объем 67 секторов, передан в ГосФАП [2] и поставляется в виде комплекта документации и записанного диска Таллинским научно-учебным центром (туда с гарантийным письмом можно обратиться по адресу: 200026, Таллин, бульвар Кадака, 165, директору ТНУЦ В. И. Коваленко). Следующая версия будет содержать средства обмена по локальной сети, работать со строчными и прописными буквами, поддерживать накопитель на гибких дисках повышенной емкости и поступит на тиражирование в 1988 г. Соответствие основных точек входа интерпретаторов Бейсика-60 и Бейсика-67 предполагается опубликовать в ближайших номерах ИНФО; для программ, не содержащих кодовых обращений к интерпретатору, версии полностью совместимы. В составе программного обеспечения 36-часового курса переподготовки по информатике распространена версия Бейсика-60 с «быстрым» ДОСом, разработанная перспективной московской фирмой «ALV SOFTWARE» (Александр Голов & Василий Березутский недавно окончили школы и работают в компьютерном центре ЛЭМЗ и НИИ ШОТСО).

ППП «Школьница» разработана в ВЦ СО АН СССР под руководством одного из основоположников школьной информатики в СССР Г. А. Звенигородского замечательным коллективом сначала школьников, затем студентов и воинов Советской Армии — Наташей Глаголевой, Виталием Цикозой, Женей Налимовым, Пашей Земцовым (в качестве авторов системы я упоминаю «стариков»). Как уже было сказано, (—1)-я версия системы появилась на «Агате» в 1983 г., затем была еще одна версия на «Apple», перенос которой на «Агат» летом 1984 г. дал

версию А1.0 (впервые в полном объеме реализованы все типы данных, все режимы графики, заменен экранный редактор). Версия А1.0 эксплуатировалась весь учебный год в кабинете школы № 166, на ней «взращены» первая серия пакетов по школьным предметам и второе поколение авторов «Школьницы», среди которых надо упомянуть Настю Будневу («Сольфеджио»), Валю Трескову («Производство чугуна»), Егора Боровикова («Дежурик»), Сережу Гавриленко («Муравей» и др.), Андрея Петрова («Химия»). Весной 1985 г. версия А1.0 прошла межведомственные испытания и вскоре уступила место версии А1.1 (первая версия на серийном «Агате»; появился Робик, усовершенствован и получил новый синтаксис блок графического вывода, издана документация [3]). На летней школе юных программистов 1985 г. была объявлена и эксплуатировалась версия А1.2 (удвоен пользовательский объем системы, появился отладочный комплекс), которая к зиме переросла в А1.2' (появился ассемблер и отдельный редактор отладочного комплекса). Версия А1.2 распространялась бригадами ЦК ВЛКСМ осенью 1985 г. (Ленинград, Таллин, Рига, Паневежис, Вильнюс, Алма-Ата, Фрунзе, Ташкент). Летом 1986 г. объявлена версия А1.3, серьезно исправленная в октябре (действующее обозначение: А1.3 от 17.10.86). В ней появилась команда ЗАПУСК, исправлен текстовый вывод в графике (для работы программ, работавших на А1.2 и А1.2' с измененным алфавитом, нужно повторить редактирование алфавита), отработано наполнение системного диска (появился редактор алфавитов, звуковой вывод нот, полный комплект дисковых утилит: копирование, настройка скорости, копирек — устранение радиального смещения записи). Версия передана в ГосФАП [4] и поставляется ТНУЦ по заявкам. Для опытных программистов существенным дополнением к документации на версию А1.3 является распространенное в составе программного обеспечения 36-часовой переподготовки по информатике описание диалогового отладочного комплекса, обеспечивающе-

го работу на ассемблере, включающее инструкцию по программированию на «голой» машине и карты памяти системы «Школьница». Языки версий А1.1 — А1.3 полностью преемственны, очередные версии являлись расширением предыдущих.

Сравнение Бейсика и Рапиры в качестве опорных языков при преподавании информатики выявляет преимущества Рапиры по любым критериям. Установки центральных, областных и/или районных (уровень методического руководства преподаванием информатики зависит от оснащенности региона техникой) методистов на использование Бейсика проистекают скорее из борьбы с Фокалом и другими мнемокодами. В ряде случаев приходилось в конце учебного года переходить в приказном порядке с Рапиры на Бейсик; Г. А. Кривошеев в Черемушкинском УПК Москвы изучил со своими группами Бейсик за два спаренных практических занятия, после чего школьники продолжали работать на Рапире. Вообще, некоторое подмножество тезисов, принятых в обоснование разработки Рапиры Г. А. Звенигородским, можно кратко сформулировать так: из существующих языков для освоения компьютера лучше других подходит Бейсик, но человек, изучавший программирование на Бейсике, не имеет шансов стать хорошим программистом, поскольку вместо опыта конструирования данных и действий приобретает привычку «плести лапшу» из переходов и меток; учебный язык должен обладать диалогом не хуже, чем у Бейсика, и содержать современные структуры для описания управления и информации. Отличие Рапиры от этого направления заключено во второй половине названия «учебно-производственный язык»: система поддерживает не только обучение курсу ОИВТ, она предоставляет одновременно производственную систему программирования, которая продемонстрировала серьезные преимущества по сравнению с Бейсиком как при разработке учебных программ по школьным предметам, так и при решении задач тестирования, обработке результатов эксперимента, построении справочных систем и др.

Пакеты общего назначения. Персональная ЭВМ в настоящее время не имеет права на существование без программных пакетов, ориентированных на манипулирование данными и подготовку документов; три кита такого обеспечения общего назначения — текстовый редактор, электронная таблица и система управления базами данных (существует мнение, что сам класс «персональные ЭВМ» появился на свет благодаря разработке и широкой популярности программы VISICALC — первой электронной таблицы).

Текстовым редактором на «Агате» работают «Система подготовки текстов «Агат-автор» [5] и автономный редактор отладочного комплекса ППП «Школьница» [4]. «Агат-автор» в силу своей специализации обладает более широким набором функций, особенно функций управления выдачей на бумагу (подсистема «форматтер»), тем не менее пользователь, привыкший к диалогу редактора «Школьницы», возможно, предпочтет набирать текст не переучиваясь; этот текст затем можно через диск передать в «Агат-автор» на распечатку. Получившая распространение в 1985 г. версия «Агат-автора» сильно устарела: с тех пор появились соответствующие машинописной странице (32*64 и 16*64) режимы отображения при редактировании, возможность изменения шрифта печати внутри строки, режим замены при вводе (очень удобен при редактировании таблиц), универсальный драйвер печатающего устройства (пока не найдено принтера, под который пришлось бы его переделывать) и др. Текущая версия передана в ТНУЦ и поставляется с III квартала 1987 г.

Электронная таблица «Агата» — система численного моделирования (СЧМ) [7] — изготовлена путем адаптации VISICALC2 и в основном соответствует этой программе по набору функций. Предусмотрена возможность передачи результатов работы СЧМ в текстовые редакторы (через файл на диске фрагмент таблицы СЧМ включается, например, в текст отчета или статьи). Пакет передан в ТНУЦ и поставляется с III квартала 1987 г.

Первая система управления базами данных на «Агате» — архивно-информационная система АИСТ — была написана на Рапире Сергеем Гавриленко, когда он учился в IX классе. Она может успешно применяться при обучении основным понятиям, используемым при работе с СУБД, и используется в качестве элемента программного обеспечения курса подготовки руководящих и педагогических кадров народного образования. АИСТ обеспечивает создание и заполнение базы данных, формирование запроса на поиск и подготовку результатов поиска в виде таблицы. Производственное использование АИСТА ограничено малыми возможностями редактирования базы и недостаточной плотностью хранения данных на диске, тем не менее есть примеры эксплуатации этой программы для хранения экспериментальных данных большого объема. В Волгоградском политехническом институте разработана СУБД РАДУГА, обладающая полным набором функций реляционной базы данных и достаточной плотностью хранения (на одну сторону диска помещается база данных объемом в несколько тысяч записей). Широкому внедрению ее препятствует несовершенство организации диалога с неподготовленным пользователем и подготовки выходного документа. Работы по РАДУГЕ продолжаются как в ВПИ, так и на ЛЭМЗ, передача того или иного варианта СУБД в ГосФАП запланирована на конец 1987 г.

Программы учебного характера. На Всесоюзной летней школе 1986 г. (Новосибирск) распространялся диск с набором учебных программ, содержание которого аннотировано в файле КАК ЗАПУСКАТЬ ПАКЕТЫ, текст которого приведен ниже.

Н. А. Юнерман проводит в школе № 166 Новосибирска апробацию некоторых из этих пакетов в учебном процессе. Следует отметить принципиальную важность этапа апробации в разработке педагогических программных продуктов: сейчас еще не сложились устойчивые требования к учебным программам массового применения, и именно апробация вновь создаваемых учебных программ дает материал для выработки

ПАКЕТ ХИМИЯ: ЗАПУСК ХИМИЯ;

ЛАБОРАТОРНАЯ РАБОТА ПО ТЕМЕ КАЧЕСТВЕННЫЙ АНАЛИЗ

ПРОГРАММА СОРТИРОВКА: ЗАПУСК СОРТИРОВКА;

ДЕМОНСТРАЦИЯ МЕТОДОВ СОРТИРОВКИ МАССИВОВ

ПАКЕТ ПРОИЗВОДСТВО СЕРНОЙ КИСЛОТЫ: ЗАПУСК СЕРНОКИСЛОТ;

ДЕМОНСТРАЦИЯ И КОНТРОЛЬ ЗНАНИЙ (КУРС ХИМИИ)

ПАКЕТ ГЕОМЕТРИЯ ДЛЯ МАЛЫШЕЙ: ЗАПУСК ГЕОМЕТРИЯ;

ЗНАКОМСТВО С ГЕОМЕТРИЧЕСКИМИ ФИГУРАМИ

ПАКЕТ ОБРАБОТЧИК: ЗАПУСК ОБРАБОТЧИК;

ОБРАБОТКА РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТА (ПОСТРОЕНИЕ ГРАФИКОВ, АППРОКСИМАЦИЯ)

ПАКЕТ АЛГЕБРА: ЗАПУСК АЛГЕБРА;

ИССЛЕДОВАНИЕ ФУНКЦИЙ И ПОСТРОЕНИЕ ГРАФИКОВ (9 КЛАСС)

ГРАФИЧЕСКИЙ РЕДАКТОР: ЗАПУСК ПГП;

ПОДГОТОВКА ГРАФИЧЕСКИХ ПРОГРАММ

ПАКЕТ ПОЛИТЭКОНОМИЯ: ЗАПУСК ПЭК;

ЛЕКЦИЯ И ПРАКТИЧЕСКОЕ ЗАНЯТИЕ ПО ТЕОРИИ ПРИБАВОЧНОЙ СТОИМОСТИ

ПАКЕТ ИСТОРИЯ: ЗАПУСК ИСТОРИЯ;

ИСТОРИЯ СССР — ДАТЫ, СТОЛЫПИНСКАЯ АГРАРНАЯ РЕФОРМА

ПАКЕТ РУССКИЙ ЯЗЫК: ЗАПУСК РУССКИЙ ЯЗ;

СПРЯЖЕНИЕ ГЛАГОЛОВ (УПРАЖНЕНИЯ), УДАРЕНИЕ (ИГРА)

32

таких требований. Важно не довольствоваться первыми успехами — новизна и престижность компьютера обуславливают эйфорию при первом знакомстве даже с тривиальными программами типа перелистывания учебника на экране. В качестве основных сложностей, проявляющихся после ослабления начальных восторгов, следует упомянуть отвлечение учеников от изучаемого предмета на компьютер, стремление «обмануть» средства контроля знаний (на это нередко расходуют намного больше усилий, чем требуется для получения этих знаний), непривлекательность учебных программ по сравнению с играми, которые быстро проникают в школы, субъективное ощущение утраты (а иногда и объективная потеря) контакта с учениками у учителя.

Полное перечисление учебных программ «Агата» даже без аннотаций выходит за рамки данной статьи.

Упомянувшийся комплект программного обеспечения для 36-часового курса подготовки кадров народного образования довольно подробно, хотя и не беспристрастно, описан в [7]. Отметив высокий накал самокритики (один из авторов участвовал в формировании этого комплекта до своего ухода из НИИ ШОТСО) данной статьи и представив читателю оценку ее литературных

находок, исправим некоторые неточности.

1. Все четыре системы («Школьница», «Бейсик», «Агат-автор» и СЧМ) совместимы по текстовым файлам, специализация операционных систем вызвана стремлением исключить лишние функции для облегчения знакомства с каждой из них (одну операционную систему класса RSX изучить труднее, чем все четыре системы «Агата»).

2. «Шашки» и «Шахматы» на «Агате» — разработки полностью оригинальные, алгоритмы игр и их реализация принадлежат М. В. Левину и В. Е. Бруну (если мне предъявят аналогичную зарубежную программу, я берусь доказать, что адаптация выполнена с «Агата», но не наоборот). «Кубик Рубика» — работа А. Н. Филиппова, даже похожих программ на «Apple» не встречалось.

3. «Агат-автор» написан В. А. Цалиевым и мной. «Apple Writer», конечно, тоже готовит тексты, тем же свойством обладают «Word star», «Final Word» (их называли в качестве возможных прототипов видевшие «Агат-автор» люди, думавшие, что внутри «Агата» INTEL-процессор) и другие программы. СЧМ — адаптация, выполненная А. Н. Андроновым с «Visicalc 2», копией не является, поскольку превосходит оригинал по объему пользовательской па-

мости и доступности диалога. Вообще, уверенность в невозможности для себя разработать хотя бы среднюю по размеру систему не должна автоматически распространяться на других.

4. Исполнитель «Шлюзик» в комплект не включен, ППС «Химлаб» в природе отсутствует (видимо, дважды упомянута «Химия»).

Эксплуатация

Проблемы ремонта не существует только у тех технических средств, серийный выпуск которых еще не начат. «Агат» уже настолько стар, что на повестке дня вопросы не только гарантийного, но и послегарантийного обслуживания.

Гарантийный ремонт «Агатов» выполняют предприятия-изготовители, представители их ремонтных участков выезжают на места эксплуатации. Сроки прибытия ремонтников нередко вызывают нарекания, однако следует учитывать, что, например, 15 мастеров ремонтного участка ЛЭМЗ обслуживают около 6000 ЭВМ, размещенных по всей стране (1/6 земной суши). Для более оперативного и качественного ремонта полезно при вызове сообщать предполагаемую причину неисправности — это позволяет снабдить ремонтника подходящим комплектом ячеек на замену. Вообще «держатель» кабинета ВТ должен вырабатывать чутье на неисправности, учиться «предъявлять» их, чему немало способствует общение с опытным ремонтником. Некоторым приемам самопомощи (например, блокировка «залипшего» контакта в клавиатурной кнопке подкладываем бумажки) научились практически все.

Послегарантийный ремонт поручено осуществлять ремонтным предприятиям производственных объединений ГКВТИ

В настоящее время происходит также передача им полномочий по гарантийному ремонту. На сентябрь 1987 г. ЛЭМЗ имеет договоры на поставку ремонтных комплектов с 24 производственными объединениями ГКВТИ, оформление договоров с остальными объединениями продолжается.

Нередко удается привлечь к сопровождению кабинета инженера-электронщика из числа родителей или шефов, и тогда возникает вопрос: где достать схему машины? Электрические схемы в составе ремонтного комплекта документации централизованно размножены ГКВТИ и высылаются по заявкам своих предприятий.

По некоторым послегарантийным машинам еще не проведены доработки электрической схемы, избавляющие от мучений с дребезгом клавиш (клавиатура, конец 1985 г.) и повышающие сбойную надежность (модули процессора и контроллера диска, начало 1986 г.). Извещения на эти доработки переданы в ГКВТИ и должны поступить на ремонтные предприятия.

33

Новые разработки

В текущем году будут предъявлены на испытания новые аппаратные и программные средства ПЭВМ «Агат», в том числе:

модуль межмашинного обмена, обеспечивающий возможность построения локальной сети из «Агатов» (до 64 узлов) с удалением до 1000 м и пропускной способностью 62 тыс. бит в секунду;

комплект накопителя повышенной емкости — дисковод (133 мм, две стороны) с контроллером, на двух сторонах обычного диска размещается 840К байт;

модуль видеогенератора «Apple II», обеспечивающий работоспособность большинства программ этой ЭВМ;

модуль программатора ПЗУ и другие интерфейсные ячейки.

Будет выполнен переход на 4-регистровый знакогенератор (на экране появятся строчные русские и латинские буквы). Аппаратные разработки выполняются в виде комплектов, продающихся владельцам ПЭВМ «Агат».

Программная поддержка сети и накопителя повышенной емкости выполняется прежде всего в системах программирования Бейсик и Рапира, передача версии Бейсика в ГосФАП планируется на конец 1987 г. Запланирована также передача в серийное производство СУБД, в комплекте модуля Apple-видеогенера-

тора возможна поставка UCSD-реализации Паскаля.

Большое внимание уделяется повышению надежности ПЭВМ; разработаны два новых варианта клавиатуры — с емкостными и герконовыми замыкателями, переход на черно-белый монитор избавляет от большинства отказов по вине цветного телевизора. Отличную надежность демонстрирует комплект накопителя на ГМД повышенной емкости — при работе с ним забываешь, что такое «ГРРРР», предвещающее «ошибку ввода/вывода», и что такое несовместимость дисководов.

Разработана одноплатная модификация «Агата» с Apple-режимами памяти и экрана, более дешевая и надежная, с большей памятью. Вопрос с запуском ее в серийное производство пока не решен (будем решать его осенью**, см. ниже), отрасль просвещения изрядно насыщена перспективами появления перспективных ПЭВМ.

Нас еще не закрыли

В истории разработки системы «Школьница» заметную роль сыграла аббревиатура JNO (произносится *джно*, произошла от метки JumpNaОшибку), обозначавшая гибрид приветствия, проклятия и нелестного эпитета. Эта смесь довольно точно описывает отношение автора к соболезнующе-удивленному, а иногда и злорадному вопросу, часто всплывающему на различного рода сборищах при встрече с давно не видевшими меня знакомыми: «А вас еще не закрыли?»

** Статья подготовлена в августе.— *Примеч. ред.*

Закрытие «Агата», как показывает опыт, явление сезонное, достигающее максимума в июле — августе (1984 г.: разгромный отзыв О. Ф. Титова и В. Ф. Топоровского с летней школы, никаких надежд на получение дисководов из Болгарии; 1985 г.: увольнение главного конструктора, приказ о переводе коллектива в НИИсчетмаш; 1986 г.: широкое обсуждение вопроса, прекратить только поставки в просвещение или производство вообще; 1987 г.: запрет поставок по результатам инспекции Госстандарта; 1988 г.: еще неизвестно что). Полное отсутствие закрытия, как и следовало ожидать, приходится на январь — февраль, причем, если весенний период проходит относительно спокойно, осенью следует опасаться неожиданных демонстраций у очень высокого начальства.

Литература

1. Benchmark characteristics of IBM PC // Personal Computer World. 1984. № 8.
2. Система программирования «Бейсик» для персональных ЭВМ «Агат». Фг. 00012-01. Таллин: ТНУЦ, 1986.
3. Пакет прикладных программ автоматизации школьного учебного процесса «Школьница»: Руководство программиста. М.: НИИ ШОТСО, 1985.
4. Пакет прикладных программ автоматизации школьного учебного процесса «Школьница». 3533847.00042-01. Таллин: ТНУЦ, 1987.
5. Система подготовки текстов «Агат-автор». Фг. 00017-01. Таллин: ТНУЦ, 1987.
6. Система численного моделирования для персональных ЭВМ «Агат». Фг. 00012-01. Таллин: ТНУЦ, 1987.
7. Рыба ли рак?/Иглицкий А., Крылова Е., Луцкий В., Петров М. // Информатика и образование. 1987. № 6.

Технические средства диалога с персональным компьютером

Для кого предназначен ПК? В первую очередь для пользователя, который не является профессиональным программистом — такой ответ дают большинство специалистов по информатике. Отсюда вытекает актуальность проблемы обеспечения эффективного диалога.

Почему эта проблема не была решена раньше?

До недавнего времени эксплуатировались не прикладные системы, а отдельные программы с весьма небольшим набором функций. Средства диалога назывались (с оттенком пренебрежения) сервисом. Действительно, не так уж важно, в каком формате ввести, скажем, дату своего рождения в программе расчета биоритмов, но в таком деле, как ведение банка историй болезни, удобство представления данных и манипулирования ими приобретает первостепенное значение. Словом, из «оракула», печатающего закодированные ответы, компьютер трансформируется в инструмент для повседневной работы наподобие телефона или скоросшивателя. Возникло естественное требование, чтобы этот инструмент был удобным, чтобы для его освоения требовался минимум специальных знаний.

Попытаемся рассказать о современных требованиях к средствам диалога и сегодняшнем состоянии этих средств. Несмотря на многообразие каналов, по которым человек может получать информацию от компьютера (от музыки до тактильных ощущений), ограничимся традиционным экраном компьютерного дисплея.

Аппаратные возможности

Сегодня внешний облик персонального компьютера неотделим от «телевизора» и плоской панели клавиатуры. А ведь сравнительно недавно самым распространенным устройством, через которое человек обменивался сообщениями с

ЭВМ, была электрическая пишущая машинка.

Немного истории. Несмотря на совершенно иные по сравнению с машинкой возможности, «телевизор с клавиатурой» (иначе — дисплей) вначале имитировал машинку. Экран служил всего лишь «электронным листом бумаги», на котором «печатались» литеры. Некоторые параметры дисплеев и сейчас определяются историей развития вычислительной техники. Например, количество символов в строке, у многих моделей равное 80, унаследовано от 80-колонной перфокарты.

А что теперь? Стало ясно, что изобразительные возможности ЭВМ при появлении дисплея перешли в новое качество и произвели настоящую революцию в области организации диалога. Разработчики поняли, как нужно использовать эту «бумагу» — бумагу, на которой можно мгновенно создавать изображения и, главное, с которой можно мгновенно стирать. Их можно сравнить с учеником, который сменил тетрадку и ручку на доску, мел и тряпку.

Современный дисплей зачастую специализирован, т.е. отходит от «телевизионно-перфокарточных» стандартов. Например, в полиграфических системах экран вытянут не в ширину, а в высоту и повторяет пропорции листа писчей бумаги.

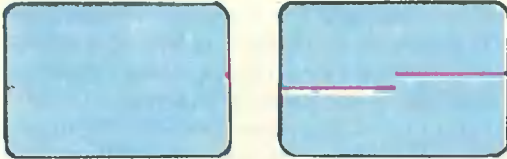
Заканчивая сравнение дисплея и пишущей машинки, приведем один курьез. Среди продукции фирмы «Тэнди» есть программа, которая проводит курс обучения десятипальцевой слепой печати, изображая при этом на цветном экране каретку с заправленным в нее листом и весьма искусно имитируя звук удара синтезатором.

Как устроен дисплей? Оставив в стороне физические принципы, на основе которых получается изображение, можно сказать, что экран дисплея современных ПК формируется из точек, состав-

ляющих матрицу $K \times M$. Чем больше K и M , тем меньше искажений в изображении. Специалисты в области компьютерного диалога называют эту матрицу растром, хотя это несколько расходится с телевизионной терминологией.

Прямоугольный растр — не единственно возможный и не всегда самый удачный. Наиболее заметные искажения он вызывает при попытке изобразить прямую с малым углом наклона к вертикали или горизонтали; в худшем случае она превращается в две точки — у краев экрана или в два параллельных отрезка (вертикальных или горизонтальных) со «ступенькой» в центре (см. рис.). Из геометрии известны типы так называемых покрытий плоскости правильными многоугольниками. В частности, плоскость покрывается системой правильных шестиугольников, центры которых образуют «шестиугольный» растр; в нем точки более «плотно» упакованы и ряд геометрических фигур изображается с меньшими искажениями.

36



Что будет, если показать машинописную страницу по телевизору? Дисплей с обычной телевизионной трубкой обычно обеспечивает 512×256 точек растра, т.е. 256 строк и 512 столбцов. Много это или мало? Подсчитаем, какое количество знаков — букв или цифр — может уместиться в строке дисплея со стандартной телевизионной трубкой. Отдельный знак сравнительно сносно изображается при помощи матрицы 7×9 . Используя одну позицию под межбуквенный промежуток, мы получаем размер строки, равный 64 знакам. Всего таких строк — 21 (под межстрочный промежуток мы выделили 3 позиции). Но нас остановят медики-гигиенисты. Текст, в котором интервал между литерами меньше одной трети их средней высоты, а интервал между строками меньше этой высоты, долго читать с дисплея опасно — портится зрение. Взяв требуемые промежутки, мы получим уже 51 знак в строке при 14 строках. Таким образом, дисплей со стандартной телевизионной трубкой сильно «недотягивает» по емкости до машинописной страницы. На практике же выпускаемые

у нас бытовые телевизоры не способны обеспечить и такой емкости из-за низкой стабильности изображения и краевых эффектов. Чтобы не слишком утомлять глаза, приходится укрупнять знаки вдвое как по вертикали, так и по горизонтали (подобная возможность предусмотрительно заложена в БК-0010).

Какой нужен растр? Дисплей 15ИЭ-00-013, входящий в состав моделей ДВК, имеет более качественный кинескоп, чем у обычного телевизора: точек растра 640×300 . Но оптимальны ли эти параметры? Из наблюдений на выставках автор вынес впечатление, что зрительный комфорт при обработке графической и текстовой информации начинается с матрицы 1000×1000 , однако эти размеры остаются пока, к сожалению, привилегией дорогих специализированных систем.

Как построить изображение на экране? В дисплее, как правило, выделено 2 режима работы: текстовый и графический. В текстовом режиме экран можно рассматривать как матрицу, на пересечении строки и столбца которой находится знакоместо — прямоугольный участок, где может быть высвечен знак. Все такие прямоугольники одинаковы. Расстояния между соседними знакоместами (межзнаковые промежутки) равны, одинаковы и межстрочные промежутки.

Знакоместо позволяет высветить ровно один знак — никакие наложения, как на печатающих устройствах, невозможны. Например, если бы мы захотели подчеркнуть слово в тексте или проставить ударения, пришлось бы освободить верхнюю и нижнюю строки и помещать в них соответствующие знаки. Легко сообразить, что подчеркивание при этом превратится в черту не под знакоместом, а над ним, соответственно и ударение станет больше похоже на запятую.

Набор всех знаков, которые могут быть высвечены, составляет алфавит данного дисплея. Каждому знаку присвоен цифровой код. Аппаратный компонент, ответственный за перевод этого кода в точечное изображение, называется знакогенератором. Знакогенераторы по-разному относятся к заложенному в них алфавиту. Некоторые «разрешают» программисту его менять, в других он, как говорится, «запаян», в частности в 15ИЭ-

00-013. Как и на печатающих устройствах, непременным элементом каждого алфавита служит так называемое *корыто* □ («пустое» знакоместо), применяемое для изображения интервалов внутри текста.

В графическом режиме программист управляет высвечиванием каждой точки растра. Впрочем, в последних моделях ПК выдачу графики облегчает так называемый видеопроцессор, который манипулирует целыми «кусками» растра.

Промежуточное положение между графическим и текстовым режимами занимает так называемая псевдографика. Ее элементы представлены, в частности, в БК-0010. Псевдографика позволяет строить простейшие многоугольники, чертить разного рода таблицы и диаграммы. Псевдографические элементы бывают строчные и межстрочные (межзнаковые). Знакогенератор БК-0010 снабжен строчными элементами. Сюда входят вертикальная и горизонтальная черты, а также набор разного рода «уголков». В этих знаках, в отличие от обычных, заполнен и межзнаковый (межстрочный) промежуток, что позволяет образовывать из них непрерывные линии. Межстрочные (межзнаковые) элементы, как явствует из их названия, располагаются на экране только в межстрочном (межзнаковом) промежутке и образуют прямоугольные контуры вокруг текста. Именно с их помощью, кстати, удобно осуществлять подчеркивание и проставлять ударения.

Сложно ли нарисовать на дисплее географическую карту? Вообще говоря, во взаимодействии текста и графики немало проблем. Высвечивать текст не в системе знакомест, а в произвольной позиции растра могут знакогенераторы лишь в дорогих системах. Еще сложнее обстоит дело с поворотом надписи на некоторый угол. Поэтому изображение на современном массовом компьютере, скажем, географической карты представляет для программиста трудную задачу.

Цвет и яркость. Теперь более подробно рассмотрим понятие «высвечивание», которым мы пользовались. Точка растра, вообще говоря, может быть не только в состояниях включено — выключено.

У черно-белого изображения ЭВМ мо-

жет регулировать яркость. Например, в БК-0010 предусмотрено 2 градации яркости. Цветное изображение создается при помощи так называемой палитры, в которую входит обычно около 16 цветов (заметим, что разные тона одного цвета считаются самостоятельными цветами), выбранных из значительно большего количества потенциально допустимых цветов (обычно их 256).

То, что палитра позволяет высвечивать лишь небольшую часть потенциально возможных цветов, объясняется ограниченным объемом памяти, используемой для хранения изображения. Дело в том, что для этого служит точно такая же память, как и в самом компьютере, т. е. длинная цепочка байтов. Шестнадцать цветов кодируются как раз четырьмя двоичными разрядами — полубайтом. Для хранения информации о цвете каждой из 512×256 точек нам понадобится, следовательно, не такой уж маленький объем — 64К байт. Напомним, что именно такова доступная для программиста память в ДВК-2М. Существуют, правда, более тонкие методы хранения изображений. Например, информация о цвете может храниться не для одной точки, а для некоторой области. Но здесь возникают трудности, связанные с ограничениями на форму такой области и ее минимальный размер.

Как выделить на экране элемент изображения? Поговорим теперь о средствах выделения элементов изображения, которыми располагает новый носитель информации — дисплей. В рукописи, для того чтобы выделить участок текста, фактически нет иного средства, кроме подчеркивания. В рукописях В. И. Ленина некоторые особо важные места подчеркнуты трижды и даже четырежды; полиграфия реже использует подчеркивание, заменяя его разрядкой, набором более крупным или более жирным шрифтом. На дисплее есть несколько основных средств выделения: оконтуривание; изменение яркости и/или цвета; инверсия; мерцание.

Под оконтуриванием подразумевается построение около выделяемой области экрана какой-либо геометрической фигуры. Самый простой пример оконтуривания — то же подчеркивание. Об-

ласть может быть заключена в окружность или перекрестие, рядом с ней может находиться указывающая на нее стрелка.

Увеличение яркости и изменение цвета представляют собой некоторый эквивалент печати жирным шрифтом.

Под инверсией понимается обмен средств изображения объекта на средства изображения фона. Например, если знак был светящимся на темном фоне, то, будучи инвертирован, он окажется темным на светлом фоне.

Мерцание означает периодический переход какого-либо участка экрана из одного состояния в другое, а затем опять в первоначальное. Например, знак может появляться и исчезать с определенной частотой.

38

На практике мерцание часто сочетается с каким-либо из трех других способов выделения. Таким комбинированным способом является периодическое инвертирование знака или мерцающее перекрестие. Периодическое изменение яркости встречается реже.

Подвижный указатель (курсор). Проектировщик диалоговых систем должен дать в руки пользователю эффективный способ указать на экране объект, над которым будет производиться операция (например, указать слово в тексте, которое нужно удалить). Такое подвижное средство выделения (подвижный указатель) называют курсором, или маркером.

Во всех дисплеях предусмотрен так называемый аппаратный курсор, указывающий на знак или на точку. В 15ИЭ-00-013 курсор изображается в виде мерцающей черты под знаком, в БК-0010 представляет собой инвертированную зону экрана или же перекрестие. Бывают и неудачные варианты. Так, в польском дисплее «Мера» курсор — это просто засвеченное знакоместо, так что он закрывает знак, на который указывает. Аппаратный курсор родился все из той же имитации пишущей машинки и первоначально обозначал место, где будет «напечатана» нажатая на клавиатуре литера.

Как «воздействовать» на компьютер? Теперь от устройства отображения — экрана дисплея — перейдем к описа-

нию устройств, через которые компьютер получает от нас информацию. Кстати, нужно отметить: экран и эти устройства в современных компьютерах функционируют независимо. Например, буква, введенная с клавиатуры, отобразится на экране, только если существует программа, которая примет эту букву от клавиатуры и пошлет ее на экран. В таблице приведена некоторая классификация устройств ввода информации с примерами из бытовой техники.

Устройства ввода информации

	Дискретные	Непрерывные
Без фиксации положения	Наборный диск телефона	Звонок в квартире (какая угодно продолжительность)
С фиксацией положения	Переключатель программ в телевизоре	Регулятор громкости (плавное изменение с фиксацией силы звука)

Попробуем найти в ней место для клавиатуры дисплея. Ее алфавитно-цифровая часть попадает в левый верхний угол таблицы. Дискретность сигнала связана с тем, что можно передать только небольшой набор знаков. Нажатая клавиша не фиксируется, при отпускании она переходит в нейтральное положение.

Все устройства могут быть реализованы самыми разными способами в зависимости от вида информации, для обработки которой они предназначены. Например, для ввода текста используется клавиатура, подобная клавиатуре обычной пишущей машинки. Ввод чертежа в системах автоматизированного проектирования осуществляется устройством «сколка»: это подвижный визир с перекрестием, перемещаемый по специальному планшету, на котором находится чертеж. Пример устройства ввода команд — так называемая функциональная клавиатура. Самая распространенная функциональная клавиша — это клавиша «закончить ввод». Команда в принципе может подаваться и голосом. Словом, разнообразие технических решений велико, однако их использова-

ние ограничено стоимостью массового компьютера.

Среди средств «воздействия» особое место занимают манипуляторы — устройства, позволяющие передать в компьютер направление и скорость движения руки. Наиболее популярны световое перо, планшет, джойстик, «мышь», трек-болл, «кот».

Световое перо позволяет указывать объект так же, как учитель делает это указкой на доске.

Планшет — это «слепой экран» со своим «слепым пером», лежащий на столе перед пользователем. Движения пера по планшету преобразуются в движение какого-либо объекта на экране.

Джойстик представляет собой рычажок на шаровой опоре, которая позволяет вращать его во всех направлениях. Неудобен короткий ход рычага, из-за этого необходим возврат в нейтральное положение при достаточно длинном перемещении объекта. Поэтому компьютер получает сигнал лишь о поступательном движении джойстика. Его в основном используют для «толчка», задания вектора скорости.

«Мышь» — это коробочка, которую пользователь катает (двигает) по любой поверхности. Вмонтированный шарик или система колес определяет направление сдвига. «Мышь» удобнее светового пера, так как рука пользователя может лежать на столе, удобнее джойстика, так как ее «ход» ограничен только размерами поверхности, однако «мышь» проигрывает им в тонких движениях, например с ее помощью трудно провести ровную линию.

Трек-болл — это перевернутая «вверх брюхом» «мышь». Он представляет собой своеобразный глобус, который может вращаться вокруг любой оси. Рука пользователя лежит на верхней части «глобуса». Понятно, что трек-болл тем удобнее, чем больше его диаметр. Недостатком трек-болла является его относительная громоздкость и необходимость постоянных возвратно-поступательных движений руки.

«Кот» используется так же, как трек-болл, но в сравнении с ним значительно компактнее. Обычно это слегка выпуклый диск, вмонтированный в клавиатуру,

обтянутый эластичной «шкурой», которая легко скользит по нему в любом направлении. Возможен и сенсорный вариант этого устройства.

К манипуляторам относятся также различные дорогие устройства, позволяющие работать в модели трехмерного пространства. Они применяются преимущественно в системах автоматизированного проектирования (САПР).

Расширение аппаратных возможностей при помощи программ

Узнав, что позволяет делать аппаратура персонального компьютера для отображения и ввода информации, зададимся вопросом: как же использовать ее наилучшим образом? Прежде всего напишем ряд основных ограничений, накладываемых аппаратурой.

1. Самое «парадоксальное» ограничение: компьютер у нас всего один. А хотелось бы иметь несколько (прервать работу на одном и пересесть за другой, если скажем, что-то срочно потребовалось, или сравнить два текста, выведя их на экраны).

2. Изображаемая картинка может не поместиться на экране.

3. Клавиатура не позволяет изображать подробные пояснения около каждой клавиши и оперативно менять эти пояснения в зависимости от контекста (клавиш мало, а команд много).

4. Изображение никак не структурировано («в каждой строчке только точки»).

5. Устройства ввода не ставят заслона неверной или бессмысленной информации.

Указанные недостатки можно попытаться компенсировать программно.

Полиэкранный. Программа (а точнее — целая программная система) может моделировать несколько компьютеров. При этом физический экран разбивается на несколько окон; каждое окно — это экран моделируемого компьютера. Окна могут изображаться рядом либо перекрывать друг друга наподобие листов бумаги, лежащих на столе. Этот вариант даже лучше, чем независимые физические компьютеры, так как между окнами возможен оперативный обмен

информацией. Полиэкранный позволяет, оставив работу в некотором окне, затем возобновить ее с того же места.

«Плавающее окно» и «плавающие форточки». Большую картинку можно просмотреть путем организации так называемого *плавающего окна*, напоминающего окно автомобиля: картинка как бы проплывает мимо него. Некоторые картинки полезно просматривать с помощью нескольких взаимосвязанных окон, которые иногда называют «форточками» (например, в одной из «форточек» можно держать часть боковика таблицы, относящуюся к информации из другой «форточки»). Система должна предупреждать о том, есть ли за границей окна или «форточки» невидимая пока информация. Обычно это делают, изменяя положение соответствующей границы. Неплохо иметь возможность увидеть (в масштабе) положение «плавающего окна» на изображении.

Окно может «плыть» не только по участку плоскости. Например, если вслед за верхним краем картинки мы оказываемся на нижнем, то становится ясно, что мы движемся по цилиндру.

«Меню». Описания задаваемых клавишами команд можно вынести на экран. Но как «нажать» нарисованную клавишу? Есть несколько способов. Например, можно высветить пояснения в нижней части экрана, а под экраном расположить клавиши в ряд. Но наилучшее решение проблемы — с помощью манипулятора. Световое перо может непосредственно указать необходимое обозначение (надпись); другие манипуляторы могут оперативно переместить в нужное место курсор, а кнопка «выбор» осуществит «нажатие» выделенной таким образом псевдоклавиши. Набор изображенных на экране псевдоклавиш разработчики назвали *меню*. Подобно клавиатуре, меню бывают двух типов. Эти типы соответствуют клеткам первого столбца (см. табл.). Различия хорошо прослеживаются на примере панели дисплея 15ИЭ-00-013. Алфавитно-цифровая клавиатура не допускает нажатия одновременно двух и более клавиш, нажатая клавиша не фиксируется. Однако на панели есть группа клавиш, которые могут быть нажаты в любом сочетании.

Они имеют два положения (нажатое и отжатое) и могут быть зафиксированы в любом из них. Это клавиши «дуп», «лин», «ред», сочетание которых задает режим работы дисплея.

Как правило, используется целая система меню-«клавиатур».

Для просмотра большого, не помещающегося на экране, меню также применяют метод «плавающего окна». Важным частным случаем такого метода является окно, в котором помещается ровно один пункт меню. Этот прием позволяет заменить движение курсора по меню движением окна (или, что то же самое, движением меню мимо окна).

Простейшее структурирование: «трафарет». Почти в каждой картинке, изображаемой на экране, можно выделить условно-постоянную (*трафарет*) и условно-переменную части. Например, в таблице в условно-постоянную часть войдут заголовок и шапка.

Обычно трафарет защищен от изменений, он корректируется отдельно от «наполнения». Трафарет предполагает, кроме того, наличие набора блоков, из которых можно формировать условно-переменную часть. Словом, «резиновая мысль», так остроумно высмеянная Ильфом и Петровым в «Золотом тельке», оказалась полезным элементом компьютерного диалога.

Использование трафарета и «плавающих окон» в условно-переменной части позволяет организовать весьма гибкую систему просмотра и корректировки информации. По этому принципу, в частности, построен так называемый динамический электронный бланк для ускоренного просмотра базы данных, разработанный известным американским специалистом А. Кеем.

Структурный курсор. Мы рассмотрели курсор — подвижной указатель, указывающий на какой-то элемент изображения. В отличие от *аппаратного* курсора, указывающего на знак или на точку, программная система поддерживает *структурный* курсор, перемещающийся по объектам определенного вида, например на плане дома только по дверным проемам. Он незаменим, когда налицо «вложенность» изображаемых объектов, т. е. «точечное» указание на объект не-

избежно есть указание и на какую-либо его часть. Имея в своем распоряжении структурный курсор, мы можем применить к различным структурным единицам одни и те же командные клавиши.

Однако наличие структурного курсора не означает, что следует отказаться от аппаратного. Последний полезен именно тем, что «не знает» границ объектов и позволяет быстро попасть в требуемую зону экрана, если в нее не нужно слишком тщательно «прицеливаться».

Очувствление. Так разработчики называют способность устройств ввода не пропускать недопустимые или не имеющие смысла данные или команды, которую эти устройства приобретают под управлением программы. Любопытно, что возможна такая организация диалога, когда, скажем, программист при вводе программ на Паскале не в состоянии допустить ни одной синтаксической ошибки.

В применении к дисплею очувствление может означать следующее: блокировку выдачи (возможно, со звуковым сигналом), блокировку клавиатуры, торможение манипулятора.

Блокировка выдачи может, например, происходить, когда в цифровое поле мы пытаемся занести букву.

Блокировка клавиатуры, т. е. программное отключение каких-либо клавиш, реализуется редко, так как является весьма дорогостоящей (за исключением блокировки клавиатуры целиком).

Торможение манипулятора отражает ограничения на перемещения управляемого им объекта. При бедных технических средствах торможение может выражаться в замедлении движения на экране. При наличии манипулятора типа трек-болл или джойстик торможение может иметь буквальный смысл, имитируя усилия, затрачиваемые на перемещения или преобразования.

Заключение

Как применять описанные нами технические средства и приемы? В любом диалоге существует два основных способа сообщить или узнать что-либо: рассказать и показать. Например, можно

рассказать, как пройти к вам домой, а можно начертить план. Конечно, лучше один раз увидеть, чем сто раз услышать; в то же время нельзя любой разговор свести к жесту, к показу, как хотели сделать герои Свифта в «Путешествии Гулливера».

Для компьютерного диалога характерен скорее не разговор, а переписка. Используется также способ получения данных путем перемещения в некотором информационном пространстве, получивший название навигации. Навигацию иначе можно определить как особый вид доступа к информационному объекту, доступ по положению объекта относительно других. Задача разработчика диалоговых средств здесь заключается в том, чтобы по возможности адекватно изобразить на экране это информационное пространство. 41

Успехи навигационного подхода демонстрируют, в частности, популярные прикладные программы — электронные таблицы. В них данные располагаются на так называемом автобланке — прямоугольной матрице с достаточно емкими ячейками. В каждой ячейке может храниться числовая или текстовая информация. Ячейка адресуется очень похожим на шахматный способ: по горизонтали буквами А—Z, по вертикали числами. Легко могут быть определены различные операции над содержимым ячеек. При просмотре используется метод «плавающих окон» и «плавающих форточек» с индикацией текущих координат. Существуют (правда, в экспериментальном плане) системы, генерирующие изображения канцелярского шкафа, папок с бумагами и других деловых аксессуаров. В отличие от настоящего электронный шкаф легко расширяется, папки мгновенно создаются, мгновенно уничтожаются, имеют большую емкость. К тому же внутри папки может оказаться... еще один шкаф.

Навигация и переписка обязательно должны дополнять друг друга в соответствии с особенностями работы в определенной предметной области. Однако анализ предметной области и ее профессионального языка — тема другого разговора...

Система «Рига»

42

Компьютеризация образования сейчас находится в стадии поиска оптимального варианта школьной ЭВМ, состава кабинета вычислительной техники, его программного обеспечения и методики проведения занятий с использованием компьютера. Ни одна из этих проблем еще не решена. Особенно сложны в решении две последние, поскольку они связаны с рождением необычной для новой технологии обучения. Характерным здесь является необходимость комплексного подхода, учитывающего важнейшие факторы, большинство из которых существует независимо от желания специалистов, формирующих эту технологию. Распространена точка зрения, согласно которой программный продукт для школ можно получить, просто отбирая лучшие программы, разработанные в разных организациях. Как показывает опыт, таким путем удастся сформировать лишь пакеты разрозненных программ, практически непригодных для учителя.

Система «Рига» создавалась специально для обучения, как попытка комплексного подхода к проблеме его компьютеризации. При разработке системы учитывались следующие предпосылки:

техническая основа — КУВТ-86 (микро-ЭВМ ДВК-2М, соединенная с двенадцатью ЭВМ БК-0010) как наиболее массовая отечественная техника для школ;

методическая основа — централизованное управление учебным процессом во время проведения занятий, при их максимально возможной автоматизации;

программная основа — однородность программного обеспечения (языка программирования учебных предметов и выполнения операций по обмену информацией внутри КУВТ-86);

инструментальная основа — упрощенный язык программирования с возможностью его совершенствования.

При разработке системы специально исследовалась роль учительской ЭВМ, входящей в комплект вычислительной техники. Некоторые специалисты полагают,

что эта ЭВМ не нужна в КУВТ-86 и ничего не дает учителю, кроме дополнительных сложностей эксплуатации увеличения стоимости комплекта. Тем более что он до сих пор оснащался малыми средствами поддержки локальной сети, которые состоят из отдельных программ (NETVK, GETVK, PUTVK), позволяющих производить операции по пересылке программ на языке Фокал. Фактически эти средства сводят функции учительской ЭВМ ДВК-2М к замене магнитофона на диски, что и было замечено рядом разработчиков. Они предложили перейти от сети на основе ДВК-2М—БК-0010 к более простой на основе единого для всех БК-0010 магнитофона. Такой вариант вдвое снижает стоимость КУВТ и резко повышает надежность системы. Однако при этом учитель не только тратит больше времени на операции по рассылке программ, но и лишает себя возможности автоматизировать процесс обучения. Становится также недоступным и программное обеспечение, имеющееся у ДВК-2М и которое могли бы использовать способные ученики.

В системе «Рига» функции ДВК-2М существенно расширены, при одновременном упрощении работы учителя с КУВТ-86. Это достигнуто за счет введения шести режимов работы учителя с ДВК-2М;

- 1) передача данных из ДВК-2М в БК-0010;
- 2) прием данных из БК-0010 в ДВК-2М;
- 3) печать;
- 4) просмотр каталогов;
- 5) обмен (электронная почта);
- 6) сбор информации о процессе обучения.

Последние два режима возможны только при работе с файлами на Т-языке.

Упрощение работы системы достигнуто вводом единой программы обслуживания сети, работающей в режиме диалога. Запуск системы «Рига» осуществляется

после включения КУВТ-86 и установки диска с программой обслуживания сети в верхний дисковод. Для запуска нужно ввести в ДВК-2М директиву X0. Запуск системы выполняется так:

160000
 @X0
 ОС ДВК V04.00

После этого ЭВМ покажет кадр (рис. 1). В ответ на этот кадр нужно ввести номер одного из шести режимов работы. При ошибке ввода кадр повторяется. Например, после ввода цифры 1 получим следующий кадр (рис. 2).

В режиме передачи файлов из ДВК-2М возможна загрузка в БК-0010 не только файлов на Т-языке, но и текстов, подготовленных на БК-0010 с помощью специального редактора текстов и программ на языке Фокал или в машинных кодах.

Передача файлов в БК-0010 может быть индивидуальной и групповой, например, если вместо номеров БК ввести пробел, можно послать файл сразу на все включенные БК-0010. Это позволяет существенно сократить время на подготовку рабочих мест учеников. Загрузка файла сопровождается двоичным отображением его на экранах БК-0010, что позволяет контролировать процесс загрузки. При неудаче следует повторить загрузку. И иногда при этом может оказаться полезным выключить и включить БК-0010 для его инициализации.

СИСТЕМА «РИГА»	
CR: НИИ ФТТ ЛГУ, Рига, тел 261393 ВЕРСИЯ: 870612	
1 передача в БК 2 прием из БК 3 печать 4 каталог 5 электронная почта 6 параметры системы «РИГА»	
Введите номер режима >1	

1	Передача в БК 1 файл на Т-языке (тип. RIG) 2 файл-документация (тип. RED) 3 программа на ФОКАЛЕ (тип. FOC) 4 другие файлы
	Введите номер >1
	Введите имя файла > P2
	Введите номера БК (через запятую) или пробел — для всех БК >1,6, 7, 11
	идет передача в БК
	не готовы БК: 6, 11

Перед передачей файла на БК-0010 нужно прекратить работу программы, проще всего это сделать, нажав на БК-0010 клавишу «СТОП».

Если учитель забыл имя нужного файла, то он может запросить каталог в первом кадре системы «Рига». ЭВМ покажет кадр (рис. 3).

4	Каталог 1 каталог верхнего диска 2 каталог нижнего диска
	Введите номер > 2

В этом кадре можно выбрать каталог верхнего диска (системного) или нижнего (с обучающими программами). При запросе нижнего диска будет выдан кадр, изображенный на рис. 4.

Каталог	
Дискета	
имя файла:	содержание:
P1	— клавиатурный тренажер
P2	— обучение клавише ВВОД
P3	— обучение вводу цифр
P4	— обучение вводу чисел
P5	— обучение вводу букв
P6	— обучение вводу слов

Структура каталога в системе «Рига» всегда содержит имена файлов (за-

главные латинские буквы) и краткое их содержание. Нужно запомнить (выписать) имя нужного файла, ввести точку (чтобы вернуться в первый кадр) и, вызвав режим передачи файлов, послать на БК-0010 выбранный файл.

Прием файла с любого БК-0010 в ДВК-2М производится также в системе «Рига» в режиме приема файла. На рис. 5 показан пример ввода файла

44

5

Прием из БК	2
1 файл на Т-языке (тип RIG) 2 файл-документация (тип RED) 3 программа на ФОКАЛЕ (тип FOC) 4 другие файлы	
Введите номер >2	
Введите номер БК>8	
Идет прием из БК	
Введите имя файла>R4	
удалить старый файл (Д) или (Н)>Д	

Принимаемый из БК-0010 файл может быть введен первоначально либо с магнитофона, либо с ДВК-2М. Если такой файл уже есть в ДВК-2М, а его нужно заменить на новый, то ЭВМ сообщит: «удалить старый файл (Д) или (Н)». В этом случае нужно ввести букву Д — для замены старого файла или Н — для отмены ввода новой версии файла.

При вводе файла типа RIG, RED, FOC ЭВМ запросит адрес загрузки (ввода) файла в память БК-0010 и его длину. Нужно ввести соответствующие восьмеричные коды. Операция по вводу новых файлов должна выполняться очень осторожно, так как можно испортить базу данных системы «Рига».

Следующий режим работы системы «Рига» — печать файлов. Имеется в виду возможность использования КУВТ-86 для распечатки программ на Т-языке и языке Фокал, а также печать текстов, подготовленных на БК-0010 с помощью специального редактора текстов. В последнем случае, подключив устройство печати прямо к БК-0010, можно даже готовить документы в режиме прямой

копии с экрана БК-0010, т. е. со всеми спецсимволами и рисунками. Этот режим включен в каждый файл, имеющий расширение RED.

На рис. 6 показан пример работы в режиме ПЕЧАТЬ для обычной конфигурации КУВТ-86. Вначале нужно указать тип файла, который будет печататься. Далее ЭВМ запросит имя файла (в этом примере это R3). Если такого файла нет или указан неверный тип, то ЭВМ снова повторит запрос (ввод точки здесь и далее означает отказ от выполнения операции).

Найдя файл, ЭВМ уточнит формат печати. На запрос «рулон» можно ответить «ДА», если предполагается непрерывный вывод всего файла, и «НЕТ», если нужен останов после вывода каждой страницы. Ответив «ДА» на предложение нумеровать страницы, нужно задать номер первой страницы. Далее ЭВМ спросит: «печатать без пропуска строк»? по ответу «НЕТ» ЭВМ будет после каждой строки выводить одну пустую строку. Сообщение «32 символа в строке» предполагает печать файла на Т-языке, в остальных случаях нужно отвечать «НЕТ». Длина страницы в строках зависит от цели распечатки. Стандартный лист содержит примерно 63 строки отпечатанные без пропусков и 42 строки — с пропусками между строк. Этот параметр лучше всего подобрать экспериментально.

Файл, подготовленный на БК-0010,

Печать	3
1 файл на Т-языке (тип RIG) 2 файл-документация (тип RED) 3 программа на ФОКАЛЕ (тип FOC)	
Введите номер > 2	
Введите имя файла > R3	
рулон > НЕТ	
нумеровать страницы > ДА	
введите начальный номер > 1	
печать без пропуска строк > ДА	
32 символа в строке > НЕТ	
длина страницы 29 > НЕТ	
введите длину > 63	
заменять непечатные символы пробелами > НЕТ	
устройство печати включено > ДА	

6

может содержать символы, которых нет у печатающего устройства. В этом случае они будут заменены на пробел, если в ответ на предложение ЭВМ «заменять символы пробелами» ответить «ДА», или на точки, если ответить «НЕТ». В конце диалога ЭВМ спросит: «устройство печати включено»? Нужно включить его и ответить: «ДА». Затем ЭВМ начнет печать файла.

Следует иметь в виду, что различные печатающие устройства подключаются к ДВК-2М по-разному. Чтобы печать выполнялась правильно, нужно устройство установить в режим КОИ-8. В термопечати для этого нужно подпаять один провод между устройством и ЭВМ. Этот провод должен соответствовать восьмому биту выводимой информации. В других устройствах печати потребуются всего лишь переключить задатчик режима вывода.

Следующий режим системы «Рига» — «электронная почта» — предназначен для обмена информацией между учениками и учителем по инициативе учителя. Возможна посылка сообщений из ДВК-2М в БК-0010, а также прием информации из БК-0010 во время ее набора. «Электронная почта» работает только во время исполнения программы на Т-языке. На рис. 7 показан пример диалога в этом режиме. На запрос «введите номера активных БК» перечисляются номера ЭВМ, которые будут участвовать в обменах. Ввод пробела означает, что участвуют все ЭВМ. После задания номеров следует выдача сообщения из ДВК-2М о не подготовленных для связи БК-0010 и перечень ЭВМ, с которыми установлена текущая связь. Далее ДВК-2М запросит номера БК-0010, которым нужно передать сообщение, после чего возникнет текст «введите сообщение». Если в ответ ввести текст сообщения, то он появится на экранах всех БК-0010, с которыми установлена текущая связь. Например, на БК-0010 послано сообщение «НАЖМИТЕ КЛАВИШУ Q» (рис.8). Оно принято в верхней строке БК-0010.

В связи с различием отображения информации на экранах ДВК-2М и БК-0010 сообщения при обмене между ЭВМ трансформируются:

5

Электронная почта

Введите номера активных БК>1, 2, 3, 9, 12
текущая связь: 1, 2, 3, 9, 12
кому передать >
введите сообщение >
НАЖМИТЕ КЛАВИШУ Q

ДВК: НАЖМИТЕ
КЛАВИШУ Q

не готовы: 2, 3, 12

7

НАЖМИТЕ КЛАВИШУ Q

ТРЕНАЖЕР КЛАВИАТУРЫ

Версия: 12 мая 1986

СР. НИИ ФТТ ЛГУ им. П. Стучки,
Рига

Я Q

45

8

все строчные буквы заменяются на заглавные;

все спецсимволы БК-0010 заменяются в ДВК-2М на символ ■;

сообщение с ДКВ-2М уменьшается до 24 символов при отображении на экране БК-0010.

Сбор сообщений с БК-0010 происходит в том случае, если в ответ на предложение «введите сообщение» нажать только клавишу ВК (рис. 9). Для

5

Электронная почта

введите номера активных БК>
текущая связь: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
кому передать >
введите сообщение >

БК01: Я
БК02: Q
БК03: ЯФQ

БК05: ГДЕ БУКВА 'Ь' ?
БК06: 12345
БК07: Я НИЧЕГО НЕ ПОНИМАЮ...

БК10: ПОМОГИТЕ!
БК11: RIGA
БК12: 10PRINT «ВВЕДИТЕ ЧИСЛО»

не готовы: 4, 8, 9

9

тех БК-0010, с которыми нет связи, строки сообщений на экране ДВК-2М тоже пустые. Если после принятия сообщений включить печатающее устройство и ввести «ПЕЧАТАЙ», то все собранные сообщения будут отпечатаны на бумаге.

Информация с БК-0010 принимается только по инициативе учителя. Это позволяет следить за набором строк учащихся, не мешая их работе, что более продуктивно, чем обход учителем рабочих мест школьников. Электронная почта, даже в такой простой форме, позволяет автоматизировать решение многих задач. Например, можно показать учащимся, как работать с клавиатурой БК-0010.

46 Учитель рассылает на все БК-0010 файл P2 и объясняет, что клавиатура БК-0010 похожа на пишущую машинку. Она имеет строчные и заглавные буквы. Перед вводом строчных букв надо нажать клавишу СТР (она внизу пульта). Перед вводом заглавных букв надо нажать ЗАГЛ. Но, в отличие от машинки, БК-0010 имеет сразу несколько алфавитов: русский, латинский и графический. Чтобы ввести буквы русского алфавита, нужно нажать клавишу РУС. Учитель предлагает учащимся нажать клавишу с буквой Я, и они выполняют задание.

Учитель запрашивает в режиме электронной почты информацию с БК, просматривает ее и дает указания школьникам, не выполнившим задание. Аналогично объясняется назначение клавиш ЛАТ, ПР, ПР, НР. Далее учитель может показать, как «стирать» неверные символы а также, где находятся символы 0 О Ъ * □ — — , '. После этих упражнений проводится самостоятельная работа с файлами P2,..., P6, P1.

Другой пример использования электронной почты связан с автоматизацией анкетирования учащихся, педагогов, родителей. Для этого учитель через ДВК-2М или устно задает вопрос, а школьники набирают на БК-0010 ответы (не нажимая ВВОД). Учитель собирает ответы в ДВК-2М, печатает их, а затем называет следующий вопрос. В конце анкетирования получается аккуратно отпечатанный протокол, который можно напечатать при необходимости в двух-

трех экземплярах, если использовать копировальную бумагу.

Следующий пример — автоматизация проверки знания учебных предметов. Для этого в режиме электронной почты задаются вопросы, принимаются и печатаются ответы, как при анкетировании. Преимущество такого способа перед традиционным — легкость контроля при выполнении задания, простота выявления типичных ошибок.

С помощью режима электронной почты можно проводить деловые игры с заданными взаимоотношениями между участниками, ведением протокола всех обменов информацией для последующего анализа. Простым примером может быть игра в почту, когда одному ученику поручается принимать и рассылать сообщения по адресатам, указанным в начале самих сообщений.

Следующий режим системы «Рига» — отбор данных о прохождении учебного процесса во время заданий с КУВТ-86. Этот режим работает только для файлов на Т-языке. По запросу учителя можно получить до 312 параметров (26 на каждого учащегося), связанных с их работой. Это могут быть результаты ответов на вопросы (верно, ошибка), число попыток решения задачи, число затребованных справок, время на получение ответа, оценки за ответы и т. п. На рис.10 показан пример запроса параметров. Здесь буквами А, В, С, D..., X, Y, Z обозначены 26 параметров, а ЭВМ, с которых они получены, обозначены БК01, ..., БК12. Соглашения по допустимым значениям параметров следующие:

если значение лежит в пределах от 0 до 9, то оно выводится;

если значение меньше 0, то выводится символ «—»;

если значение больше 9, то выводится символ «+».

Для регистрации параметров нужно включить печатающее устройство и ввести приказ «ПЕЧАТАЙ» сразу после выдачи значений на экран. Режим сбора параметров позволяет начать исследование по созданию «электронных» журналов успеваемости, которые, в отличие от обычных, фиксируют конкретные ошибки учеников, а не общие оценки

Параметры системы «Рига»

введите номера активных БК > 2, 3, 4, 9, 10, 12

	O	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
БК02:	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0
БК03:	0	1	0	1	1	0	0	1	0	1	1	0	0	0	0	1	1	0	8	0	0	0	0	0	0	0	0
БК04:	0	1	1	0	0	1	1	1	0	0	1	0	0	1	0	1	1	0	+	0	0	0	0	0	0	0	0

не готовы БК: 9, 10, 12

успеваемости. Это позволяет выявлять структуру ошибок, находить главные ошибки и исправлять их. Обобщение подобной информации по всем учащимся данного класса позволит своевременно корректировать процесс обучения. Конечно, все это возможно лишь при наличии специальных программ анализа электронных журналов.

База данных системы «Рига»

Ценность любой обучающей системы определяется прежде всего ее базой знаний, т. е. количеством и актуальностью обучающих, тренирующих и контролирующих программ. Кроме того, нужно иметь достаточно удобный язык создания таких программ.

База знаний системы «Рига» ориентирована прежде всего на решение массовых задач компьютерного всеобуча; а именно: демонстрацию возможностей ЭВМ, обучение вводу информации в ЭВМ и элементам программирования. База знаний легко расширяется и записана на Т-языке, специально созданном для целей составления учебных материалов. Основное отличие Т-языка от известных языков программирования (Бейсик, Паскаль, Фокал...) в том, что основная информационная единица в нем — это текстовый кадр, а не оператор, как в популярных языках программирования. Тексты в системе «Рига» могут быть практически на любых национальных языках. Транслятор Т-языка включает в себя операции, характерные для обучающих процедур (разбор ответов и построение иллюстраций). Сами файлы системы «Рига» являются программно-текстовыми, т. е. содержат интерпретатор Т-языка, редактор текстов и сами учебные тексты. Это делает их независимыми от версии Т-языка и технических средств. Например, такой файл мо-

жет быть вызван с магнитной ленты, или с дискеты, либо из памяти любой ЭВМ, способной по запросу БК-0010 выдать этот файл.

База знаний системы «Рига» к середине 1987 г. насчитывала десятки файлов, представляющих общий интерес. Они хранятся на четырех дискетах.

Дискета первого контакта

Дискета предназначена для проведения двух-трех первых занятий с ЭВМ. В итоге ожидается формирование навыков работы с клавиатурой БК-0010 и получение общих представлений о возможностях ЭВМ и программировании. Ее содержание совпадает с перечнем, приведенным в статье (1). Лишь файл «ЛАТВИЯ» заменен на «ГОРОДА СССР», а к файлу «РОБОТ» добавлено два файла: описание языка исполнителя и задачник к файлу «РОБОТ». На этой же дискете есть документация по работе с системой «Рига»:

R — правила работы с КУВТ-86.

R — изменения в Т-языке.

R — методика работы в системе «Рига».

R — методика работы с дискетой первого контакта.

R — методика работы с дискетой обучения языку.

Дискета обучения языку Бейсик

Дискета предназначена для полной или частичной автоматизации процесса обучения языку Бейсик. Это вызвано желанием помочь тем школам, где опытных учителей информатики нет. Дискета обеспечивает минимум знаний по информатике. Набор файлов полностью отражает основные формы обучения: урок, упражнение, контроль и решение задач.

Кроме того, имеется простой транслятор, позволяющий вводить в БК-0010 программы на самом популярном языке микро-ЭВМ. Продолжительность работы с дискетой десять—пятнадцать уроков, в зависимости от числа программ, составляемых учащимися. Состав дискеты:

P11 — программированный урок по основным операторам Бейсик, PRINT, INPUT, LET. Урок рассчитан на 20 минут индивидуальной работы с ЭВМ, во время которой учащийся получит основные сведения об операторах, достаточные для составления линейных программ.

Урок основан на непрерывном контроле понимания учебного материала

P12 — тренажер на запоминание понятий, данных в уроке № 1.

P13 — тест знаний по материалу урока № 1.

48 P14 — сборник задач для практического закрепления понятий урока № 1.

BASIC — транслятор языка Бейсик (разработка Вильнюсского университета) для ввода и запуска программ.

P15 — ответы и методические указания по решению задач из P14

P16 — программированный урок № 2 по операторам GO TO, IF THEN, GOSUB, RETURN.

P17 — программированный урок № 3, объясняющий понятия массива, индекса, ввода, вычисления и вывода элементов массива.

P18 — тренажер на понятия уроков № 2 и 3.

P19 — тест знаний на понятия уроков № 2 и 3.

P20 — сборник задач для практического закрепления понятий уроков № 2 и 3.

P21 — информация о диалектах языка Бейсик и транслятора Бейсик разработки Вильнюсского университета.

Дискеты демонстрации возможностей ЭВМ

Дискеты предназначены для проведения занятий по использованию ЭВМ при решении наиболее популярных задач: подготовка текстов, табличные расчеты, ввод и поиск информации, а также программированное обучение. Состав дискет:

P30 — описание редактора текстов системы «Рига»

P31 — лабораторная работа по редактированию текстов.

P32 — задачник по редактированию текстов.

P33 — описание учебной базы данных «ЕВРОПА».

P34 — задачник по базе данных «ЕВРОПА».

P35 — учебная база данных «ЕВРОПА».

P36 — учебная база данных «КЛАСС».

P37 — описание учебной электронной таблицы «РАСЧЕТ».

P38 — учебная электронная таблица «РАСЧЕТ».

P39 — описание учебного редактора графической информации «ГРАФИК» и задачника к нему.

P40 — учебный редактор графической информации «ГРАФИК».

Примеры программированных занятий:

P41 — по музыке.

P42 — по биологии.

P43 — по химии.

P44 — по истории.

P45 — по литературе.

P46 — по физике.

P47 — по алгебре.

P48 — по геометрии.

P49 — по астрономии.

P50 — по русскому языку.

P51 — по английскому языку.

P52 — по немецкому языку.

P53 — по арифметике.

P54 — по физкультуре.

P55 — по географии.

P56 — по этике и психологии.

P57 — по военной подготовке.

P59 — по основам Советского государства и права.

Система «Рига» является попыткой комплексного подхода к решению задач компьютеризации обучения. Учительская ЭВМ в системе «Рига» не только выполняет роль хранилища файлов, но и помогает автоматизировать ряд операций учителя во время урока.

База знаний системы «Рига» позволяет решать основные задачи информатики и показывает возможность развития программированного обучения по школьным предметам.

Предполагается дальнейшее совершенствование системы «Рига». Это развитие Т-языка, расширение функции автоматизации труда учителей, увеличение базы знаний системы.

Литература

Кузьмин Ю. Я. Т-язык. // Информатика и образование. 1987. № 2. С. 64—70.

Лабораторные работы на ДВК-1

По программе и учебным планам курса «Основы информатики и вычислительной техники» на II курсе СПТУ отводится 22 ч на лабораторно-практические работы с ЭВМ. В кабинете информатики нашего профтехучилища установлены десять ДВК-1М, поэтому лабораторно-практические работы написаны с учетом возможностей интерпретатора языка Бейсик для этих ЭВМ. Здесь представлено 11 лабораторных работ по 2 ч. Каждая работа состоит из двух частей. В первой дается теоретический материал, подкрепленный готовыми программами. Во второй предусмотрена самостоятельная работа учащихся по составлению программ, их редактированию, исполнению и анализу результатов.

Цели первой части работы:

1. Повторить основные операторы и их применение при составлении простейших программ.
2. Ввод программ с клавиатуры дисплея, их редактирование.
3. Анализ результатов исполнения программы.

Во второй части от учащихся требуется:

1. Умение составлять программы по образцам приведенных.
2. Умение составлять и редактировать новые программы с использованием фрагментов из программ, предложенных в первой части.

Если учащимся выполнена только первая часть работы, то ставится оценка «3», так как в этой части работы отсутствует самостоятельность мышления. В зависимости от выполнения самостоятельной работы ставится оценка «4» или «5».

При проведении лабораторно-практической работы в начале урока преподаватель повторяет с учащимися теоретический материал, необходимый для работы. Им указывается справочный материал, который можно найти на стендах в кабинете вычислительной техники,

объясняется порядок выполнения лабораторной работы.

При выполнении первой части преподаватель отвечает на вопросы учащихся, вместе с ними анализирует полученные результаты.

Первая часть работы выполняется под руководством преподавателя. Вторая часть выполняется учащимися самостоятельно. Ее результат предъявляется преподавателю.

Лабораторная работа № 1

49

Тема: Методика работы с ДВК-1М.

Цель: Освоение методики работы с ДВК-1М.

Задания и порядок выполнения работы

Порядок включения ДВК.

1. Нажать клавишу «сеть» на передней панели блока логики.
2. Нажать клавишу «ред» на клавиатуре.

Задание 1.

1. Нажимая поочередно на алфавитно-цифровые клавиши, воспроизведите на экране дисплея буквы, цифры и спец-символы.
2. Проверьте, на какие клавиши распространяется действие клавиш «лат», «рус», «нр» и «вр». В каком виде отображаются при этом символы на экране?
3. Отобразите на экране текст какого-нибудь стихотворения из 8—10 строк.
4. Отобразите на экране дисплея основные операторы Бейсика.

Задание 2.

1. В соответствии с указанными на клавишах обозначениями выполнить перемещение маркера по экрану дисплея: «↓» вниз, «↑» вверх, «→» вправо, «←» влево, «↵» в начало следующей строки, «↶» в начало страницы.

2. Заполните строку любыми символами и поместите маркер в начало, а затем в середину строки. Нажмите поочередно клавиши «→» и «←». Проследите во время этой операции за пове-

дением знаков в позиции, обозначенной маркером.

3. Поместите маркер в начало строки символов и нажмите клавишу «пробел» (стирание символов в строке).

Задание 3.

Используя спецсимволы и буквы, изобразите на экране дисплея картинку (домик, кошку и т. д.).

Сообщите преподавателю о завершении работы.

Выключите ПЭВМ нажатием на клавишу «сеть». Наведите порядок на рабочем месте.

Лабораторная работа № 2

Тема: Интерпретатор Бейсика на ДВК-1М.

50

Цель: Изучение и практическое освоение методов работы в режиме команд на языке Бейсик.

1. Пуск и настройка интерпретатора языка Бейсик.

1. Нажать клавишу «сеть» на передней панели блока логики.

2. Нажать на клавиатуре клавиши «лин», «дуп», «ред».

3. Включить тумблер «сеть» на задней панели ЭВМ.

4. Нажать клавиши «пит», «пуск», «пр» на передней панели блока сопряжения.

5. Набрать число 160000 и букву G. На экране высветится:
БЕЙСИК ДВК НЦ

6. Нажать на клавишу «вк». На экране высветится:

ЖДУ

II. Работа в режиме команд.

С помощью клавиатуры ДВК построчно набрать следующую программу. После ввода каждого оператора (строки) нажимать клавишу «вк», при этом маркер переходит в начало следующей строки.

```
10 FOR I=1 TO 30
20 FOR J=1 TO I
30 PRINT '□';
40 NEXT J
50 PRINT '*';
60 NEXT I
70 END
```

III. Выполнение программы.

Для выполнения программы необходимо набрать на клавиатуре команду RUN и нажать клавишу «вк». На экране высветится график линейной функции.

IV. Введите поочередно следующие изменения в программу.

```
50 PRINT '*'
```

Наберите на клавиатуре команду RUN и выполните измененную программу. Обратите внимание на изменение изображения на экране.

```
30 PRINT '□=□';
```

```
50 PRINT 'ABC'
```

Повторите те же операции, что и в первой задаче IV раздела.

```
10 FOR I=-8 TO 8
```

```
20 FOR J=1 TO I*1
```

```
30 PRINT '□';
```

```
50 PRINT '*'
```

Набрать команду RUN. Выполнить программу. На экране высветится график функции $y=x^2$ ($x^2 \rightarrow I*1$)

Самостоятельная работа

1. Написать программу, выполнить ее и по результатам построить график функции: $y=3x^2-9x-5$.

2. Написать программу для заполнения страницы экрана «ковром» из спецсимволов.

Результаты самостоятельной работы показать преподавателю. Отключить ПЭВМ и навести порядок на рабочем месте.

Лабораторная работа № 3

Тема: Ввод и редактирование линейных программ.

Цель: Научить вводить линейные программы. Познакомить со стандартной функцией INT.

Ввод и редактирование программ.

1. Составить программу для подсчета значений функции:

$y=0,3x^2-1,25x^3+5,64x-8,32$ для X равных: 0,5, -8,126, 3,35, 0,0015.

```
10 INPUT X
```

```
20 LET Y=0.3*X*X-1.25*X*X*X+
+5.64*X-8.32
```

```
30 PRINT 'X='; X, 'Y='; Y
```

```
40 END
```

```
RUN
```

После появления знака вопроса набрать первое значение X и нажать клавишу «вк». Прочитать на дисплее результат. Затем снова набрать оператор RUN и следующее значение X.

2. Написать программу для нахождения значения выражения $Y = \frac{ax^2 + b}{4 - 3c}$.

При $a=3, b=4, c=2, x=1$.

При $a=0.32, b=1.243, c=0.453, x=0.01$.

```
10 INPUT X, A, B, C
20 LET Y = (A*X*X + B) / (4 - 3*C)
30 PRINT 'X='; X 'Y='; Y
40 END
```

3. Стандартная функция позволяет находить целую часть от числа или выражения: $\text{INT}(X) : \text{INT}(5.3) = 5, \text{INT}(5.6) = 5$.

Выполнить программу, которая находит целую часть от деления: $y = \frac{a+b}{a}$ при

a) $a=15.3, b=0.485$; б) $a=112.343, b=18.024$

```
10 INPUT A, B
20 LET Y = (A + B) / A вычисляет значение выражения
30 LET Z = INT(Y) находит целую часть
40 PRINT 'Y='; Y выводит на экран значение выражения и целую часть от него
   'Z='; Z
50 END
```

4. Программа для нахождения остатка от деления: $y = \frac{N}{M}$.

Выполнить программу для $N=15, M=4$; для $N=1234, M=25$.

```
10 INPUT M, N
20 LET Z = N - INT(N/M) * M вычисление остатка от деления
30 PRINT 'Z='; Z
40 END
```

Самостоятельная работа

1. Написать программу для вычисления значения функции и исполнить ее для указанных значений x .

a) $y = 5x^2 - 3x + 2, x = 0.5, x = -3, x = 1.2$

б) $y = \frac{x^3 + 2}{x + 5}, x = 0.5, x = 12.5, x = -6$.

2. Написать программу для нахождения остатка от деления $y = \frac{1986}{a}$.

Исполнить ее для $a=3, a=4, a=5$.
Результаты самостоятельной работы показать преподавателю.

Лабораторная работа № 4

Тема: Ввод и редактирование простых программ, содержащих циклы.

Цель: Научить вводить и редактировать программы, содержащие циклы.

Ввод и редактирование программ

1. Вывести на экран дисплея значения функции $y = 0.3x^3 - 4.125x^2 - 0.01x + 2.25$ для X от -5 до 5 .

```
10 FOR X = -5 TO 5
20 LET Y = 0.3*X*X*X - 4.125*X*X - 0.01*X + 2.25
30 PRINT 'X='; X, 'Y='; Y
40 NEXT X
50 END
```

2. Ввести изменения в предыдущую программу, чтобы значения функции вычислялись с шагом 0.5 . Для решения этой задачи нужно изменить только строку 10.

```
10 FOR X = -5 TO 5 STEP 0.5
   LIST
   RUN
```

3. Вывести на экран таблицу умножения в виде: $2 \cdot 3 = 6$.

Для решения этой задачи будем использовать два цикла: один цикл — по первому множителю, другой — по второму. Так как в строке только пять зон печати, то организуем два цикла, где в первом цикле первый множитель будет меняться от 1 до 5, а второй — от 1 до 9. Во втором цикле первый множитель будет меняться от 6 до 10, а второй — так же, от 1 до 9.

```
10 FOR J = 1 TO 9
20 FOR I = 1 TO 5
30 PRINT I ' * ' J '='; I * J,
40 NEXT I
50 NEXT J
60 PRINT
70 FOR J = 1 TO 9
80 FOR I = 6 TO 10
90 PRINT I ' * ' J '='; I * J,
100 NEXT I
110 NEXT J
120 END
```

Результат показать преподавателю.

4. Генерация случайных чисел.

Функция $\text{RND}(0)$ генерирует случайные числа от 0 до 1.

```
10 LET A = RND(0)
20 PRINT "A="; A
30 END
```

После набора команды RUN ЭВМ выдаст на экран значение A больше 0, но меньше 1. Еще раз исполните программу, набрав команду RUN . Вы получите то же самое число, что и первый раз. Чтобы получить другое число, в программу нужно добавить оператор RANDOMIZE .

Измените программу, набрав:

```
5 RANDOMIZE
LIST
RUN
```

Исполните программу два-три раза. Убедитесь, что получаются различные случайные числа.

5. Получить на экране дисплея 10 случайных чисел от 0 до 1. Чтобы десять раз не набирать команду RUN, будем использовать цикл.

```
10 RANDOMIZE
20 FOR K=1 TO 10
30 LET E=RND(0): PRINT E,
40 NEXT K
50 END
```

6. Получить на экране 10 случайных чисел от 1 до 15. Это значит, что число не должно быть больше 15 и меньше 1.

52

Формула получения случайных чисел на интервале: $(B-A)*RND(0)+A$
Добавим еще условие: числа должны быть только целыми. Для этого будем использовать функцию INT

```
10 RANDOMIZE
20 FOR K=1 TO 10
30 LET E=INT ((15-1)*RND(0)+1)
40 PRINT E
50 NEXT K
60 END
```

Самостоятельная работа

Написать программу для получения 20 случайных целых чисел от 10 до 50.

Лабораторная работа № 5

Тема: простые программы ветвления. Операторы IF THEN, GO TO

Цель: научить вводить и редактировать простые разветвляющиеся программы.

Ввод и редактирование программ

1. Написать программу для нахождения значений функции

$$y = \begin{cases} 5x - 3, & \text{если } x < 0 \\ \sqrt{x}, & \text{если } x \geq 0 \end{cases}$$

для x равного $-8.34, -0.158, 1.21, 625$.

В приведенной ниже программе используется оператор ветвления IF THEN и оператор безусловного перехода GO TO

```
10 INPUT X
20 IF X=0 THEN LET проверяет условие
Y=SQR(X): GO x=0, и если условие
TO 40 выполняется, то вы-
числяет  $y$  как  $\sqrt{x}$ 
и уходит на печать
```

```
30 LET Y=5*X-3 при невыполнении
условия вычисляет
у как  $5x-3$ 
```

```
40 PRINT 'X='; X, организует цикл по
'Y='; Y вводу значений  $x$ 
50 GO TO 10
```

2. Ввести программу для нахождения наибольшего общего делителя (НОД) чисел M и N.

Исполнить ее для $M=1000, N=625, M=1236312, N=9822$.

```
10 INPUT M, N
20 LET X=M:LET Y=N
30 PRINT
40 IF X=Y THEN 100
50 IF X > Y THEN 80
60 LET Y=Y - X
70 GO TO 40
80 LET X=X - Y
90 GO TO 40
100 PRINT '(НОД) ЧИСЕЛ'M'и 'N' PA-
BEH'; X
110 END
```

3. Ввести программу решения квадратного уравнения и исполнить ее для следующих уравнений:

$$\begin{aligned} 1125x^2 - 45x - 324 &= 0 \\ 502x^2 - 38.5x + 12.125 &= 0 \\ 16x^2 - 256x + 1024 &= 0 \end{aligned}$$

```
10 INPUT A, B, C
11 PRINT 'РЕШАЕМ УРАВНЕНИЕ', A;
'*X*X+' B; '*X+'; C'=0'
```

```
15 LET D = B*B - 4*A*C
20 IF D > 0 THEN 50
25 IF D = 0 THEN 35
30 PRINT 'РЕШЕНИЯ НЕТ'; GO TO 65
35 LET X = -B / (2*A)
40 PRINT 'X1=X2='; X:GO TO 65
50 LET X1 = (-B - SQR(D)) / (2*A)
55 LET X2 = (-B + SQR(D)) / (2*A)
60 PRINT 'X1='; X1, 'X2='; X2
65 END
```

Самостоятельная работа

Написать программу для решения линейного уравнения $ax + b = 0$ и исполнить ее для следующих уравнений:

$$12.345x + 37.053 = 0$$

$$144.001x - 23 = 0$$

$$0x + 5 = 0$$

$$0x + 0 = 0$$

Результаты показать преподавателю.

Лабораторная работа № 6

Тема: Стандартные функции языка Бейсик.

Цель: повторить стандартные функции Бейсика и показать примеры их применения в программах.

Ввод и редактирование программ

1. Написать программу для нахождения синуса и косинуса углов: 0° , 60° , 53° , 192° , 564° , 1181° .

Будем использовать стандартные функции SIN (x) и COS (x) Аргумент у этих функций, а также у тангенса измеряется в радианах, поэтому в программу нужно ввести значение числа π и формулу перевода градусов в радианы. Если x — градусы, то $y = \frac{x\pi}{180}$ — радианы.

```
10 PRINT 'ГРАДУСЫ', 'РАДИАНЫ',  
'СИНУС', 'КОСИНУС'  
20 LET P=3.141592  
30 INPUT X  
40 LET Y=X*P/180  
50 PRINT X, Y, SIN (Y), COS (Y)  
60 END
```

2. Написать программу для вычисления квадратного корня чисел от 1 до 10 с шагом 1.

Для написания программы будем использовать цикл, а в теле цикла — стандартную функцию SQR (X)

```
10 FOR X=1 TO 10  
20 LET Y=SQR (X)  
30 PRINT 'X='; X, 'Y='; Y  
40 NEXT X  
50 END
```

3. Написать программу для вычисления абсолютной величины выражения: $y = 4x^2 - 3x - 5$.

Выполнить программу для x равных: -42 , -5 , 4 , 0.0125 .

В программе организуем цикл с помощью оператора безусловного перехода GO TO для ввода значений x .

```
10 INPUT X  
20 LET Y=ABS (4*X*X - 3*X - 5)  
30 PRINT 'X='; X, 'Y='; Y  
40 GO TO 10
```

4. С помощью стандартной функции INT можно округлять числа до любого разряда. Пусть x — число, которое нужно округлить, а D целое число, указывающее, до какого разряда нужно округлить (до десятых — 1, до сотых — 2, до тысячных — 3 и т. д.). Формула округления: $INT (X * 10^{-D} + 0,5) / 10^{-D}$

```
10 PRINT 'ЧИСЛО ДЛЯ ОКРУГЛЕНИЯ'  
20 INPUT X  
30 PRINT 'ЧИСЛО ДЕСЯТИЧНЫХ РАЗРЯДОВ'  
40 INPUT D  
50 LET B=INT (X*10^-D+0,5)/10^-D  
60 PRINT 'ПОСЛЕ ОКРУГЛЕНИЯ ЧИСЛО РАВНО' B  
70 END
```

Исполнить программу для $x = 3.2356$; $D = 1, 2, 3, 4, -1, -2$

После появления первого вопроса ввести число x , после второго D

53

Самостоятельная работа

1. Написать и исполнить программу вычисления значений функции $y = \sin 2x - \operatorname{tg} x$

для углов от -60° до 60° с шагом 10° .

2. Написать программу, которая выводит на экран любое число в виде натурального (например: число — 5, на экране 5; число 8.32, на экране — 8 и т. д.). Исполнить эту программу для чисел: $-5, -8.32, 5.124, 7.54, -125.005$.

Лабораторная работа № 7

Тема: Линейные массивы.

Цель: научить составлять простейшие программы по обработке линейных массивов.

Линейный массив — это линейная таблица, где каждому номеру соответствует свой элемент. Примеры массивов: таблицы квадратов, кубов чисел, таблицы измерения температуры и т. д.

Линейный массив обозначается именем (буквой) и номером последнего элемента. A(5) — линейный массив из 6 элементов (0, 1, 2, 3, 4, 5 — номера элементов).

Оператор DIM резервирует в памяти ЭВМ место для массива.

Задача 1. Пусть дана таблица 1:

0	1	2	3	4	5
2	4	-3	5	7	-1

Первая строка — номера элементов, вторая — их значения.

Ввести в память ЭВМ значения элементов массива и вывести на экран в виде: $A(0)=2$, $A(1)=4$, и т. д.

```
10 DIM A(5)           обнуляет эле-
20 FOR I=0 TO 5       менты массива
30 LET A(I)=0:NEXT I
40 LET I=0
50 INPUT X
57 LET A(I)=X         присваивает
60 LET I=I+1          элементу мас-
65 IF I<=5 THEN 50    сива значения
70 FOR I=0 TO 5       из данной таб-
                       лицы
75 PRINT 'A(I)='; A(I) организует вы-
80 NEXT I             вод на экран
85 END                элементов мас-
                       сива
```

54 **Задача 2.** Дополнить программу так, чтобы найти сумму элементов массива.

```
85 LET S=0             присваивает сумме
87 FOR I=0 TO 5       значение, равное 0
90 LET S=S+A(I)
100 NEXT I            цикл подсчета суммы
110 PRINT 'S='; S     элементов
120 END
```

Задача 3. Дополнить программу так, чтобы можно было подсчитать количество отрицательных элементов в массиве.

```
10 DIM A(14)
15 LET S=0:LET N=0
20 FOR X=0 TO 14
30 LET A(X)=X*X-2
40 PRINT 'A(X)='; A(X)
50 NEXT X
60 FOR X=0 TO 14
70 LET S=S+A(X)

80 IF A(X)<0 THEN LET N=N+1
90 NEXT X
100 PRINT 'S='; S, 'N='; N
110 END
```

Самостоятельная работа

Задача 1. Написать и исполнить программу заполнения и вывода на экран таблицы квадратов от 1 до 40.

Задача 2. Написать и исполнить программу заполнения массива из 20 элементов по формуле $y=0,2x^2-3x+0,1$ и подсчета суммы положительных элементов.

Лабораторная работа № 8

Тема: Линейные массивы.

Цель: Продолжить изучение простейших программ по обработке массивов.

Введем счетчик N, присвоим ему значение 0 и будем прибавлять к нему по 1 каждый раз, как встретится отрицательный элемент.

```
85 LET S=0:LET N=0
91 IF A(I)<0 THEN LET N=N+1
110 PRINT 'S=';S, 'N='; N
```

Задача 4. Составить программу для заполнения таблицы кубов и вывода ее на экран дисплея.

```
10 DIM A(15)
20 FOR I=1 TO 15
30 LET A(I)=I*I*I
40 PRINT 'A(I)='; A(I)
50 NEXT I
60 END
```

Задача 5. Элементы массива вычисляются по формуле $y=x^2-x-2$, где x — номер элемента, а y — его значение.

Написать программу, которая: напечатает на экране первые 15 элементов;

подсчитает сумму;

подсчитает количество отрицательных элементов.

обнуляет значения суммы и счетчика отрицательных элементов
цикл подсчета значений элементов и вывод их на экран

к предыдущей сумме прибавляет значение следующего элемента
проверяет, будет ли элемент отрицательным. Если да, то увеличивает значение счетчика на 1

Ввод и редактирование программ

Задача 1. Элементы массива $A(20)$ заполняются по формуле $A(I)=2I-I^2+4$.

Написать программу, которая находит минимальный элемент в этом массиве и, если он не один, их количество и номера.

Задача 2. Написать программу, которая будет сравнивать значения выражений I^2-1 и J^3-1 для целых значений I и J и будет находить, при каких значениях I и J они равны.


```

10 DIM A(10), B(10)
20 FOR I=0 TO 10
30 LET A(I)=I*I-1
40 PRINT 'A(I)='; A(I)
45 NEXT I
47 FOR J=1 TO 10
50 LET B(J)=J*J-1
60 PRINT 'B(J)='; B(J)
70 NEXT J
80 FOR I=0 TO 10
90 FOR J=0 TO 10
100 IF A(I)=B(J) THEN PRINT 'A(I)=B(J)='; A(I)
110 NEXT J
120 NEXT I
130 END

```

цикл подсчета и вывода на экран элементов массива А

цикл подсчета и вывода на экран элементов массива В.

вложенные циклы проверки равенства

Самостоятельная работа

Задача 1. Изменить программу к задаче так, чтобы были найдены значения I и J, при которых $A(I) < B(J)$ ($I=J$).

Задача 2. Элементы массива заполняются по формуле $x(x-3)$ (x^2-81).

Написать и исполнить программу, находящую из первых 20 элементов значения x, при которых значения элементов равны 0.

Лабораторная работа № 9

Тема: Двухмерные массивы.

Цель: Научить писать простейшие программы по обработке двухмерных массивов.

Таблица, состоящая из n строк и m столбцов, называется прямоугольным массивом. Каждому элементу массива

соответствует свой номер строки и номер столбца, причем на первом месте всегда стоит номер строки. А(3,6) — двухмерный массив из 4 строк и 7 столбцов. Задавая массив, мы должны указать номера максимальной строки и максимального столбца, начинать работу можем с любого номера, только предварительно указав его.

Рассмотрим двухмерный массив А(3,5).

1	2	3	4	5
1	2	-3	5	-1
2	1	2	25	3
3	7	6	0	3

Задача 1. Загрузить значения элементов данного массива в память ЭВМ, вывести их на экран дисплея.

I — номер строки, J — номер столбца.

цикл ввода и присваивания элементам массива значений из данной таблицы

цикл вывода значений элементов на экран дисплея

2, -3, 5, 4, -1, 1, 2 и т. д. После каждого числа нажимаем клавишу «ВК».

Задача 1а. Дополнить программу для нахождения суммы положительных и отрицательных элементов.

```

10 DIM A(3,5)
20 FOR I=1 TO 3
30 FOR J=1 TO 5
40 INPUT X
50 LET A(I,J)=X
60 NEXT J: NEXT I
70 FOR I=1 TO 3
80 FOR J=1 TO 5
90 PRINT 'A(I, 'J')='; A(I, J)
100 NEXT J: NEXT I
110 END

```

После набора команды RUN на экране появляется знак «?», обусловленный оператором INPUT. При появлении знака «?» вводить значения элементов из таблицы. Числа вводятся построчно:

```

150 LET S=0:LET S1=0 присваивает S и S1 нулевые значения
160 FOR I=0 TO 3
170 FOR J=1 TO 5
180 IF A(I, J)>0 THEN LET S=S+A(I,J) сумма положительных элементов
190 IF A(I, J)<0 THEN LET S1=S1+A(I, J) сумма отрицательных элементов

```

```

200 NEXT J:NEXT I
210 PRINT'СУММА ПОЛОЖИТЕЛЬНЫХ ЭЛЕМЕНТОВ S='; S
220 PRINT'СУММА ОТРИЦАТЕЛЬНЫХ ЭЛЕМЕНТОВ 'S1='; S1
230 END

```

Задача 2. Элементы массива $A(10, 10)$ заполняются как произведение номера строки на номер столбца ($A(5,6)=30$). Вывести значения элементов на экран дисплея. Найти сумму значений элементов, номера строк и столбцов у которых совпадают ($A(6,6)=36$).

```

10 DIM A(10,10)
11 LET S=0
20 FOR I=0 TO 10
30 FOR J=0 TO 10
40 LET A(I, J)=I*J    вычисление значений
                       элементов массива
50 IF I=J THEN LET S=S+A(I, J) под-
   счет суммы A(I, J), у которых I=J
70 NEXT J:NEXT I
60 PRINT 'A('I', 'J')='; A(I, J)
80 PRINT 'S='; S
90 END

```

56

Задача 3. Значения элементов массива вычисляются по формуле $A(X, Y)=5X-3Y$.

Написать программу, подсчитывающую количество положительных и отрицательных элементов и элементов, равных нулю.

Результаты вывести на экран дисплея.

```

10 DIM A(10,10)
20 LET N=0:LET N1=0:LET N0=0
30 REM N количество положительных
   элементов
31 REM N1 количество отрицательных,
   N0 — равных нулю.
40 FOR X=0 TO 10
50 FOR Y=0 TO 10
60 LET A(X, Y)=5*X-3*Y
65 PRINT 'A('X', 'Y')='; A(X, Y)
70 IF A(X, Y)>0 THEN LET N=N+1
80 IF A(X, Y)<0 THEN LET N1=N1+1
90 IF A(X, Y)=0 THEN LET N0=N0+1
100 NEXT Y:NEXT X
110 PRINT 'N='; N, 'N1='; N1
120 IF N0<>0 THEN PRINT 'N0=';
   N0:GO TO 140
130 PRINT 'ЭЛЕМЕНТОВ, РАВНЫХ 0,
   НЕТ'
140 END

```

Самостоятельная работа

Элементы массива вычисляются по формуле $A(X, Y)=2x^2-y^2$.

Написать программу, которая: выводит на экран значения элементов массива $A(2,2)$;

подсчитывает количество отрицательных элементов;

сумму положительных элементов;

выводит на экран номера всех элементов, значения которых равны нулю.

Лабораторная работа № 10

Тема: Использование ДВК в качестве контролера знаний.

Цель: Проверить умение вводить готовые программы в память ЭВМ; проверить умение учащихся вычислять площади поверхностей многогранников и тел вращения.

Ход работы

1. На первом уроке ввести готовые программы в память ДВК.

2. На втором уроке взять нужные модели геометрических тел и вычислить площади их поверхностей и объемы.

За эту работу выставить две оценки: первую оценку за ввод программ поставить в журнал по информатике, вторую, согласно поставленной ЭВМ оценке, поставить в журнал по геометрии.

Список программ

1. Площадь поверхности прямой призмы.

2. Объем призмы.

3. Площадь поверхности правильной треугольной пирамиды.

4. Площадь поверхности правильной четырехугольной пирамиды.

5. Объем пирамиды.

6. Площадь поверхности и объем цилиндра.

7. Площадь поверхности и объем конуса.

8. Площадь поверхности сферы и объем шара.

Площадь поверхности прямой призмы

5 PRINT 'A, B, C, D...—длины СТОРОН ОСНОВАНИЯ ПРИЗМЫ'

6 PRINT 'M — ВЫСОТА ОСНОВАНИЯ, ПРОВЕДЕННАЯ К A'

7 PRINT 'H — ВЫСОТА ПРИЗМЫ'

10 PRINT 'СКОЛЬКО СТОРОН В ОСНОВАНИИ?'

20 INPUT N

30 PRINT 'ИЗМЕРЬТЕ И ВВЕДИТЕ

```

ДЛИНЫ СТОРОН ОСНОВАНИЯ'
31 PRINT 'ВЫСОТУ ОСНОВАНИЯ И
    ВЫСОТУ ПРИЗМЫ'
40 IF N=3 THEN INPUT A, B, C, M,
    H:GO TO 80
50 IF N=4 THEN INPUT A, B, C, D, M,
    H:GO TO 90
60 IF N=5 THEN INPUT A, B, C, D, E, M,
    H:GO TO 100
70 IF N=6 THEN INPUT A, B, C, D, E, F,
    M, H:GO TO 110
80 LET S1=(A+B+C)*H+3*A*M:GO TO
    120
90 LET S1=(A+B+C+D)*H+
    +4*A*M:GO TO 120
100 LET S1=(A+B+C+D+E)*H+
    +5*A*M:GO TO 120
110 LET S1=(A+B+C+D+E+F)*H+
    +6*A*M:GO TO 120
120 PRINT 'ВЫЧИСЛИТЕ ПЛОЩАДЬ ПО-
    ВЕРХНОСТИ И СООБЩИТЕ МНЕ'
130 INPUT S
140 LET K=ABS(S-S1)
150 IF K<=0.1 THEN PRINT 'ОЦЕНКА
    5':GO TO 200
160 IF K<=1 THEN PRINT 'ОЦЕНКА
    4':GO TO 200
170 IF K<=2 THEN PRINT 'ОЦЕНКА
    3':GO TO 200
180 PRINT 'ОЦЕНКА 2. ПОВТОРИТЕ ВЫ-
    ЧИСЛЕНИЯ'
190 GO TO 120
200 PRINT 'S='; S1
205 END

```

Рассматривать для вычисления пло-
щади поверхности призмы, в основа-
нии которых лежат: треугольник, квад-
рат, прямоугольник, правильные пяти-
угольник и шестиугольник.

Объем призмы

```

10 PRINT 'A — ДЛИНА СТОРОНЫ
    ОСНОВАНИЯ, M — ВЫСОТЫ ОСНО-
    ВАНИЯ,'
20 PRINT 'H — ДЛИНА ВЫСОТЫ
    ПРИЗМЫ.'
30 INPUT A, M, H
40 PRINT 'ВВЕДИТЕ N — ЧИСЛО СТО-
    РОН ОСНОВАНИЯ'
50 INPUT N
60 IF N=3 THEN LET V1=0.5*A*M*H:GO
    TO 100
70 IF N=4 THEN LET V1=A*H:GO
    TO 100
80 IF N=5 THEN LET V1=2.5*A*M*H
90 IF N=6 THEN LET V1=3*A*M*H
100 PRINT 'ВЫЧИСЛИТЕ И СООБЩИТЕ
    МНЕ ВЕЛИЧИНУ ОБЪЕМА ПРИЗ-
    МЫ'
110 INPUT V

```

```

120 LET K=ABS (V-V1)
130 IF K<=0.1 THEN PRINT 'ОЦЕН-
    КА 5':GO TO 170
140 IF K<1 THEN PRINT 'ОЦЕНКА 4':
    GO TO 170
150 IF K<3 THEN PRINT 'ОЦЕНКА 3':
    GO TO 180
160 IF K>3 THEN PRINT 'ОЦЕНКА 2':
    GO TO 180
170 PRINT 'V='; V1:GO TO 200
180 PRINT 'ПОВТОРИТЕ ВЫЧИСЛЕНИЯ
    И НОВЫЙ РЕЗУЛЬТАТ СООБЩИТЕ
    МНЕ'
190 GO TO 110
200 END

```

Площадь поверхности правильной треугольной пирамиды

```

10 PRINT 'ИЗМЕРЬТЕ И СООБЩИТЕ
    МНЕ ДЛИНУ СТОРОНЫ ОСНОВА-
    НИЯ'
20 PRINT 'ВЫСОТЫ ОСНОВАНИЯ,
    ДЛИНУ АПОФЕМЫ ПИРАМИДЫ'
25 INPUT A, M, H
30 PRINT 'ВЫЧИСЛИТЕ ПЛОЩАДЬ
    ПОВЕРХНОСТИ И РЕЗУЛЬТАТ'
35 PRINT 'СООБЩИТЕ МНЕ'
40 INPUT S
50 LET S1=1.5*A*H+1/2*A*M
60 IF ABS (S-S1)<=0.1 THEN PRINT
    'ОЦЕНКА 5':GO TO 110
70 IF ABS (S-S1)<1 THEN PRINT
    'ОЦЕНКА 4':GO TO 110
80 IF ABS (S-S1)<2 THEN PRINT
    'ОЦЕНКА 3':GO TO 110
90 PRINT 'ОЦЕНКА 2. ПОВТОРИТЕ
    ВЫЧИСЛЕНИЯ И НОВЫЙ РЕЗУЛЬ-
    ТАТ'
95 PRINT 'СООБЩИТЕ МНЕ'
100 GO TO 40
110 PRINT 'S='; S1
120 END

```

57

Площадь поверхности правильной четырехугольной пирамиды

```

10 PRINT 'ИЗМЕРЬТЕ И СООБЩИТЕ
    МНЕ ДЛИНУ СТОРОНЫ ОСНО-
    ВАНИЯ'
11 PRINT 'И ДЛИНУ АПОФЕМЫ ПИРА-
    МИДЫ'
20 INPUT A, M
30 PRINT 'ВЫЧИСЛИТЕ ПЛОЩАДЬ
    ПОЛНОЙ ПОВЕРХНОСТИ И РЕ-
    ЗУЛЬТАТ'
31 PRINT 'СООБЩИТЕ МНЕ'
40 INPUT S
50 LET S1=2*A*M+A*A
60 IF ABS (S-S1)<=0.1 THEN PRINT
    'ОЦЕНКА 5':GO TO 110
70 IF ABS (S-S1)<1 THEN PRINT

```

```

'ОЦЕНКА 4':GO TO 110
80 IF ABS (S—S1)<2 THEN PRINT
'ОЦЕНКА 3':GO TO 110
90 PRINT 'ОЦЕНКА 2. ПОВТОРИТЕ ВЫ-
ЧИСЛЕНИЯ И НОВЫЙ РЕЗУЛЬТАТ'
91 PRINT 'СООБЩИТЕ МНЕ'
100 GO TO 40
110 PRINT 'S='; S1
120 END

```

Объем пирамиды

```

10 PRINT 'A — СТОРОНА ОСНОВАНИЯ,
M — ВЫСОТА ОСНОВАНИЯ,'
15 PRINT 'ПРОВЕДЕННАЯ К A, H —
ВЫСОТА ПИРАМИДЫ'
20 PRINT 'ИЗМЕРЬТЕ И СООБЩИТЕ
МНЕ ЗНАЧЕНИЯ A, M, H'
30 INPUT A, M, H
40 PRINT 'ВВЕДИТЕ ЧИСЛО СТОРОН
ОСНОВАНИЯ ПИРАМИДЫ'
50 INPUT N
60 IF N=3 THEN LET V1=1/3*0.5*A*
M*H:GO TO 100
70 IF N=4 THEN LET V1=1/3*A*M*H:
GO TO 100
80 IF N=5 THEN LET V1=1/3*2.5*A*
M*H:GO TO 100
90 IF N=6 THEN LET V1=A*M*H:
GO TO 100
100 PRINT 'ВЫЧИСЛИТЕ ОБЪЕМ ПИРА-
МИДЫ И РЕЗУЛЬТАТ СООБЩИТЕ
МНЕ'
110 INPUT V
120 LET K=ABS (V—V1)
130 IF K<=0.01 THEN PRINT 'ПРАВИЛЬ-
НО, ОЦЕНКА 5':GO TO 180
140 IF K<=1 THEN PRINT 'ОЦЕНКА 4':
GO TO 180
150 IF K<=2 THEN PRINT 'ОЦЕНКА 3':
GO TO 180
160 PRINT 'ОЦЕНКА 2. ПОВТОРИТЕ
ВЫЧИСЛЕНИЯ И НОВЫЙ РЕЗУЛЬ-
ТАТ'
161 PRINT 'СООБЩИТЕ МНЕ'
170 GO TO 110
180 PRINT 'V='; V1
190 END

```

Рассматривать для вычисления объема пирамиды, в основании которых лежат: треугольник, квадрат, прямоугольник, правильный пятиугольник или шестиугольник.

Площадь поверхности и объем цилиндра

```

10 PRINT 'ИЗМЕРЬТЕ И ВВЕДИТЕ
ДЛИНУ РАДИУСА И ВЫСОТЫ ЦИ-
ЛИНДРА'
20 INPUT R, H
30 LET S1=6.28*R*H+6.28*R*R
35 LET V1=3.14*R*R*H

```

```

40 PRINT 'ВЫЧИСЛИТЕ И СООБЩИТЕ
МНЕ ЗНАЧЕНИЯ S И V. КОТОРЫЕ
ВЫ ПОЛУЧИЛИ'
50 INPUT S, V
60 IF ABS (S—S1)<=0.01 THEN 100
70 IF ABS (S—S1)<1 THEN 140
80 IF ABS (S — S1)<2 THEN 180
90 PRINT 'ПОВТОРИТЕ ВЫЧИСЛЕНИЕ
ПЛОЩАДИ ПОВЕРХНОСТИ ЦИ-
ЛИНДРА':GO TO 40
100 IF ABS (V—V1)<=0.01 THEN PRINT
'ОЦЕНКА 5':GO TO 210
110 IF ABS (V—V1)<1 THEN PRINT
'ОЦЕНКА 4':GO TO 210
120 IF ABS (V—V1)<2 THEN PRINT
'ОЦЕНКА 3':GO TO 210
130 PRINT 'ПОВТОРИТЕ ВЫЧИСЛЕНИЕ
ОБЪЕМА ЦИЛИНДРА':GO TO 40
150 IF ABS (V—V1)<1 THEN PRINT
'ОЦЕНКА 4': GO TO 210
160 IF ABS (V—V1)<2 THEN PRINT
'ОЦЕНКА 3':GO TO 210
170 GO TO 130
180 IF ABS (V—V1)<=0.01 THEN PRINT
'ОЦЕНКА 4':GO TO 210
190 IF ABS (V—V1)<2 THEN PRINT
'ОЦЕНКА 3':GO TO 210
200 GO TO 130
210 PRINT 'S='; S1, 'V='; V1
220 END

```

Площадь поверхности и объем конуса

```

10 PRINT 'ИЗМЕРЬТЕ И ВВЕДИТЕ ДЛИ-
НУ РАДИУСА, ОБРАЗУЮЩЕЙ'
11 PRINT 'И ВЫСОТУ КОНУСА'
20 INPUT R, L, H
30 LET S1=3.14*R*(L+R)
35 LET V1=1/3*3.14*R*H
40 PRINT 'ВЫЧИСЛИТЕ И СООБЩИТЕ
МНЕ ПЛОЩАДЬ ПОВЕРХНОСТИ'
41 PRINT 'И ОБЪЕМ КОНУСА'
50 INPUT S, V
60 LET K=ABS (S—S1):LET N=ABS (V—
—V1)
70 IF K<0.01 THEN 110
80 IF K<1 THEN 150
90 IF K<2 THEN 180
100 PRINT 'ПОВТОРИТЕ ВЫЧИСЛЕНИЕ
ПЛОЩАДИ ПОВЕРХНОСТИ КОНУ-
СА':GO TO 40
110 IF N<=0.01 THEN PRINT 'ОЦЕНКА
5':GO TO 210
120 IF N<1 THEN PRINT 'ОЦЕНКА 4':GO
TO 210
130 IF N<2 THEN PRINT 'ОЦЕНКА 3':
GO TO 210
140 PRINT 'ПОВТОРИТЕ ВЫЧИСЛЕНИЕ
ОБЪЕМА КОНУСА':GO TO 40
150 IF N<1 THEN PRINT 'ОЦЕНКА 4':
GO TO 210
160 IF N<2 THEN PRINT 'ОЦЕНКА 3':GO

```

```

TO 210
170 GO TO 140
180 IF N<=0.01 THEN PRINT 'ОЦЕНКА
4':GO TO 210
190 IF N<2 THEN PRINT 'ОЦЕНКА 3':
GO TO 210
200 GO TO 140
210 PRINT 'S='; S1; 'V='; V1
220 END

```

Площадь поверхности сферы и объем шара

```

10 PRINT 'ИЗМЕРЬТЕ И ВВЕДИТЕ ЗНА-
ЧЕНИЯ РАДИУСА'
20 INPUT R
30 LET S1=4*3.14*R*R:LET V1=
4/3*3.14*R*R*R
40 PRINT 'ВЫЧИСЛИТЕ И СООБЩИТЕ
МНЕ ЗНАЧЕНИЯ ПЛОЩАДИ ПО-
ВЕРХНОСТИ'
41 PRINT 'И ОБЪЕМА ШАРА'
50 INPUT S, V
60 LET K=ABS (S-S1):LET N=ABS
(V-V1)
70 IF K<=0.01 THEN 110
80 IF K<1 THEN 150
90 IF K<2 THEN 180
100 PRINT 'ПОВТОРИТЕ ВЫЧИСЛЕНИЕ
ПЛОЩАДИ СФЕРЫ':GO TO 40
110 IF N<=0.01 THEN PRINT 'ОЦЕН-
КА 5':GO TO 210
120 IF N<=1 THEN PRINT 'ОЦЕНКА 4':
GO TO 210
130 IF N<=2 THEN PRINT 'ОЦЕНКА 3':
GO TO 210
140 PRINT 'ПОВТОРИТЕ ВЫЧИСЛЕНИЯ
ОБЪЕМА ШАРА':GO TO 40
150 IF N<=1 THEN PRINT 'ОЦЕНКА 4':
GO TO 210
160 IF N<=2 THEN PRINT 'ОЦЕНКА 3':
GO TO 210
170 GO TO 140
180 IF N<=0.01 THEN PRINT 'ОЦЕН-
КА 4':GO TO 210
190 IF N<=2 THEN PRINT 'ОЦЕНКА 3':
GO TO 210
200 GO TO 140
210 PRINT 'S='; S1, 'V='; V1
220 END

```

Лабораторная работа № 11

Тема: Игровые программы.

Цель: Проверить умение вводить в память ДВК готовые программы. Развивать у учащихся интерес к вычислительной технике.

Ход работы

1. На первом уроке ввести в ПЭВМ программы, предложенные преподавателем.

2. На втором уроке учащиеся играют с ПЭВМ, согласно введенным программам.

Оценка учащимся ставится за правильность ввода программ.

Список программ

1. Посадка на Луну.
2. Морской бой.
3. Игра Баше.
4. Отгадай число.
5. Проверь свои способности.
6. Потопи корабль.

Посадка на луну

```

10 LET R=1.74E6:LET H=
=15000:LET G0=1.62
20 LET V=1000:LET M=500:LET M1=
250
30 LET U=3000:LET T1=0
40 PRINT 'ВЫ ПИЛОТ РАКЕТЫ, НАХО-
ДЯЩЕЙСЯ НАД ПОВЕРХНОСТЬЮ
ЛУНЫ НА'
41 PRINT 'ВЫСОТЕ 15 КМ. ВАМ НУЖ-
НО ПОСАДИТЬ РАКЕТУ НА ЛУНУ'
42 PRINT 'ЗАДАВАЯ ДИСКРЕТНО ВРЕ-
МЯ ТОРМОЖЕНИЯ В СЕКУНДАХ И
ВЕЛИЧИНУ'
43 PRINT 'РАСХОДА ТОПЛИВА В КГ/С,
НАЧИНАЙТЕ ПОСАДКУ'
44 PRINT 'У ВАС 250 КГ ТОПЛИВА'
50 PRINT
60 PRINT 'ВВЕДИТЕ ВРЕМЯ ТОРМО-
ЖЕНИЯ (Т С) И ВЕЛИЧИНУ РАС-
ХОДА'
61 PRINT 'ТОПЛИВА (L КГ/С)'.
70 INPUT T, L
80 LET G1=G0/(1+H/R)^2
90 LET A1=U*L/(M+M1)-G1
100 LET V1=V-A1*T
110 LET H1=H-V*T+A1*T^2/2
120 LET M2=M1-L*T
130 LET T1=T1+T
140 PRINT 'ПОСЛЕ ТОРМОЖЕНИЯ В
ТЕЧЕНИЕ «Т» СЕКУНД:'
150 PRINT 'ВЫСОТА НАД ПЛАНЕТОЙ
H='; H1
151 PRINT 'СКОРОСТЬ СПУСКА V='; V1
152 PRINT 'УСКОРЕНИЕ ТОРМОЖЕНИЯ
A='; A1
153 PRINT 'ОСТАЛОСЬ ТОПЛИВА M=';
M2
154 PRINT 'ОБЩЕЕ ВРЕМЯ ТОРМОЖЕ-
НИЯ T1='; T1
155 PRINT
160 IF A1>49 THEN PRINT 'СЛИШКОМ
БОЛЬШАЯ ПЕРЕГРУЗКА. ВЫ ПО-
ГИБЛИ':GO TO 260
170 IF A1=49 THEN PRINT 'ПРЕДЕЛЬ-
НАЯ НАГРУЗКА':GO TO 230

```

```

180 IF M2<0 THEN PRINT 'ТОПЛИВО
      КОНЧИЛОСЬ': GO TO 260
182 IF H1<=2 THEN 184
183 GO TO 195
184 IF H2>=-2 THEN 192
185 IF V1>5 THEN PRINT 'БОЛЬШАЯ
      СКОРОСТЬ. ВЫ РАЗБИЛИСЬ': GO TO
      260
186 LET D=V2-2*A1*H
187 LET T2=(-V+SQR(D))/A1
188 PRINT 'ВЫ ЗАТРАТИЛИ НА ПОСАД-
      КУ ЛИШНИЕ' T-T2 'СЕКУНД'
189 PRINT 'ПРОЖГЛИ ПРИЛИЧНУЮ
      ЯМУ И РАКЕТА УПАЛА'
190 PRINT 'ЖДИТЕ ПОМОЩИ':GO TO 260
192 IF V1<=5 THEN PRINT 'МЯГКАЯ
      ПОСАДКА. МОЛОДЕЦ!': GO TO 260
193 PRINT 'СЛИШКОМ БОЛЬШАЯ СКО-
      РОСТЬ. ВЫ РАЗБИЛИСЬ': GO TO 260
195 IF V1=0 THEN PRINT 'ВЫ ЗАВИСЛИ
      НАД ЛУНОЙ': GO TO 230
60 200 IF V1<0 THEN PRINT 'ВЫ УДАЛЯЕ-
      ТЕСЬ ОТ ЛУНЫ': GO TO 230
210 IF V1>0 THEN PRINT 'СЛЕДИТЕ
      ЗА СКОРОСТЬЮ': GO TO 230
220 PRINT 'ПРОДОЛЖАЙТЕ ПОСАДКУ.
      ВСЕ ИДЕТ НОРМАЛЬНО'
230 LET H=H1 : LET V=V1 : LET M1=M2
240 GO TO 50
260 END

```

Обозначения

M — масса ракеты (500 кг)
M1 — масса топлива (250 кг)
H — высота над планетой (15 км)
H1 — высота над планетой в данный момент времени
V — начальная скорость спуска (1000 м/с)
V1 — скорость спуска на высоте H1
G0 — ускорение свободного падения у поверхности планеты (1.62)
R — радиус планеты
U — скорость вылета раскаленных газов из сопла (3000 м/с)
L — расход топлива в кг/с
49 м/с — предельное ускорение
A1 — ускорение, с которым движется ракета
G1 — ускорение свободного падения на высоте H1
T — дискретное время торможения
T1 — общее время торможения

Допущения

1. В течение одного интервала времени не изменяется полная масса ракеты.
2. Не учитывается сопротивление атмосферы планеты.
3. Положительное направление ско-

рости примем ориентированным к центру планеты: $V>0$ — ракета приближается к планете, $V=0$ — зависает над планетой, $V<0$ — удаляется от планеты.

4. Положительное ускорение направлено от центра планеты, отрицательное — к центру.

«Морской бой»

```

5 RANDOMIZE
7 LET N=10:LET N1=10
10 DIM K (10, 10), K1 (10, 10)
20 FOR I=1 TO 10
30 FOR J=1 TO 10
40 LET K (I, J)=0:LET K1 (I, J)=0
50 NEXT J
60 NEXT I
70 DEF FNK (J)=INT (RND(0)*J+1)
80 FOR I=1 TO 10
90 LET I1= FNK (10)
100 LET J1= FNK (10)
120 IF K1 (I1, J1)=2 THEN 90
125 LET K1 (I1, J1)=2
130 NEXT I
150 FOR I=1 TO 10
160 PRINT 'ВВЕДИТЕ КООРДИНАТЫ
      КОРАБЛЯ'
165 INPUT I1, J1
170 LET K (I1, J1)=1
190 NEXT I
200 GOSUB 5000
210 LET I1= FNK (10)
220 LET J1=FNK (10)
230 IF K (I1, J1)=2 THEN 210
240 IF K (I1, J1)=1 THEN 280
245 PRINT 'ВЫСТРЕЛ ПО КВАДРАТУ'
      I1 '=' J1
250 PRINT 'ОХ! КАКАЯ НЕУДАЧА! Я
      ПРОМАЗАЛА:'
260 LET K (I1, J1)=2
270 GO TO 340
280 PRINT 'ХА-ХА'
290 PRINT 'ВАШ КОРАБЛЬ В КВАДРАТЕ'
      I1 '=' J1 'ПОТОПЛЕН'
300 LET K (I1, J1)=2
295 LET N=N-1:IF N=0 THEN GO TO
      1000
310 GOSUB 5000
320 PRINT 'Я СТРЕЛЯЮ СНОВА'
330 GO TO 210
340 PRINT 'ВВЕДИТЕ КООРДИНАТЫ
      КОРАБЛЯ'
350 INPUT I1, J1
360 IF K1 (I1, J1)=2 THEN 390
370 PRINT 'ХА-ХА! МИМО!!!'
380 GO TO 210
390 PRINT 'ВЫ ПОДБИЛИ МОЙ КО-
      РАБЛЬ'
395 LET N1=N1-1
396 IF N1=0 THEN 1020
400 LET K1 (I1, J1)=1:GOSUB 5000

```

```

420 GO TO 340
1000 FOR I=1 TO 30:PRINT:NEXT
1005 PRINT 'ВЫ ПРОИГРАЛИ'
1010 STOP
1020 PRINT 'Я ПРОИГРАЛА': STOP
5000 PRINT 'ВАШЕ МОРЕ (ВВЕСТИ 30
      ПРОБЕЛОВ). МОРЕ ЭВМ'
5010 FOR I=1 TO 10
5015 FOR J=1 TO 10
5020 IF K(I,J)=1 THEN PRINT '  *';
5025 IF K(I,J)<>1 THEN PRINT '  .';
5030 NEXT J
5035 PRINT 'ВВЕСТИ 30 ПРОБЕЛОВ';
5040 FOR J=1 TO 10
5045 IF K1(I,J)=1 THEN PRINT '  O';
5050 IF K1(I,J)<>1 THEN '  X';
5055 NEXT J
5060 PRINT
5065 NEXT I
5075 RETURN

```

После ввода программы на листе бумаги нарисовать два квадрата размером 10 на 10 клеток. И строчки и столбцы пронумеровать числами от 1 до 10. Один квадрат — ваше море, другой — море ЭВМ. На своем разместить 10 кораблей. Каждый корабль — одна клетка. Наберите команду RUN и после появления знака вопроса введите координаты кораблей. На экране появятся два моря: ваше и ЭВМ. На вашем море корабли будут отмечены звездочками, а на море ЭВМ — пока только точки. Если ЭВМ попадет в корабль, то она уберет соответствующую звездочку, а если вы попадете в ее корабль, то она поставит знак ☒ на месте своего корабля.

Игра Баше

После ввода программы задайте общее число предметов (N) и максимальное число предметов, которое можно брать за один раз (M). Проигрывает тот, кто берет последний предмет.

```

10 PRINT 'ВВЕДИТЕ N,M'
30 INPUT N,M
35 LET N=INT(N):LET M=INT(M)
40 PRINT 'ЧЕЙ ХОД? 1 — ЕСЛИ ПЕР-
      ВЫЙ ХОД ДЕЛАЕТ МАШИНА'
41 PRINT '0 — ЕСЛИ ПЕРВЫЙ ХОД ЧЕ-
      ЛОВЕКА'
50 INPUT I
70 IF I>0 THEN 210
70 PRINT 'ВАШ ХОД'
80 INPUT Y
85 LET Y=INT(Y)

```

```

90 IF Y<1 THEN 70
100 IF Y>M THEN 70
160 LET N=N-Y
170 IF N>0 THEN 200
180 PRINT 'ВЫ ПРОИГРАЛИ'
190 STOP
200 LET T=Y
205 PRINT 'ОСТАЛОСЬ'; N
210 LET R=N
212 IF N<=M THEN 250
215 LET R=N/(M+1)
220 LET R=N-INT(R)*(M+1)
230 IF R=0 THEN 290
240 IF R=1 THEN 270
250 LET X=R-1
255 IF X<=0 THEN 290
260 GO TO 300
270 LET X=INT(RND(0)*M+1)
280 GO TO 300
290 LET X=M
300 IF X>M THEN 290
302 LET X=INT(X)
305 LET N=N-X
310 IF N<=0 THEN 340
320 PRINT 'ВЗЯЛА'; X
325 PRINT 'ОСТАЛОСЬ'; N
330 GO TO 70
340 PRINT 'ПОЗДРАВЛЯЮ! ВЫ ВЫИГРА-
      ЛИ'
350 END

```

61

Игра «Отгадай число»

Машина задумывает число от 1 до 100. Ваша задача — отгадать это число. Выигрывает тот, кто за меньшее число ходов отгадает задуманное ЭВМ число.

```

10 PANDOMIZE
15 LET E=INT(RND(0)*100)+1
20 LET X=0
25 PRINT 'УГАДАЙ ЧИСЛО, КОТОРОЕ
      Я ЗАДУМАЛА'
30 INPUT C
35 IF C=E THEN 75
40 IF C>E THEN 60
45 PRINT 'МАЛО'
50 LET X=X+1
55 GO TO 25
60 PRINT 'МНОГО'
65 LET X=X+1
70 GO TO 25
75 PRINT 'ПРАВИЛЬНО, ВСЕГО ША-
      ГОВ'; X
80 PRINT 'ХОЧЕШЬ ПОИГРАТЬ ЕЩЕ?'
85 PRINT 'ЕСЛИ ДА, ТО НАБЕРИ 1'
86 INPUT A
90 IF A=1 THEN 15
100 PRINT 'ВСЕГО ХОРОШЕГО!'
105 END

```

Проверь свои способности

```
10 LET X=0
20 PRINT 'ПРОФЕССОР ЛЕГ СПАТЬ В
    ВОСЕМЬ ЧАСОВ, А ВСТАЛ'
21 PRINT 'В ДЕВЯТЬ. СКОЛЬКО ЧАСОВ
    СПАЛ ПРОФЕССОР?'
30 INPUT A
40 IF A=1 THEN LET X=X+1
50 PRINT 'НА ДВУХ РУКАХ 10 ПАЛЬ-
    ЦЕВ. СКОЛЬКО НА ДЕСЯТИ?'
60 INPUT B
70 IF B=50 THEN LET X=X+1
80 PRINT 'СКОЛЬКО ЦИФР В ДЮЖИ-
    НЕ?'
90 INPUT C
100 IF C=2 THEN X=X+1
110 PRINT 'СКОЛЬКО НУЖНО СДЕЛАТЬ
    ЗАПИЛОВ, ЧТОБЫ РАСПИЛИТЬ'
111 PRINT 'БРЕВНО НА ДВЕНАДЦАТЬ
    ЧАСТЕЙ?'
120 INPUT M
130 IF M=11 THEN LET X=X+1
140 PRINT 'ВАМ ВРАЧ СДЕЛАЛ ТРИ
    УКОЛА В ИНТЕРВАЛЕ ТРИДЦАТЬ'
150 PRINT 'МИНУТ. СКОЛЬКО ВРЕМЕНИ
    ОН ЗАТРАТИЛ?'
151 INPUT N
160 IF N=30 THEN LET X=X+1
170 PRINT 'СКОЛЬКО ЦИФР 9 В ИНТЕР-
    ВАЛЕ ОТ 1 ДО 100?'
180 INPUT C
190 IF C=1 THEN LET X=X+1
200 PRINT 'У ПАСТУХА БЫЛО 10 ОВЕЦ.
    ВСЕ, КРОМЕ ОДНОЙ, РАЗБЕЖАЛИСЬ.'
201 PRINT 'СКОЛЬКО ОВЕЦ ОСТАЛОСЬ?'
210 INPUT E
220 IF E=1 THEN LET X=X+1
230 IF X=7 THEN PRINT 'ГЕНИЙ'
240 IF X=6 THEN PRINT 'ЭРУДИТ'
250 IF X=5 THEN PRINT 'НОРМАЛЬНЫЙ
    ЧЕЛОВЕК'
260 IF X=4 THEN PRINT 'СПОСОБНО-
    СТИ СРЕДНИЕ'
270 IF X=3 THEN PRINT 'СПОСОБНО-
    СТИ НИЖЕ СРЕДНЕГО'
280 IF X<=2 THEN PRINT 'ВАМ НАДО
    ОТДОХНУТЬ'
290 END
```

«Потопи корабль»

```
5 LET Z=0
10 PRINT 'ВЫ ЛЕТИТЕ НА БОМБАРДИ-
    РОВЩИКЕ НАД МОРЕМ'
20 PRINT 'ВСТРЕЧНЫМ КУРСОМ ИДЕТ
    КОРАБЛЬ ПРОТИВНИКА'
21 PRINT 'ВАМ НУЖНО ЕГО ПОТО-
    ПИТЬ'
22 PRINT 'У ВАС ВСЕГО ДЕСЯТЬ БОМБ'
23 PRINT 'НА КАКОЙ ВЫСОТЕ ПОЛЕ-
    ТИТЕ? (В КМ)'
30 INPUT P
31 IF P<=20 THEN 32
32 IF P>=0.2 THEN 40
33 PRINT 'НА ТАКОЙ ВЫСОТЕ БОМ-
    БАРДИРОВЩИК ЛЕТЕТЬ НЕ МОЖЕТ':
    GO TO 23
35 PRINT
40 PRINT 'НА КАКОЙ СКОРОСТИ ВЫ
    ПОЛЕТИТЕ? (В КМ/Ч)'
50 INPUT V1:PRINT
51 IF V1<=1500 THEN 52
52 IF V1>=900 THEN 60
53 PRINT 'ТАКОЙ СКОРОСТИ У БОМ-
    БАРДИРОВЩИКА НЕТ': GO TO 40
60 PRINT 'ВЫБЕРИТЕ СКОРОСТЬ, С
    КАКОЙ ВАМ НАВСТРЕЧУ ИДЕТ'
61 PRINT 'КОРАБЛЬ ПРОТИВНИКА
    (В КМ/Ч)'
64 INPUT V0
65 IF V0<=60 THEN 70
68 PRINT 'СЛИШКОМ БОЛЬШАЯ СКО-
    РОСТЬ': GO TO 60
70 PRINT 'НА КАКОМ РАССТОЯНИИ ОТ
    КОРАБЛЯ СБРОСИТЕ БОМБУ? (В КМ)'
80 INPUT L
90 LET L1=INT((V0+V1)*SQR(2*
    P/35.3))
100 LET L1=INT(L1*0.1+0.5)/0.1:LET
    Z=Z+1
110 IF Z=10 THEN PRINT 'БОМБЫ ИЗ-
    РАСХОДОВАНЫ. КОРАБЛЬ ЦЕЛ':
    GO TO 145
120 IF L1>L THEN PRINT 'НЕДОЛЕТ.
    ОСТАЛОСЬ БОМБ'; 10-Z: GO TO 70
130 IF L1<L THEN PRINT 'ПЕРЕЛЕТ.
    ОСТАЛОСЬ БОМБ'; 10-Z: GO TO 70
140 IF L1=L THEN PRINT 'КОРАБЛЬ ПО-
    ТОПЛЕН. ИЗРАСХОДОВАНО БОМБ'; Z
145 END
```


Графическая информация на алфавитно-цифровом дисплее

Выпускаемые отечественной промышленностью ЭВМ ДВК-2 поставляются совместно с алфавитно-цифровыми дисплеями (АЦД) 15ИЭ-00-013, предназначенными для ввода—вывода только буквенно-цифровой информации. Это вызывает трудности при организации машинной графики из-за отсутствия графических терминалов. Однако, как показывает опыт вывода графической информации на дисплей, отдельные ограничения можно преодолеть.

В статье (2) говорится, что формирование графической информации на экране АЦД осуществляется с помощью единичных графических элементов (ЕГЭ). Положение такого элемента, определяемое двумя координатами (номером строки и позицией в строке), задается программно без учета функциональных возможностей дисплея и его режимов работы. В этом случае даже простые программы формирования изображений с помощью ЕГЭ становятся громоздкими, требуют использования ряда дополнительных функций и искусственных приемов для подготовки экрана задания законов организации изображений и т. д. При этом функциональные возможности дисплея, упрощающие построение программ, не используются.

Предлагается формирование графических изображений осуществлять с помощью ЕГЭ, используя различные режимы работы АЦД.

Дисплей 15ИЭ-00-013 обеспечивает выполнение двух систем команд № 1 и № 2, которые позволяют управлять маркером (пу-

тем перемещения его по экрану), стирать символ, строку, страницу (3). С практической точки зрения наибольший интерес представляет система команд № 2, обеспечивающая работу под управлением набора командных последовательностей символов. В таблице 1 представлены командные последовательности символов, названия команд и выполняемые при этом операции.

Для организации машинной графики наибольший интерес представляет команда «Прямая адресация маркера», обеспечивающая перемещение маркера в позицию, задаваемую двумя кодами, численные величины которых соответствуют номеру строки и положению маркера в строке. Коды положений маркера для дисплея 15ИЭ-00-013 с размерами экрана 24 строки (25-я строка является строкой управления) по 80 позиций (символов) в строке, представлены в таблице 2.

Рассмотренные наборы командных последовательностей символов можно применять в операторах вывода любого языка программирования (Бейсик, Фортран, Паскаль и т. д.) в функции CHR, которая получает целочисленное значение аргумента и возвращает соответствующую литеру.

В качестве примера рассмотрим возможность организации машинной графики в Бейсик—РАФОСе.

В общем виде оператор вывода на экран дисплея и командная последовательность символов будут:

```
PRINT CHR X (A) CHR X (B),
```

63

Таблица 1

Набор командных последовательностей символов

Командная последовательность	Название команды	Выполняемая операция
(27) (65)	Маркер вверх	Перемещение маркера на одну строку вверх — вниз (позиция маркера сохраняется)
(27) (66)	Маркер вниз	
(27) (67)	Маркер вправо	
(27) (69)	Маркер влево	Перемещение маркера на один шаг по строке вправо — влево
(27) (72)	Переход к системе команд № 1	Осуществляет переход к системе команд № 1
(27) (74)	Возврат	
(27) (74)	Стирание страницы	
(27) (89)	Прямая адресация маркера	Перемещение маркера в начало страницы Стирание текста от позиции маркера до конца страницы Перемещение маркера в позицию, задаваемую следующими двумя символами

Таблица 2

Десятичные коды положений маркера

		Номера	Десятичные коды
Строки		1	32
		2	33
	
		24	55
Позиции в строке		1	32
		2	33
	
		80	111

где А, В — десятичные коды командных последовательностей символов или коды положений маркера (см. табл. 1, 2).

При этом допускается запись в одной строке нескольких командных последовательностей символов.

64

Для выполнения команды «Прямая адресация маркера» необходимо использовать командную последовательность символов (27) (89) и определить на экране дисплея положение маркера в виде кодов номера строки (Y) и номера позиции (X). Тогда команда

```
PRINT CHR (27) CHR (89)
CHR (31+Y) CHR (31+X)
```

произведет перемещение маркера в точку с Y-й строки на X-й позиции ($Y=1,24$; $X=1,80$).

Используя необходимые наборы командных последовательностей символов, можно разработать комплекс программ вывода графической информации на экран дисплея. При проектировании программ машинной графики необходимо предусмотреть, чтобы программными средствами можно было:

- установить режим работы дисплея;
- подготовить экран АЦД для вывода графической символики;

- здать закон перемещения маркера по экрану дисплея;

- переместить маркер в заданную координатами X, Y точку экрана;

- вывести на экран в заданной точке выбранную графическую символику (ЕГЭ).

Под установкой режима работы АЦД предполагается выбор соответствующей системы команд. Для дисплея 15ИЭ-00-013 такую операцию можно произвести программным способом с помощью оператора

```
PRINT CHR (23)
```

и аппаратным способом — заданием в 25-й управляющей строке дисплея требуемого режима работы клавишей «СВД» (более подробно об этом говорится в статье (1)).

Подготовка экрана к работе заключается

в выполнении команд «Возврат» и «Стирание страницы», которые обеспечивают перемещение маркера в начало экрана (точка в левом верхнем углу с координатами (1,1) и стирание текста от позиции маркера до конца страницы. Командная последовательность символов для подготовки экрана к работе совместно с оператором вывода имеет вид:

```
PRINT CHR (27) CHR (72)
CHR (27) CHR (74)
```

Для задания закона перемещения маркера по экрану целесообразно предварительно провести анализ выводимой графической информации.

Вывод горизонтальных отрезков (линий)

Вывод на экран горизонтальных отрезков можно показать на примере построения оси абсцисс с ее последующей оцифровкой. Вывод графической символики оси абсцисс следует производить при фиксированном значении номера строки ($Y=CONST$) и изменяющемся значении номера позиции ($X=2(1)78$ — такая запись означает изменение значения X от 2 с шагом 1 до 78). На последней позиции для наглядности можно отдельным символом вывести стрелку в виде знака «>» и название оси «X». Для оцифровки оси достаточно при фиксированном значении номера строки организовать изменение значения позиции X, например, через 10 позиций ($X=1(10)79$) с последующим выводом числового значения номера позиции. Пример программы вывода горизонтальной оси приведен в приложении 1. В качестве ЕГЭ (выводимого символа) определен знак «—», команда «Прямая адресация маркера» выделена в отдельную подпрограмму.

Вывод вертикальных отрезков (линий)

Вывод на экран вертикальных отрезков можно показать на примере организации оси ординат с ее последующей оцифровкой. Закон перемещения маркера по экрану в этом случае следующий: при фиксированном значении номера позиции ($X=CONST$) изменяется значение номера строки ($Y=1(1)23$); а для оцифровки оси необходимо при фиксированном номере позиции изменять номера строки ($Y=1(5)23$) с выводом числовых (положительных, отрицательных и нуля) значений номера строки. Пример отдельного фрагмента программы вывода вертикальной оси и ее оцифровки дан в приложении 2, где показан закон перемещения маркера по экрану для вывода графической символики (ЕГЭ — буква «l») и оцифровки

оси. Все подготовительные этапы и подпрограмма «Прямая адресация маркера» аналогичны рассмотренной ранее программ в приложении 1, поэтому в дальнейшем не рассматриваются.

Разработанные программы позволяют произвести вывод на экран как отдельных оцифрованных осей (горизонтальной или вертикальной), так и прямоугольной декартовой системы координат или ее частей с равномерной координатной сеткой, номограммы с параллельными шкалами и т. д. для последующего вывода графических зависимостей в функциональных шкалах (4). Вывод графических зависимостей в этом случае не представляет каких-либо принципиальных трудностей: при изменяющихся значениях абсциссы ($X=1(1)79$) производится определение значения ординаты функции, заданной математической формулой, и отображение ее на экране.

Вывод наклонных отрезков (линий)

При организации вывода горизонтальных и вертикальных отрезков (линий) предполагалось изменение одного из параметров (координат) при фиксированном значении другого. На практике часто возникает задача построения наклонных линий, когда закон перемещения маркера по экрану предполагает изменение как одной, так и другой координаты. Например, пусть требуется соединить две заданные точки отрезком прямой линии. В этом случае необходимо предварительно вычислить значения функции (отрезка прямой линии) для промежуточных значений аргумента. При определении значений функции для промежуточных значений аргумента можно использовать различные методы интерполяции, позволяющие с заданной точностью вычислять значение функции (4). В качестве примера в приложении 3 предложен фрагмент программы построения отрезков наклонных линий, в котором использован принцип линейной интерполяции, заключающийся в вычислении значения функции по формуле

$$y = y_1 + \frac{x - x_1}{x_2 - x_1} (y_2 - y_1). \quad (1)$$

В связи с произвольным выбором координат начальной (x_1, y_1) и конечной (x_2, y_2) точек предварительно анализируются значения абсцисс x_1, x_2 и ординат y_1, y_2 для задания положительного $A=1$ (в сторону увеличения) или отрицательного $A=-1$ (в сторону уменьшения) шага параметра цикла. При этом при равенстве абсцисс ($x_1=x_2$) или ординат ($y_2=y_1$) имеют место рассмотренные выше случаи формирования

горизонтальных или вертикальных отрезков линий.

Предложенная программа позволяет организовать вывод горизонтальных, вертикальных и наклонных линий на экран дисплея. Основной ее недостаток тот, что при определенных условиях (углах наклона, близких к 0° и 90°) формируются изображения с неровными краями ступенчатого вида. Это называют лестничным эффектом, или «зазубренностью». Для графических терминалов также характерен подобный эффект, но он менее заметен для пользователя.

В функциональных шкалах, формирование которых было рассмотрено выше, возможно построение различных графических зависимостей и фигур: геометрических (квадратов, треугольников, многоугольников, эллипсов, окружностей), фигур Лиссажу, спиралей, синусоид и т. д. Построение геометрических фигур с помощью отрезков линий не сложно, поэтому подробно рассмотрим построение на экране дисплея эллипса, спирали, фигуры Лиссажу, колебательного процесса.

Для построения эллипса необходим выбор соответствующих значений координаты начальной точки, полуосей A и B , общего количества точек N . Значения функции и аргумента задаются известными формулами:

$$\left. \begin{aligned} x &= x + A \cos\left(\frac{\pi i}{N}\right) \\ y &= y + B \sin\left(\frac{\pi i}{N}\right) \end{aligned} \right\} \quad (2)$$

Где x, y — координаты точек эллипса, i — текущий номер точки. Фрагмент программы построения дан в приложении 4. Выбирая координаты начальной точки, значения полуосей A и общее количество точек, можно добиться качественного формирования фигуры эллипса на экране АЦД.

Формирование спирали на экране АЦД осуществляется по следующему закону перемещения маркера:

$$\left. \begin{aligned} x &= 0,5 A \cos(A) + C \\ y &= A \sin(A) + D \end{aligned} \right\} \quad (3)$$

Где A — параметр ($A=0(0,03)30$); C, D — координаты начальной точки спирали. В качестве примера в приложении 5 предложен фрагмент программы формирования спирали на экране АЦД; целесообразно значения координат начальной точки спирали выбирать следующими:
 $C=13..15; D=40..50$.

Закон перемещения маркера для отображения фигуры Лиссажу имеет вид:

$$\left. \begin{aligned} x &= M_1 \cdot \cos(K \cdot A + F_1) + N_1 \\ y &= M_2 \cdot \cos(L \cdot A + F_2) + N_2 \end{aligned} \right\} \quad (4)$$

Для получения качественного изображения фигуры Лиссажу на экране АЦД рекомендуется параметры в выражении (4) выбирать следующими:

$$M_1=10; M_2=20; L=1; K=4; N_1=13; N_2=40.$$

Представляет значительный интерес формирование на экране дисплея колебательного процесса. Задавая различные законы перемещения маркера по экрану, можно отобразить периодические колебательные процессы, затухающие колебательные процессы и т. д. В качестве примера исследовано формирование затухающего колебательного процесса, описываемого формулой

$$y=10 \cdot \text{COS}(0,25 \cdot x-1) \cdot e^{-\frac{x}{50}}. \quad (5)$$

Реализация программы формирования затухающего колебательного процесса не представляет труда, поэтому программа не приводится.

66

Используя изложенные выше принципы организации вывода графической информации на экран дисплея, можно произвести определение экстремальных значений функции и вывод этих значений на график; возможен вывод на экран одновременно нескольких графиков с различной графической символикой — ЕГЭ.

Как показывает опыт внеклассной работы со старшеклассниками в Рязанском радиотехническом институте, наилучшим образом отображаются геометрические фигуры с вертикальными и горизонтальными отрезками прямых линий. Основными недостатками при этом являются невысокая разрешающая способность АЦД, ограниченный набор ЕГЭ, лестничный эффект.

Рассмотренный подход организации вывода графической информации на экран АЦД позволил школьникам под руководством авторов статьи разработать набор простей-

ших графических программ, предложенных в приложениях, и достаточно сложных демонстрационных, обучающих и контролируемых программ. В процессе работы были разработаны программы «мультипликации» — перемещения графической символики (ЕГЭ) в различных условиях (в многоугольниках, лабиринтах и т. д.), отлажена демонстрационная программа «Стек» для знакомства с работой стековой памяти. Завершается отладка контролируемых программ по основам информатики и вычислительной техники, географии, физике. Применение отдельных программ при изучении этих примеров обеспечили глубокое их усвоение учащимися. Отдельные программы были представлены в виде докладов на ХХХIV студенческой научно-технической конференции, посвященной 117-й годовщине со дня рождения В. И. Ленина, на секции «Информатика и вычислительная техника в школе», в Рязанском радиотехническом институте. Эти программы рекомендованы к внедрению в учебный процесс средних школ и техникумов.

Л и т е р а т у р а

1. *Архангельский А.* Мир ЭВМ // Информатика и образование. 1987. № 3. С. 67—71.
2. *Буланова Н., Волков Д., Широков П.* Графика на алфавитном дисплее / Информатика и образование. 1987. № 3. С. 39—45.
3. Дисплей алфавитно-цифровой 15ИЭ-00-013: Инструкция по эксплуатации ЩЦМ 3.778.012 ИЭ. 1980.
4. *Пулькин С. П.* Вычислительная математика: Пособие для учащихся IX—X классов по факультативному курсу. М., 1974.

ПРИЛОЖЕНИЕ 1	170 PRINT X-1
	180 NEXT X
ПРОГРАММА ОРГАНИЗАЦИИ ОЦИФРОВАННОЙ ГОРИЗОНТАЛЬНОЙ ОСИ	190 STOP
10 REM УСТАНОВКА РЕЖИМА РАБОТЫ АЦД	1000 REM ПРЯМАЯ АДРЕСАЦИЯ МАРКЕРА
20 PRINT CHR\$(23)	1010 PRINT CHR\$(27)CHR\$(89)
30 REM ПОДГОТОВКА ЭКРАНА АЦД	1020 PRINT CHR\$(Y+31)CHR\$(X+31);
40 PRINT CHR\$(27)CHR\$(72)	1030 RETURN
50 PRINT CHR\$(27)CHR\$(74)	1040 END
60 REM ФОРМИРОВАНИЕ ГОРИЗОНТАЛЬНОЙ ОСИ	
70 LET Y=10	
80 FOR X=1 TO 70	ПРИЛОЖЕНИЕ 2
90 GOSUB 1000	
100 PRINT "-"	ФРАГМЕНТ ПРОГРАММЫ ОРГАНИЗАЦИИ ОЦИФРОВАННОЙ ВЕРТИКАЛЬНОЙ ОСИ
110 NEXT X	
120 GOSUB 1000	
130 PRINT ">";"X"	10 REM ФОРМИРОВАНИЕ ВЕРТИКАЛЬНОЙ ОСИ
140 REM ОЦИФРОВКА ОСИ	20 LET X=2
150 FOR X=1 TO 79 STEP 10	30 FOR Y= 1 TO 23
160 GOSUB 1000	40 GOSUB 1000

```

50 PRINT "I"
60 NEXT Y
70 REM ОЦИФРОВКА ОСИ
80 LET X=1
90 FOR Y=2 TO 23 STEP 5
100 GOSUB 1000
110 PRINT (12-Y)
120 NEXT Y
130 LET X=1
140 LET Y=1
150 GOSUB 1000
160 PRINT "/!\\"
170 STOP

```

ПРИЛОЖЕНИЕ 4

ФРАГМЕНТ ПРОГРАММЫ ФОРМИРОВАНИЯ ЭЛЛИПСА

```

10 REM ФОРМИРОВАНИЕ ЭЛЛИПСА
20 PRINT "ВВЕДИТЕ ЗНАЧЕНИЯ ПОЛУОСЕЙ А И В"
30 INPUT A,B
40 PRINT "ВВЕДИТЕ НАЧАЛЬНУЮ ТОЧКУ ФИГУРЫ"
50 INPUT X0,Y0
60 PRINT "ВВЕДИТЕ ОБЩЕЕ ЧИСЛО ТОЧЕК ФИГУРЫ"
70 INPUT N
80 FOR I=0 TO 2*N
90 LET X=X0+A*JCOS(3.14*I/N)
100 LET Y=Y0+B*JSIN(3.14*I/N)
110 GOSUB 1000 \ PRINT ","
120 NEXT I
130 END

```

ПРИЛОЖЕНИЕ 3

ФРАГМЕНТ ПРОГРАММЫ ВЫВОДА НАКЛОННЫХ ЛИНИЙ

```

10 REM ВЫВОД НАКЛОННЫХ ЛИНИЙ
20 PRINT "ВВЕДИТЕ КООРДИНАТЫ 1-Й ТОЧКИ"
30 INPUT X1,Y1
40 LET X=X1 \ LET Y=Y1
50 GOSUB 1000 \ PRINT ","
60 PRINT "ВВЕДИТЕ КООРДИНАТЫ 2-Й ТОЧКИ"
70 INPUT X2,Y2
80 LET X=X2 \ LET Y=Y2 \ LET A=1
90 GOSUB 1000 \ PRINT ","
100 IF X1=X2 THEN 180
110 IF X1>X2 THEN LET A=-1
120 FOR X=X1 TO X2 STEP A
130 REM ЛИНЕЙНАЯ ИНТЕРПОЛЯЦИЯ
140 LET Y=Y1+(X-X1)*(Y2-Y1)/(X2-X1)
150 GOSUB 1000 \ PRINT ","
160 NEXT X
170 END
180 IF Y1>Y2 THEN LET A=-1
190 FOR Y=Y1 TO Y2 STEP A
200 LET X=X1
210 GOSUB 1000 \ PRINT ","
220 NEXT Y
230 STOP

```

ПРИЛОЖЕНИЕ 5

ФРАГМЕНТ ПРОГРАММЫ ФОРМИРОВАНИЯ СПИРАЛИ

```

10 REM ФОРМИРОВАНИЕ СПИРАЛИ
15 INPUT X0,Y0
20 FOR A=0 TO 20 STEP 0.03
30 LET X=A*JCOS(A)+X0
40 LET Y=0.5*A*JSIN(A)+Y0
50 GOSUB 1000 \ PRINT ","
60 NEXT A
70 STOP

```

М. АЛЕКСЕЕВ, Т. АЛЕКСЕЕВА

Реализация Е-практикума на «Искре-1256»

В 1986 г. в школах № 10 и 18 г. Миасса шефы установили семь микро-ЭВМ «Искра-1256». Для обслуживания техники, помощи учителям, организации кружков и создания учебных программ молодые инженеры и педагоги — энтузиасты нового дела — объединились в общественное конструкторское бюро по компьютеризации школ. Одна из первых работ ОКБ — интерпретатор учебного русского алгоритмического языка с графическими командами и простой системой управления.

В комплект «Искры-1256» кроме интерпретирующего процессора с алфавитно-цифровым дисплеем, клавиатурой и встроенным

кассетным магнитофоном также входят обычно мозаичное устройство печати и планшетный потенциометрический графопостроитель (ГП). В ПЗУ защита единственная для «Искры-1256» операционная система с интерпретатором языка, представляющего собой русскую версию Бейсика, и средствами редактирования и отладки программ на этом языке. Режим работы в кодах отсутствует, но в языке есть богатые возможности работы с внешними устройствами.

При создании системы вначале планировалось реализовать хотя бы графические команды управления пером ГП, подобные операторам языка Лого. Отсюда название

системы — ГРУША — графика учебная школьная. Затем удалось ввести практически все основные команды и структуры русского алгоритмического языка (РАЯ). При этом система вплотную приблизилась к ограничениям по объему оперативной памяти и времени бессбойного считывания такой большой программы магнитофоном.

Минимальный объем ОЗУ для ГРУШИ — 16К, программа из 1100 операторов «Искры-125б» вводится около двух минут. Наличие устройства печати и ГП не необходимо, но желательно для полного использования всех возможностей системы.

Система ГРУША представляет собой диалоговую программу для «Искры-125б», обрабатывающую текстовые строки с командами РАЯ и интерпретирующую эти команды. Работая в диалоге с ней, можно ввести с клавиатуры или считать с МЛ один или несколько «алгоритмов» РАЯ с автоматической нумерацией строк, просмотреть их на экране, отредактировать, отпечатать на

бумаге, записать на МЛ и, наконец, исполнить, наблюдая пооператорную прокрутку на экране и, для графических «алгоритмов», движение пера ГП по полю рисунка. Система сама определяет готовность устройства печати и выводит тексты «алгоритмов» и протоколы прокрутки на бумагу, если печать включена, или на экран, если выключена. Всегда можно выйти в меню возможностей системы или вызвать краткий справочник по РАЯ.

Важная особенность системы заключается в том, что для переходов из режима в режим и управления сервисом нет никаких специальных команд. Для этого служат функциональные клавиши верхнего ряда клавиатуры, вызывающие соответствующие подпрограммы, а подробности ГРУША уточняет вопросами и подсказками. Вдоль ряда функциональных клавиш помещается специальная планка с соответствующими надписями, и в меню возможностей рядом с номерами клавиш даются краткие пояснения.

68

ГРУША — ГРАФИКА УЧЕБНАЯ ШКОЛЬНАЯ

(С) МИАСС—ШКОЛА 18

ВЕРСИЯ 23.09.86

СИСТЕМА УПРАВЛЯЕТСЯ ФУНКЦИОНАЛЬНЫМИ КЛАВИШАМИ ДАЛЬНЕГО РЯДА:

- | | |
|---|---------------------------------|
| 1 — МЕНЮ — ПОДСКАЗКА | 10 — ОТМЕНИТЬ ГРУППУ КОМАНД |
| 2 — ПЕРО УБРАТЬ (ПОКАЖИ) | 15 — ИЗМЕНИТЬ ФОРМАТ ЭКРАНА |
| 3 — ВЫДАТЬ ПРОГРАММУ
НА ЭКРАН ИЛИ БУМАГУ | 16 — СПРАВОЧНИК |
| 4 — ИСПОЛНИТЬ ПРОГРАММУ | 18 — (3—) БЕЗ ОТСТУПОВ |
| 5 — ПЕРО — В НАЧАЛО | 19 — (3—) — КОМПАКТНО |
| 6 — ПЕРО ОПУСТИТЬ | 20 — (19—) — БЕЗ НОМЕРОВ КОМАНД |
| 7 — ПЕРО ПОДНЯТЬ | 26 — МЛ НА 1 ЗАПИСЬ НАЗАД |
| 8 — ВВЕСТИ ИЛИ ВСТАВИТЬ КОМАНДУ | 27 — МЛ НА 1 ЗАПИСЬ ВПЕРЕД |
| 9 — ИСПРАВИТЬ КОМАНДУ | 28 — ЗАПИСАТЬ ПРОГРАММУ НА МЛ |
| | 29 — СЧИТАТЬ ПРОГРАММУ С МЛ |

ЖЕЛАЕМ УСПЕХОВ!

Большинство возможностей из меню понятны без комментариев. Опасные для неопытного пользователя функции работы с МЛ (ф26—ф29) намеренно вынесены на верхний регистр, требующий нажатия двух клавиш. «ФОРМАТ» — переключение экрана из режима «16 строк по 64 символа» в режим «8 строк по 32 крупных символа» и обратно, например для демонстрации на уроке. «В НАЧАЛО» — начальная установка ГП: перо — в левом нижнем положении, текущее направление движения — вверх. «ПОКАЖИ» — убирает перо в правый верхний угол, освобождая поле рисунка. Команды ПОКАЖИ и В НАЧАЛО можно использовать в программах как встроенные «вспомогательные алгоритмы». Встроены также стандартные математические функции: SIN, COS, TG, ARCSIN, ARCCOS, ARCTG, LN и EXP. Возведение в степень обозначается символом «Λ», неравенство — «#». Программа на РАЯ может состоять из не-

скольких «алгоритмов» общим объемом до 200 операторов. Переменные, обозначаемые латинскими буквами, локальны в «алгоритме», обмен данными производится через аппарат формальных-фактических параметров в заголовках «алгоритмов» (до 10 параметров), предусмотрена многократная вложенность вызовов «алгоритмов» и циклов, рекурсии не допускаются.

Всю программу или отдельные «алгоритмы» можно распечатать разными способами: ф3 — в столбик по оператору в строке с выделением структур отступами, ф18 — простым столбиком без отступов, а для экономии бумаги или при необходимости отобразить на экране вместе много операторов — ф19 или ф20, при этом система выводит, разделяя пробелами, несколько операторов в строке. Печать на бумагу идет непрерывно, а на экране текст как бы листается клавишей ПУСК.

Для примера представляем программу,

набранную и отредактированную с помощью клавиш ф8, ф9, ф10 и распечатанную нажатием клавиши ф3.

```

1 АЛГ ТЕСТ
2 НАЧ
3   НАПРАВО<60>
4   ВПЕРЕД<200>
5   ЗВЕЗДА<9,30,20,В>
6   ПОКАЖИ
7 КОН
8 АЛГ ЗВЕЗДА<А,В,С,Д>
9 НАЧ
10  ЕСЛИ А>1 И В>5 И С<180.5 ТО
11    РИСУЙ
12    I:=0
13    ПОКА I<А НЦ
14      ВПЕРЕД<В>
15      НАПРАВО<С>
16      НАЗАД<В>
17      НАЛЕВО<360/А+С>
18      I:=I+1
19    КЦ
20    D:=0
21    НЕ РИСУЙ
22    ИНАЧЕ
23      D:=999
24    ВСЕ
25 КОН

```

Головной «алгоритм» ТЕСТ выводит перо ГП на середину поля, вызывает «вспомогательный алгоритм» ЗВЕЗДА, а затем отводит перо, освобождая поле. ЗВЕЗДА вычерчивает А-конечную звезду с длиной стороны зубца В мм и углом при вершине С градусов. Выходной параметр D равен 0 при нормальном исполнении и 999 — при неверном задании параметров.

Нажав клавишу ф4, через пару минут

```

получаем звезду. Во время исполнения
программы на экран или включенное устрой-
ство печати выводится протокол прокрутки,
где для арифметических операторов справа
печатается результат, для команд пера —
значение операнда, а для логических опе-
раторов — 0 или 1 при ложности или
истинности. Конец протокола прокрутки:
17 НАЛЕВО<360/А+С> 60.00
18 I:=I+1 8.000
19 КЦ
13 ПОКА I<А НЦ 1.000
14 ВПЕРЕД<В> 30.00
15 НАПРАВО<С> 20.00
16 НАЗАД<В> 30.00
17 НАЛЕВО<360/А+С> 60.00
18 I:=I+1 9.000
19 КЦ
13 ПОКА I<А НЦ 0.000
19 КЦ
20 D:=0 0.000
21 НЕ РИСУЙ
22 ИНАЧЕ
24 ВСЕ
25 КОН
6 ПОКАЖИ
7 КОН

```

ПРОГРАММА ВЫПОЛНЕНА !!!

Система ГРУША используется в школе № 18 на уроках информатики в девятых классах. Школьники легко и быстро осваивают ее. Изучение РАЯ сопровождается практическими работами на «Искре-1256».

С запросом на передачу системы можно обратиться по адресу: 456321, Челябинская обл., г. Миасс, пр. Октября, 25, средняя школа № 18. Телефоны 2-34-16, 2-28-62.

69

Таллинский научно-учебный центр: программное обеспечение для ПЭВМ

В нашем журнале (№ 6 за 1987 г.) сообщалось о том, что в ТНУЦ можно купить систему «Школьница», предназначенную для эксплуатации на «Агате». Там же можно приобрести и другие программные средства.

Система программирования (СП) «Бейсик» для ПЭВМ «Агат» позволяет получать программы не только на Бейсике, но и на ассемблере «Агата», а также гибридные программы — на Бейсике с включением фрагментов, реализованных на ассемблере. В ней предусмотрено контекстное редактирование (поиск и замена фрагмента текста одновременно по всей программе или

ее части). Файлы на гибком диске, с которыми она оперирует, могут быть: двоичными (программы на ассемблере), программами на Бейсике, текстовыми. Как подмножество в ней содержится язык APPLESOFT для ПЭВМ «Apple».

СП «Бейсик» занимает на гибком магнитном диске (ГМД) 33К байт, для своего функционирования требует не менее 64К байт ОЗУ. Ориентировочная стоимость — 100 руб.

Система численного моделирования (СЧМ) для ПЭВМ «Агат» предназначена для обработки табличной информации) «элек-

тронная таблица») и может использоваться при решении экономических, инженерных и других задач. С ее помощью пользователь, не имеющий специальной математической подготовки, может эффективно выбирать варианты своих действий (в зависимости от исходных данных), не прибегая к формальным методам оптимизации. СЧМ имеет широкий выбор возможностей представления информации на экране дисплея, позволяет сохранять таблицы на ГМД в виде стандартного текстового файла для операционной системы «Агат». Ее зарубежными аналогами являются пакеты Visicalc, Supercalc, Multiplan, от которых она отличается оригинальным способом просмотра альтернативных возможностей исполнения директивы в командном режиме и русскоязычной мнемоникой директив.

70

Для функционирования СЧМ необходимы ПЭВМ «Агат» исполнения 7, имеющая 64К байт оперативной памяти, и принтер типа D-100, CPA 80 или ERSON FX-85. На ГМД она занимает около 30К байт вместе с загрузочным модулем, после включения ПЭВМ загружается автоматически, предоставляет в распоряжение пользователя 31К байт ОЗУ. Ориентировочная стоимость — 100 руб.

Один из форматов файлов, создаваемых СЧМ, может быть прочитан системой «Агат-автор» и текстовым редактором ППП «Школьника».

Система подготовки текстов «Агат-автор» для ПЭВМ «Агат» предназначена для подготовки, редактирования и распечатывания текстов различных видов, которые могут быть введены с клавиатуры или считаны с ГМД. Система отличается от других отечественных текстовых редакторов хорошим описанием, большим набором директив, доступностью для неподготовленного пользователя, значительно облегчает и интенсифицирует работу по выпуску документации, в том числе технической, оформление статей, отчетов и т. п.

«Агат-автор» занимает на ГМД 24К байт, автоматически загружается при включении ПЭВМ. Для функционирования этой системы требуется не менее 96К байт ОЗУ. Ориентировочная стоимость — 100 руб.

Следующие три пакета (система программирования «Бейсик», РЕФОР и РЕБУС) предназначены для ПЭВМ «Роботрон 1715» (СМ 1904).

Система программирования «Бейсик» предназначена для работы с программами на Бейсике, позволяет использовать функции и подпрограммы, производить контекстное редактирование, выполнять программу в режимах интерпретации и компиляции, использовать в Бейсик-программе фрагменты, напи-

санные в машинных кодах, имеет разнообразные возможности для отладки программы и диагностики ошибок. Она работает под управлением операционной системы SCP версии 5.0, требует 64К байт ОЗУ, размещается на одном ГМД типа ЕС 5288 (ИЗОТ 5253Е). Стоимость — 30 руб.

Система подготовки текстов РЕФОР предназначена для подготовки, редактирования, формирования текстов, кроме того, может использоваться для подготовки текстов программ. Она работает в полиэкранном режиме, имеет меню, которое может постоянно находиться на экране; допускает использование строчных и прописных букв русского и прописных букв латинского алфавитов. При распечатке текста РЕФОР позволяет использовать 3 типа шрифта, надстрочную и подстрочную печать, печать с наложением, выделение жирным шрифтом, задавать размеры полей страницы, межстрочный интервал, нумеровать листы, табулировать и т. д. Эта система не требует генерации и настройки, ее может использовать неподготовленный пользователь. Ближайший зарубежный аналог — WORDSTAR.

РЕФОР занимает на ГМД 64К байт, требует 64К байт ОЗУ, работает под управлением операционной системы SCP. Стоимость — 29 руб.

Система управления реляционными базами данных РЕБУС позволяет пользователю описывать структуры своих данных, изменять структуры, вводить и редактировать данные, сохранять их на ГМД, осуществлять поиск и выдачу на печать или дисплей. Во время работы можно вызвать на экран «оперативную помощь» с кратким описанием всех возможностей РЕБУСа. В состав системы входят вспомогательные средства, позволяющие быстро разработать шаблон (форму) для ввода-вывода данных, осуществить сортировку данных по русскому алфавиту.

РЕБУС требует 64К байт ОЗУ, работает под управлением операционной системы SCP, использует дисководы типа MFS 1.2, 1.4, 1.6; позволяет работать одновременно с 16 файлами, помещать в файл до 65535 записей, содержащих до 32 полей. Максимальное число символов в записи — 1000. Вместе с контрольными примерами, демонстрирующими его возможности, РЕБУС занимает 2 ГМД при записи в формате операционной системы SCP 1024×5×40 DDSS. Стоимость — 42 руб.

Для ПЭВМ на базе микропроцессора K1810BM86 (например, ЕС 1840) предназначены пакеты «Текст» и РБД.

Унифицированный функциональный пакет

программ обработки текстов «Текст» позволяет создавать тексты, редактировать их, распечатывать, сохранять на диске (ГМД или типа «Винчестер»), восстанавливать текстовые файлы в аварийной ситуации. При работе используются окна и меню.

«Текст» требует 45К байт ОЗУ, работает под управлением операционной системы МИКРОС-86, занимает на ГМД 74К байт. Ориентировочная стоимость — 300 руб.

Функциональный пакет программ управления реляционной базой данных РБД позволяет создавать базу данных, вносить в нее данные и редактировать их, выводить отчеты о базе, сливать базы данных, осуществлять связь с текстовыми процессорами на уровне передачи данных. Работа с ним возможна как в диалоговом, так и в пакетном режимах.

РБД работает под управлением операционных систем МИКРОС-80 и МИКРОС-86, требует 45К байт ОЗУ, занимает 108К байт на ГМД. Ориентировочная стоимость — 300 руб.

Для ПЭВМ с операционной системой М86 (например, ЕС 1840) с конца 1988 г. будет поставляться система программиро-

вания «Бейсик М86», основу которой составляет расширенная версия Бейсика, согласующаяся с международными стандартами. СП «Бейсик М86» — интерпретатор, содержащий не только традиционные средства отладки и исполнения программ, но и экранный редактор текста. Она позволяет работать со строками символов, целыми числами, дробными числами с простой и двойной точностью, файлами с произвольным доступом и последовательными, обращаться к портам ввода-вывода, к любому байту основной памяти, обрабатывать прерывания от таймера, ошибочные ситуации и некоторые другие события, вызывать подпрограммы, написанные на машинном языке. С ее помощью возможна организация телеобработки данных. Графические возможности Бейсика М86 также шире обычных.

СП «Бейсик М86» требует 70К байт ОЗУ в дополнение к памяти, используемой операционной системой, и один накопитель на ГМД. Ориентировочная стоимость — 300 руб.

Для заказа перечисленных программных средств необходимо направить гарантийное письмо по адресу: 200026, Таллин, бульвар Кадака, 165, ТНУЦ. Тел. 68-40-19.

71

И. АНТИПОВ, С. КОЛОБОВ, О. ЕРМОЛАЕВ, В. АНИСИМОВ
НИИ школ МП РСФСР

О методике психологического исследования адаптации школьников к учебной нагрузке при работе на компьютере

Использование ЭВМ в учебном процессе повышает степень умственной, эмоциональной нагрузки, которую испытывает учащийся. Комплексное исследование нервной, эмоциональной и мотивационно-личностной сферы школьников при работе с компьютером и специфики этой работы позволяет охарактеризовать течение процесса адаптации.

Характер перестройки организма учащихся в условиях адаптации к учебной нагрузке зависит не только от их физического, но и от психологического состояния, от факторов включенности в деятельность и степени заинтересованности в ней, а также организации рабочего места, возможностей вычислительной техники, используемых учебных пособий и т. п.

Разрабатываемый в НИИ школ Минпроса РСФСР комплекс методик исследования про-

цесса адаптации школьников к учебной нагрузке при работе с компьютером включает следующие аспекты: физиологический (общие изменения центральной и периферической гемодинамики и функционального состояния нервной системы); психологический, психофизиологический (особенности мотивационно-личностной сферы учащихся); педагогический (вопросы разнообразия средств общения с ЭВМ).

В процессе нашего исследования обучение школьников осуществляется с использованием учебных пособий, разработанных в НИИ школ Минпроса РСФСР.

На основе полученных результатов оценивается состояние здоровья учащихся на различных этапах их работы с ЭВТ: составляются рекомендации, направленные на интенсификацию учебной деятельности, оптимизацию учебного процесса, исключение

**Анкета для сбора данных медико-психологического контроля
состояния здоровья школьников,**

на основе которой были разработаны программы сбора и обработки данных

Дата _____

Школа № _____

Время _____

Класс _____

Фамилия, имя _____

Оцените свое состояние, подчеркнув соответствующую цифру (одну в каждой строке) между противоположными по смыслу характеристиками в том месте, которое, по вашему мнению, отражает соотношение между этими характеристиками в данный момент времени.

Помните! Плохих и хороших ответов нет. Поэтому отвечайте искренне!

Значения цифровых обозначений: ± 3 — качество выражено очень сильно; ± 2 — качество выражено средне; ± 1 — качество скорее выражено, чем не выражено; 0 — трудно сказать.

1. Настороение хорошее	+3 +2 +1 0 -1 -2 -3	Настроение плохое
2. Сонливость	+3 +2 +1 0 -1 -2 -3	Бодрость
3. Равнодушие	+3 +2 +3 0 -1 -2 -3	Заинтересованность
4. Спокойствие	+3 +2 +1 0 -1 -2 -3	Раздраженность
5. Увлеченность	+3 +2 +1 0 -1 -2 -3	Безучастность
6. Рассеянность	+3 +2 +1 0 -1 -2 -3	Внимательность
7. Сосредоточиться трудно	+3 +2 +1 0 -1 -2 -3	Сосредоточиться легко
8. Напрягаю зрение	+3 +2 +1 0 -1 -2 -3	Зрение не напрягаю
9. Голова не болит	+3 +2 +1 0 -1 -2 -3	Голова болит
10. Устал после работы	+3 +2 +1 0 -1 -2 -3	Не чувствую усталости
11. Недоволен результатом работы	+3 +2 +3 0 -1 -2 -3	Удовлетворен результатом работы

Спасибо!

72

возможности отрицательного влияния ЭВТ на здоровье школьников, усовершенствование учебно-методических материалов.

Исследование психологических особенностей школьников при работе с ЭВМ. Разрабатываемая методика основана на так называемом субъективном шкалировании, т. е. самооценке испытуемым своего состояния при помощи ряда признаков (см. анкету). При этом предполагается, что учащийся способен соотнести свои ощущения с заданной оценочной шкалой.

Все варианты методики составлены таким образом, чтобы позиции, по которым производится оценка учащимся своего состояния, располагались в случайном порядке, т. е. и позитивные, и негативные признаки находятся как справа, так и слева от шкалы баллов. Подобное случайное расположение оценочных позиций препятствует возникновению стереотипа при заполнении тестов-анкет.

Анкета включает три группы признаков (или три фактора). Эти группы признаков мы условно (до проведения анализа экспериментальных данных) определяем следующим образом: характеризующие состояние учащегося до начала работы (признаки 1—4); в процессе работы (5—7); после работы (8—11). Предварительно каждый фактор можно обозначить как: группа А — на-

строение, Б — интерес, В — удовлетворенность результатом. Подобная группировка представленных в анкете признаков позволяет проследить изменение настроения учащихся, изменение выраженности интереса к занятиям, а также степень субъективной удовлетворенности уроком.

Для каждого учащегося и группы школьников подсчитывается различие в отдельных субъективных оценках и суммарное различие оценок состояния до и после урока. Наличие статистически значимых различий как в сторону улучшения, так и в сторону ухудшения состояния учащихся позволяет говорить о надежности теста.

В наших исследованиях предполагается использование пакета компьютерных тестов. Возможности традиционных методик, организованных по типу «бумага — карандаш», явно недостаточны. Использование ЭВМ позволяет решить много проблем, прежде всего проблему стандартизации условий тестирования, проблему эффективной обработки показателей эксперимента, создание банка данных по различным тестовым методикам для каждого испытуемого и т. п. При помощи ЭВМ возможно создание обширного класса тестов, направленных на анализ сложноорганизованных психических процессов в реальном масштабе времени.

Разработанное нами программное обеспечение представляет собой первую очередь автоматизированной системы сбора и обработки данных психологического контроля состояния здоровья учащихся, работающих на ЭВМ. На данном этапе исследования составлены следующие программы:

1. Программа опроса учащихся с предварительной обработкой данных (МЕДИЦИН. ТСТ).
2. Программа обработки данных (анализ данных медицинского контроля всех учащихся одной школы).
3. Программа подготовки диска к вводу данных.
4. Программа просмотра файла данных на экране дисплея.
5. Программа выдачи данных из файла на печать.

Режим эксплуатации программ рассчитан на их включение в реальный учебно-воспитательный процесс средней школы, в частности в процесс обучения информатике (машинный вариант), что обеспечивается функциональным назначением модулей и особенностями реализации диалогового взаимодействия пользователя и персонального компьютера. Пользователями данного программного обеспечения являются проводящий занятия учитель, исследуемый учащийся.

Тексты указанных выше программ приводятся в работе: Антипов И. Н., Колобов С. В., Ермолаев О. Ю., Анисимов В. В. Сбор и обработка данных медико-психологического контроля состояния здоровья школьников при работе с электронной вычислительной техникой (методические рекомендации). М.: НИИ школ МП РСФСР, 1987. Ч. 1.

Диалог ЭВМ с учителем предполагает его минимальное участие в учебной деятельности. Все операции (кроме текущего контроля) могут быть выполнены во внеурочное время. Диалог ЭВМ и учащегося предполагает максимальную «дружественность» и надежную защиту программ и данных.

Последовательность работы такая: перед началом занятия учитель вызывает программу ввода данных, устанавливает текущее время, номер занятия; во время занятия программа ведет диалог с учащимся, который идентифицируется номером школы, классом, подгруппой, номером по журналу (номером рабочего места). По окончании ответов на вопросы введенные данные записываются (по команде учителя) в соответствующий файл на диске. Затем программа переходит к диалогу со следующим учеником. После окончания занятия данные могут быть просмотрены или отпечатаны.

В именах файлов зашифрованы данные о контингенте обследуемых учащихся. Программы ориентированы на обслуживание куста школ, в каждом файле собираются данные о состоянии учащихся одной подгруппы одного класса.

Архитектура программ открытая, набор легко может быть дополнен другими модулями. Оценка эксплуатационных характеристик показала, что программы надежны, обеспечивают хорошее время реакции, достаточно удобны и самодокументированы. Объем оперативной и дисковой памяти используемых ПЭВМ «Ямаха» (алгоритм может быть реализован и на других ПЭВМ) достаточен для проведения обследования учащихся одной школы без использования дополнительной дискеты (все программы пакета занимают 400 строк).

Проблема ремонта: острота не снимается



Школы, техникумы, профессионально-технические училища начинают получать и эксплуатировать вычислительную технику. Понятна тревога руководителей учебных заведений о техническом состоянии компьютеров. Остро стоит эта проблема и в Москве. Наш корреспондент встретился с директором завода по ремонту вычислительной техники С. Г. Гасановым и взял у него интервью.

Кор.: Сергей Гасанович, расскажите, пожалуйста, о вашем предприятии.

С. Гасанов: Наш завод — головное предприятие. Мы ведем гарантийный и послегарантийный ремонт компьютеров для предприятий и организаций. Коллектив у нас сплоченный, текучесть кадров минимальная. Мы сами заботимся о подготовке квалифицированных рабочих для предприятия, заключили договор с московским СПТУ-199, оснастили его техникой на сумму 200 тыс. рублей, направили туда в качестве мастеров своих сотрудников. Организовали обучение рабочих-электромехаников по сервису персональных компьютеров. Фактически училище стало филиалом нашего завода. У нас полное взаимопонимание с руководителями училища. Через некоторое время начнем получать «готовеньких» специалистов для себя. Так что наши затраты, как вы понимаете, имеют «корыстный» интерес.

Кор.: Как складываются ваши взаимоотношения с заводами-поставщиками? Какие проблемы в работе коллектива? Есть ли жалобы от потребителей?

С. Гасанов: Жалобы есть. Но с чем это связано? Вечная беда ремонтников — отсутствие запасных частей, которые нам должны бесперебойно поставлять заводы-изготовители. В первую очередь, нас волнуют персональные компьютеры и КУВТ-86. Бывает, что на наших складах скапливается большое количество этой техники. Что надо исправлять, мы установили, готовы принять срочные меры. А запчастей нет. Получается некрасиво. Тем более что, заключая договоры со школами, СПТУ, мы берем на себя обязательство поддерживать имеющуюся у них технику в рабочем состоянии. А ведь вы понимаете, что если на заводах с электронно-вычислительной техникой работают специалисты, то в школах так бывает не всегда, и машины выходят из строя чаще. В период гарантии мы должны отремонтировать компьютер за пять дней, в послегарантийный период — в течение месяца. Гладко вопрос снабжения запасными частями не проходил никогда. Отдел материально-технического снабжения постоянно напрямую обращается на заводы, выпускающие вычислительную технику, пишем письма в партийные бюро предприятий, которые должны работать с нами. Объясняем важность этих проблем с учетом школьной реформы. Словом, бьем во все колокола. Ведь перед заказчиками отвечаем мы непосредственно.

Кор.: Сейчас будет выпускаться много новых отечественных персональных компьютеров. Как вы готовитесь к этому?

С. Гасанов: В приемочную комиссию на заводах, выпускающих ВТ, обычно включают и наших представителей

Кор.: Какими путями, по-вашему, будет или должно идти снабжение школ вычислительной техникой?

С. Гасанов: Я, возможно, выражу крайнюю точку зрения, но думаю, что насыщать все школы, все учебные заведения компьютерами нецелесообразно. В первую очередь надо было бы обеспечить учебно-производственные комбинаты, дать туда пять, шесть — сколько нужно комплектов. Посмотрите, какая складывается ситуация: КПД вычислительной техники в школе крайне низок, некоторые кабинеты, оборудованные шефами, вообще стоят под замком — такие факты известны. А в учебно-производственных комбинатах и отдала от техники была бы больше, чем в КВТ в школах, и ее эксплуатация велась бы специалистами, что, конечно, приводило бы к поломкам значительно реже. Ведь не секрет, что учителя информатики на данном этапе не имеют, по объективным причинам, базового образования — отсюда зачастую идет непрофессиональное обращение с машинами. Нас, ремонтников, конечно, это не может не волновать.

Кор.: А не собираетесь ли вы при заводе открыть кооператив, связанный с вычислительной техникой?

С. Гасанов: Собираемся, и не один. Первый кооператив организуется по инициативе ГУНО и нашего завода. Идея поддержана академиком Е. П. Велиховым. Под эгидой нашего коллектива создаем кооператив по подготовке учителей информатики (начнем с 12 московских школ). Работать в нем в свободное время будут не только наши сотрудники, но и представители педагогических НИИ. Научно-методическое руководство возьмет на себя Временный творческий коллектив «Школа-1», созданный при Академии наук СССР. Технику выделит наш завод. С развитием дела, возможно, выявятся резервы, о которых мы сейчас не предполагаем.

Кор.: Какие еще кооперативы вы собираетесь открывать? Кто их будет возглавлять?

С. Гасанов: Возглавят наши ведущие механики. Мы уже говорили о том, что для школ большую трудность представляет монтаж электрооборудования. В обязанности завода это не входит. Представьте себе, что вы — директор школы и приобрели комплект вычислительной техники. Сколько возникает проблем! Необходимо сделать заземление, подвести питающий кабель, вывести розетки и т. д. Конечно, если есть сильные шефы, проблем не возникает, но 70 % школ брошены на произвол судьбы. И вы бросаетесь искать подрядчика. Как, где? И вот в Главном управлении народного образования буквально месяц назад возникла идея организовать специальный кооператив по электромонтажу в школах, СПТУ. Мы написали письмо в Моссовет с просьбой разрешить нам создать его. И третий кооператив открываем в Люблинском районе Москвы. Будем ремонтировать вычислительную технику, пишущие машинки, электромusзыкальные инструменты. На 20 000 рублей в год будем оказывать услуги населению. Это тоже важно.

Кор.: Хорошо, что вы проявляете предприимчивость.

С. Гасанов: Сейчас с кооперативами много нервозности, но я думаю, мы наладим дело.

Кор.: Позвольте пожелать вашему коллективу успехов и в основной работе, и в новых начинаниях. Очень ценно, что вы поощряете инициативу, горячо болеете за дело.

С. Гасанов: Спасибо вам. Редакции нового журнала «Информатика и образование» мои самые добрые пожелания — интересной работы, талантливых авторов, увеличения тиража.

Почему низка надежность техники?

Кабинет информатики и вычислительной техники при Карагандинском областном институте усовершенствования учителей образован совсем недавно. Задачи, стоящие перед нами, имеют большую государственную важность. Это обеспечение компью-

терного всеобуча руководящих и педагогических кадров, подготовка учителей информатики, внедрение ВТ в школу, обеспечение школ методическими материалами для работы на компьютерах и многое, многое другое. Как же обстоят дела на данный

момент? В области имеется три компьютерных класса, один из которых — у нас. Обеспечение компьютерной техникой крайне неудовлетворительное. План поставок ЭВМ в область за 1987 г. выполнен лишь на треть — из девяти классов получено три. Но не это даже главная беда, а то, что надежность поставляемой техники крайне низка. Вот несколько примеров.

КУВТ-86, установленный в нашем институте полгода назад, работал практически только месяц. Уже при монтаже три БК-0010 оказались неисправными. А ДВК-2М вообще почти не работал, постоянно в ремонте.

Совсем недавно мы получили компьютеры «Агат» Лианозовского электромеханического завода, класс, состоящий из 13 ЭВМ. Оказалось, что пять из них неисправны. Мы сразу же обратились на завод с письмом о необходимости гарантийного ремонта. Нас переадресовали в ПО Алма-атинского ВТИ, которое в свою очередь адресует в центр ВТИ в нашем городе. А специалистам нашего центра ВТИ явно не хватает квалификации, да и элементарная техническая база для ремонта отсутствует. И стоят ЭВМ мертвым грузом. Спрашивается: почему заводы-изготовители не несут ответственность за свою бракованную продукцию, почему кто-то должен за них ее ремонтировать, еще до эксплуатации. Заводы пользуются тем, что слишком велика потребность в ком-

пьютерах, и сбывают все что попало, зная заранее, что потребитель будет все равно рад, даже браку. Разве можно в таких условиях успешно организовать качественный, стабильный учебный процесс?

Учебные заведения вынуждены покупать заводомый брак, надеясь какими-либо путями отремонтировать компьютеры.

Совершенно неудовлетворительно решается вопрос об обеспечении компьютеров программными средствами. Лианозовский завод, например, поставляет «Агаты» с десятью чистыми гибкими магнитными дисками. При заводе имеется информационный центр, где осуществляется запись программ для «Агата» на диски заказчика. Стоимость только записи системы «Школьника» на 10 дисков заказчика составляет 1550 руб. В каждом информационном центре — свои расценки на программы: от 25 до 150 руб. за диск. Но рекорд принадлежит все же Лианозовскому заводу.

Как же нам быть? Кто поможет школе решать проблемы гарантийного ремонта, сервисного обслуживания, программного обеспечения?

Хочется, чтобы компьютеризация школ не тормозилась заводами-изготовителями, низкой надежностью выпускаемой техники.

Л. КОЙСИНА,
методист кабинета ИВТ ОИУУ
г. Караганда

76

От редакции. Предлагаем нашим читателям обсудить проблемы ремонта вычислительной техники. Эта анкета, как мы считаем, поможет выявить «болевые» точки, связанные с обслуживанием ЭВМ.

Итак, просим указать:

1. Какими ЭВМ (тип техники) оснащено ваше учебное заведение?
2. Комплектность поставки (наличие дисководов, число полученных дисков, тип печатающего устройства).
3. Завод-изготовитель.
4. Когда к вам поступила эта техника?
5. Когда обратились в сервисную организацию?
6. Имеется ли у вас договор с какой-либо организацией на ремонт и обслуживание машин?
7. Когда (дата) были запущены (фактически начали работать) ЭВМ?
8. Укажите причину возникшей неисправности.
9. Назовите реальные сроки проведения ремонта с момента подачи заявки в сервисную организацию.
10. Какие у вас есть претензии по ремонту ВТ?
11. Каковы ваши предложения по улучшению дел в этой области?

Просим также указать адрес и телефон ответственного лица, а на конверте сделать пометку: «Ремонт ВТ».

Журнал в журнале
для школьников, студентов,
учащихся СПТУ и техникумов
Издается при участии ЦК ВЛКСМ

Молодежная инициатива

2

В НОМЕРЕ:

КВН «НАЖМИ КНОПКУ ЭВМ!»

«ЭЛЕКТРОНИКА» ПРИГЛАШАЕТ

НОВЫЙ ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР

ЗДРАВСТВУЙ, ПЕТЯ БЕЙСИКОВ!

САМОЕ ПРЯМОЕ ОТНОШЕНИЕ

КОНКУРС ПРОГРАММ ЗАВЕРШЕН

**А. ШЕНЬ
НЕ ПОГИБНУТЬ В БОЛОТЕ ДИЛЕТАНТИЗМА**

КВН «Нажми кнопку ЭВМ!»



78

Свои рекомендации по проведению встреч клуба веселых и находчивых по информатике предлагает преподаватель Брянского техникума советской торговли Валентина Михайловна Гончаренко.

Популярные встречи клуба веселых и находчивых подсаказали нам такую форму внеклассной работы, как КВН по информатике. Эти встречи можно проводить между командами, составленными из учащихся одной группы, класса, параллельных групп, классов, между командами техникумов, школ, училищ, с приглашением болельщиков.

Программа КВН должна быть разнообразной как по форме, так и по содержанию. При этом необходимо, чтобы интересное, забавное, занимательное, с чем учащиеся знакомятся на вечерах, соответствовало содержанию курса информатики. Следует тактично, но достаточно настойчиво и последовательно подчеркивать тесную связь между программным материалом и вопросами, затронутыми на встрече.

Основная цель встречи клуба веселых и находчивых — повышение интереса к новому предмету, поэтому желательно привлечь к ее подготовке и проведению как можно больше учащихся.

Подготовку к КВН мы рекомендуем начинать за полтора-два месяца до встречи. Для руководства всей подготовительной работой создается оргкомитет, в который входят преподаватель информатики и несколько учащихся. Ребятам нужно предоставить возможность проявить самостоятельность и инициативу при составлении плана встречи, разработке сценария, подготовке выступлений.

Конечно, каждой команде необходимы название, гимн, девиз, эмблема.

Составляя текст приветствия команде противника, болельщикам, жюри, не забудьте о шутливых пожеланиях, остроумных замечаниях и т. п.

Каждая команда должна подготовить домашнее задание, например, на такую тему: «Придумать сказку или рассказ, в которых главными действующими лицами являются единичка и нолик».

Можно подготовить световую газету, состоящую из нескольких юмористических рисунков, сделанных на прозрачной пленке. Каждый рисунок сопровождается эпиграмма, дружеский шарж, пожелание нерадивым ученикам или похвала активистам. Световая газета демонстрируется с помощью кодоскопа.

Оргкомитет по проведению КВН должен позаботиться о соответствующем оформ-

лении помещения, в котором будет проходить встреча. Здесь будут уместны и стенгазеты, и выставка творческих работ, и лозунги, рисунки, плакаты различного содержания, и задачи, софизмы.

Чтобы КВН прошел в хорошем темпе, организованно, не забудьте подготовить доску, мел, тряпку, бумагу, карандаши, кодоскоп, фломастеры, расставить мебель, подготовить места для болельщиков.

Очень полезна будет выставка книг и журналов. Некоторые издания можно раскрыть на той странице, где имеется оригинальное высказывание, интересная мысль, любопытный факт, увлекательная задача. Не нужно размещать экспозицию в закрытых стендах. Ее следует разместить так, чтобы учащиеся могли свободно полистать любую книгу, любой журнал. Дополнением к выставке может быть фотомонтаж, посвященный какому-то одному вопросу, например «История кибернетики» или «Норберт Винер — основоположник кибернетики». Его сопровождает краткая биография или сообщение.

Болельщики, конечно, должны приготовить красочные плакаты с пожеланиями своей команде. Необходимо подумать и о приглашенных билетах для преподавателей и членов жюри, об оформлении объявления о встрече.

Предлагаем примерный план сценария КВН: представление команд и жюри; клятва судей; приветствие командам и жюри; размин-

ка команд (эстафета); разминка капитанов; конкурсы писателей, программистов, художников; конкурс «Управляемый карандаш»; конкурсы команд, болельщиков; машинная музыка; фокус; проверка домашнего задания; машинные «шедевры»; конкурс капитанов; подведение итогов, награждение победителей.

Ведущие КВН не только объявляют очередные выступления, но и исполняют некоторые интермедии шуточного характера.

Разминку команд можно провести в виде эстафеты, предложив такие вопросы: кто умнее: человек или компьютер? Кто сильнее: человек или робот? Если робот принял человеческий облик, по каким качествам можно отличить от него человека? В каких, на ваш взгляд, видах спорта победу одержат люди, а в каких — роботы?

В разминке капитанов числа, заданные в двоичной системе счисления, записать в десятичной.

В конкурсе команд можно включить такие вопросы: Первая ЭВМ. В каком году? Где? Первая ЭВМ в Советском Союзе. Поколения ЭВМ. Устройство ЭВМ. Что такое АСУ? Что такое кибернетика?

В конкурсе писателей могут быть предложены самые разнообразные темы, например «Написать рассказ о развитии ЭВМ на планетах, населенных разумными существами».

Конкурс программистов предполагает составление различных программ.

Задания к конкурсу «Управляемый карандаш» можно найти в журнале «Информатика и образование» (№ 3, 1986 год).

После рассказа ведущего о двоичной системе счисления участникам встречи предлагаются два задания.

Фокус 1.

Угадывание предмета по таблицам

Ведущий. Вы видите перед собой различные геометрические фигуры и инструменты. Вот их названия:

1. Куб.
2. Шар.
3. Окружность.
4. Круг.
5. Линейка.
6. Цилиндр.
7. Циркуль.
8. Треугольник.
9. Квадрат.
10. Параллелограмм.
11. Сегмент.
12. Транспортир.
13. Сектор.
14. Пирамида.
15. Трапеция.

Они же выписаны в этих четырех таблицах:

Таблица 4

Треугольник
Трапеция
Сектор
Пирамида
Транспортир
Квадрат
Параллелограмм
Сегмент

Таблица 3

Круг
Линейка
Циркуль
Сектор
Транспортир
Трапеция
Пирамида
Цилиндр

Таблица 2

Шар
Окружность
Цилиндр
Циркуль
Трапеция
Пирамида
Сегмент
Параллелограмм

Таблица 1

Куб
Сектор
Трапеция
Сегмент
Окружность
Линейка
Цилиндр
Квадрат

Выберите любой из этих предметов так, чтобы я не видел. С помощью несложных расчетов можно установить, какой предмет выбран. Кто желает проделать этот фокус?

Допустим, что к доске выходит ученик М. Ведущий поворачивается так, чтобы видеть только таблицу, в которой 15 названий. Затем он продолжает: «Выбери, М., любой из предметов на столе. Подними его так, чтобы видели все, кроме меня. Записан ли этот предмет

в таблице 1?» — «Да». — «А в таблице 2?» — «Нет». — «А в таблице 3?» — «Да». — «А в таблице 4?» — «Нет». Ведущий: «Я угадываю: ты выбрал линейку».

Объяснение. Каждому предмету соответствует определенный номер, под которым название предмета значится в таблице. Например, сектору соответствует номер 13, линейке — 5. Переведем все эти номера в двоичную систему счисления. Тогда каждое из чисел записывается не более чем четырьмя цифрами. Например, число 5 запишем так: 101, или 0101.

В первой таблице помещаются только такие названия, чьи номера в двоичной системе счисления имеют на первом месте справа 1. Например, слову «сектор» соответствует число 13, а в двоичной системе счисления — 1101; на первом месте справа — 1; поэтому слово помещено в таблицу 1. В таблицу 2 помещены те слова, чьи номера в двоичной системе счисления имеют на втором месте (считая справа налево) цифру 1. Например, слово «цилиндр» имеет номер 7, т. е. в двоичной записи — 111. Вторая цифра с конца (справа) — 1. Следовательно, слово «цилиндр» включаем в таблицу 2. Аналогично составлены таблицы 3 и 4. Когда М. говорит, что выбранный им предмет имеется в таблице 1 и 3, но не значится в таблицах 2 и 4, можно написать номер этого предмета в двоичной системе счисления: $0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2 + 1$, т. е. 5. Под номером 5 в таблице из 15 предметов значится слово «линейка». Значит, М. выбрал линейку.

Фокус 2. Угадывание целого числа от 1 до 31 с помощью двоичной системы счисления

Пусть этот фокус проводят два ученика — А. и Б. Б. выходит из комнаты, А. вызывает к столу 5 учеников и выстраивает их в один ряд. Затем он предлагает присутствующим назвать любое число от 1 до 31. В уме он переводит число в двоичную систему счисления и расставляет учащихся так, чтобы нулю соответствовал ученик, стоящий лицом к присутствующим, а единице — ученик, стоящий несколько боком к аудитории. Например, если предложено число 13, то в двоичной системе счисления оно запишется так: 1101, или 01101. Затем А. уходит в сторону (или вовсе выходит из комнаты). Приглашают Б., и он, посмотрев на пятерку учеников, восстанавливает в уме по их расположению загаданное число (сначала в двоичной системе, а затем переводит его в десятичную).

Можно видоизменить этот эффектный фокус: например, ставить вместо нулей и единиц мальчиков и девочек или же вместо нуля ставить книгу лицевой стороной, а вместо единицы ставить книгу обратной стороной к зрителю.

Конкурс капитанов можно провести в виде игры.

Оба капитана выбирают какое-то число n (при условии, что n не кратно 11). Например, $n=100,380$ и т. д. Потом первый может назвать любое число, но не больше, чем 10. После этого второй игрок увеличивает это число, но не больше, чем на 10; затем первый увеличивает число, названное вторым, но не больше, чем на 10, и т. д. Выигрывает тот, кто первый назовет число n . Придумайте алгоритм, кото-

рый позволил бы первому игроку выиграть.

Рекомендуемая литература

Балк М. Б., Балк Г. Д. Математика после уроков. М., 1971.

Информатика и образование. 1986. № 3; 1987. № 3, 4.

Пекелис В. Д. Кибернетическая смесь. М., 1982.

Савицкая Э. М. Внекласс-

ная работа по математике: Методические рекомендации по математике. М., 1982.

Самойленко П. И., Сергиенко Л. Ю. Внеклассная работа: Методические рекомендации по организации кабинета математики с типовым перечнем оборудования. М., 1984.

Чистяков В. Д. Математические вечера в средней школе. М., 1958.

«Электроника» приглашает

В пяти залах Московского центра «Электроника», оборудованного отечественными ЭВМ ДВК-2, БК-0010, Электроника-85, посетители имеют возможность выполнить математические, экономические и другие расчеты, составить и распечатать программу, приобрести носители информации и записать на них собственную программу.

Современному школьнику или специалисту общение с компьютером не кажется таким сложным, как, например, человеку с гуманитар-

ным образованием, да к тому же закончившему школу лет пять назад, когда в учебной программе еще не было такого предмета, как информатика. Таким посетителям в центре могут предложить квалифицированные консультации по информатике и вычислительной технике. ЭВМ поможет и тем, кто работает над курсовой или дипломом. Кроме того, в центре «Электроника» просто приятно провести время. После общения с компьютером можно посмотреть «мультики» в видеосалоне,



выпить чашку кофе с пирожным в маленьком уютном баре, оборудованном цветомузыкой. А, скажем, семейный вечер в компьютерном центре будет способствовать решению проблемы общения отцов и детей. Ведь, что говорить, компьютеризация школы дала многим родителям почувствовать, что здесь они отстают от своих детей.

— Судя по первым дням работы, наш центр весьма популярен. Все 35 компьютеров у нас сегодня заняты, — рассказывал директор центра Геннадий Тимофеевич Крайнев, который в этот день был прямо-таки нарахват: учительница, приехавшая сюда с другого конца Москвы, требовала директора, чтобы договориться о проведении здесь урока по информатике, кто-то интересовался, на чье имя писать заявление на курсы обучения языкам программирования, которые скоро откроются в центре, и т. д.

— Министерство электронной промышленности, которому мы обязаны своим существованием, в скором времени планирует открытие еще двух центров, подобных нашему, но более крупных, — в Москве и в Зеленограде.

А нам предстоит опробовать экономическую сторону дела.

Адрес центра: Москва, ул. Раменки, 12. Телефон: 931-00-03. Проезд до ст. метро «Проспект Вернадского», далее авт. 715 до ост. «Магазин «Спорт».

Т. КРАСНОВА,
студентка
ф-та журналистики МГУ

Новый персональный компьютер

За короткий срок (практически за год) временный творческий молодежный коллектив, созданный на Пензенском заводе ВЭМ, подготовил к серийному производству бытовую персональную ЭВМ «Сура». Она предназначена для массового применения в учреждениях, организациях, в домашних условиях. Предусмотрено два комплекта поставки: минимальный и расширенный. Минимальный комплект состоит из клавиатуры со встроенной микро-ЭВМ и набором проводов для подключения к бытовому черно-белому или цветному телевизору и кассетному магнитофону. В расширенный

комплект дополнительно входит цветной телевизор и кассетный магнитофон.

Объем оперативной памяти компьютера — 64К, объем постоянной памяти — 32К. Программное обеспечение предусматривает ленточную оперативную систему, интерпретатор с языка Бейсик, транслятор с языка Форт. Расширение функций обеспечивается путем подключения печатающего устройства, накопителя на гибких магнитных дисках, графопостроителя, манипулятора для игр, звукогенератора, спецплат для создания локальной сети из нескольких персональных ЭВМ. Выпуск «Суры» начат.

81

Здравствуй, Петя Бейсиков!

Школьники, учащиеся и студенты, решившие самостоятельно познакомиться с программированием, прочитав эту книгу, смогут составлять несложные программы, овладеют терминологией и идеями, необходимыми для дальнейшего знакомства с информатикой, найдут в книге клад ценных практических советов — «секретов».

Педагоги, творчески решающие проблемы обучения, тоже не останутся равнодушными. Занимательная, написанная в научно-популярном стиле книга по сути является умным и грамотным методическим пособием по информатике с прекрасными примерами и иллюстрациями. Ее содержание можно использовать при объяснении, для изготовле-

ния плакатов во внеклассной работе.

А родители вместе с детьми могут рассматривать картинки, читать отдельные страницы, главы или всю книгу. Успешный контакт с миром информатики и собственными детьми вам обеспечен.

Эту книгу под названием «Как Петя Бейсиков Тоню Соображалкину программировать учил» написал заведующий лабораторией ВЦ Латвийского государственного университета им. П. Стучки Б. Я. Мартузан, а издательство «Зинатне» (Рига) выпустило ее в 1987 г. на русском языке в переводе автора. На республиканском конкурсе научно-популярной литературы книга была удостоена диплома первой степени.

Не погибнуть в болоте дилетантизма

82

За последние годы изготовление компьютерных игр превратилось в целую отрасль. Персональные компьютеры выпускаются миллионами, и каждый из них снабжается не одним десятком игр. Играют все: от маленьких детей до пенсионеров, от академиков до домашних хозяек. Незаметно для себя играющие входят в мир будущего — мир компьютеров, с увлечением осваивая то, чему раньше приходилось учиться годами. Когда просто играть с компьютером надоедает, можно подняться на следующую ступень, запрограммировав новую игру. Используя язык Бейсик, имеющийся практически на всех персональных компьютерах, обучиться программированию можно за несколько дней. Профессия программиста — ранее удел немногих избранных — становится доступной каждому.

Такие или почти такие высказывания часто приходится слышать. Источником их, наряду с обывательскими восторгами (ах, какая умная машина!), является, по-видимому, профессиональный эгоизм программистов. Подобно европейцам, обнаружившим, что спиртное пользуется большим спросом у туземцев, и ради своей прибыли спившим целые народы, программисты и изготовители ЭВМ обнаружили, что рынок сбыта их изделий — гораздо более массовый, чем существовавшие ранее, и крепко за него держатся. Это обстоятельство является, вероятно, одной из движущих сил

мутной волны нынешнего компьютерного бума.

Более трезвый взгляд на вещи показывает, однако, что есть основания скорее для тревоги. Чтобы понять, что именно вызывает опасения, сравним компьютерные игры с традиционными.

Большинство обычных игр, как правило, рассчитаны на нескольких играющих. Элемент человеческого общения, возможно, и составляет их главную ценность. Компьютерные игры, как правило, игры для одного. Они подменяют общение суррогатом, создавая иллюзию жизни в сказочном мире, единственным обитателем (а то и властителем) которого является играющий.

Играющий в футбол или теннис дышит свежим воздухом и развивается физически. Человек, проводящий дни и годы за компьютерной игрой, может разве что научиться быстро нажимать какие-то кнопки да испортить зрение.

Часто игры (особенно детские) можно рассматривать как своеобразную подготовку к будущей деятельности. Достаточно понаблюдать немного за типичной компьютерной игрой, чтобы понять, что если она к чему-то и готовит, то лучше бы этого не делала.

Почему же компьютерные игры столь увлекательны? Говоря о причинах, обуславливающих появление программистов-маньяков, Дж. Вейценбаум отмечает, что их привлекает в программировании возможность создать в компьютере воображаемый мир, полностью

им подвластный. В играх зачастую достигается та же возможность, но гораздо меньшими усилиями.

Другая причина привлекательности игр — спекуляция на исследовательском инстинкте, побуждающем человека постигать законы, по которым живет заэкраный мир. Эта потребность, изначально направленная на познание природы, переключается в ряде случаев на исследование плодов большой фантазии автора игры.

Часто, оправдывая игры, говорят о том, что они являются ступенью к более осмысленной деятельности — программированию и что в наши дни, когда программирование становится массовой профессией, игры могут стать одним из путей к ней. Действительно, игры — первая ступень к программированию, но какому? Об этом можно судить, например, по журналу «BYTE», рассчитанному в основном на пользователей персональных компьютеров. Почетное место в нем занимают игры, но немало и статей по программированию. Просмотрев некоторые из них, мы увидим, что речь идет не о науке или искусстве программирования, а об искусстве убивать время с помощью программирования, при котором компьютер остается не более чем игрушкой. Хорошо гармонирует с этим и язык Бейсик. Действительно, при наличии доступного компьютера ему может выучиться каждый. Но это не только не приближает его к про-

фессиональному программированию, а и удаляет от него!

Многие люди, намеревавшиеся использовать компьютер для решения возникших у них задач, и не заметили, как произошло перерождение средства в цель: не компьютер стал средством решения задачи, а задача — поводом для игр с компьютером. В результате многие люди, достойные лучшей участи, погибли в болоте дилетантского программирования. Не вдаваясь в обсуждение причин этого, подчеркнем еще раз наш главный тезис: программированием должны заниматься только квалифицированные специалисты, а не миллионы дилетантов.

В каком же смысле нужно тогда понимать тезис о программировании как второй грамотности? Совсем не в том, что все должны уметь кое-как программировать и быть готовы использовать это умение в работе. Мы не считаем человека грамотным только потому, что он способен нацарапать слово на заборе. Точно так же умеющий кое-как слепить полурботающую программу на Бейсике отнюдь не является программистски грамотным. Грамотность состоит в другом — в знакомстве с фундаментальными сущностями, лежащими в основе программирования и достойными включения в общеобразовательный минимум культурного человека. К понимаемой в этом смысле компьютерной грамотности игры не имеют никакого отношения и могут сыграть лишь роль интеллектуальной соски-пустьшики.

Еще один миф, который приводится в оправдание игр, — они якобы позволяют преодолеть психологический барьер «страха перед машиной», что необхо-



димо будущим пользователям компьютеров. На самом деле этот барьер представляет собой здоровую реакцию на те отвратительные системы, которые зачастую подсовывают пользователям. Нужно не промывать мозги клиентам, сбывая негодный товар, а производить хорошие системы. (Так мы возвращаемся к вопросу о квалификации программистов!) Действительные проблемы в этой области связаны не с каким-то «психологическим барьером», а с тем, что новые технологии, использующие компьютеры, требуют и новых профессиональных навыков. Обучение этим навыкам — серьезная задача, и компьютерные игры тут ни при чем. (Пример: применение ЭВМ в машиностроительном проектировании требует, по-видимому, перехода от языка чертежей к другим

языкам описания трехмерных объектов. Это, разумеется, не имеет ничего общего с мнимым страхом перед машиной.)

Подобно вирусу, проникающему в клетку, заставляющему ее производить его копии (вместо занятия тем, чему клетка была изначально предназначена) и приводящему ее к гибели, компьютерные игры проникают в мир вычислительной техники, а замаскировавшись под персональные компьютеры — и вообще в мир науки. Небывалые темпы развития отрасли производства компьютерных игр несомненно являются признаком ее злокачественного перерождения. Будем же рассматривать производство игр наравне с производством наркотических веществ и распространим законы о спивании несовершеннолетних и на эту область!

83

Конкурс программ завершен

Не сегодня, а может, и не завтра оживут на экране дисплея картинки из школьных учебников: побегут, стелкаясь, натываясь друг на друга, электрончики, работает человеческое сердце, возникнут строгие линии гео-

метрических фигур... Не сегодня. А пока в Москве прошел второй городской конкурс на лучшую работу молодых ученых и специалистов по обучающим и игровым программам.

О нем рассказывает от-



ответственный секретарь организационного комитета Н. А. Родионов.

84

— Конкурс был организован МГК ВЛКСМ и Московским городским советом научно-технических обществ и проводился с мая по ноябрь 1987 год. Материалы стекались в горком, в отдел научно-технического творчества молодежи и оценивались экспертами. Часть программ рассматривалась с выездом на места.

— Чем отличается этот конкурс от предыдущего?

— В этот раз все работы предназначались для персональных ЭВМ и были представлены на магнитных носителях. В прошлом году некоторые разработки были выполнены для ЕС ЭВМ. Рассматривать их было сложно. По качеству программы стали несколько слабее, хотя количество их увеличилось (35 против 24).

— Расскажите подробнее о конкурсных работах.

— Сначала о программе «Эрудит II», занявшей первое место. Ее создатели — авторский коллектив факультета ВМК МГУ под руководством аспирантки И. Горячей. Эта разработка участвует в конкурсе второй раз (год назад она заняла второе ме-

сто). По нашему положению, запрещается повторное представление программ. «Эрудит-II», однако, была существенно переработана по сравнению с прошлым годом. Ее уровень на порядок выше остальных работ. Она — простое и удобное инструментальное программное средство (для многих неспециалистов в области программирования). Предоставляет возможности для ввода и корректировки учебной информации в ЭВМ и последующего предъявления этой информации обучающемуся в различных игровых сюжетах.

Радуется, что много работ поступило от школьников. Хотелось бы отметить учебно-производственный комбинат № 1 Севастопольского района Москвы, где под руководством хорошего педагога и при содействии шефов ребята делают интересные вещи. Школьники представили 11 работ. Среди них программа «Профориентирование». Чисто программистски она не сложна (серия вопросов — ответов, а затем рекомендация).

— Вы рассказали об обучающих программах, были ли на конкурсе игровые?

— Игровых работ могло быть больше. Все упирается в технические возможности.

Чтобы игра была интересной, желательно иметь цветную графику...

— Будет ли проводиться конкурс в дальнейшем? Расширится ли состав участников?

— В будущем году вновь планируется проведение конкурса. В этот раз мы недобрали работ, из-за того что, получив постановление, условия конкурса, райкомы комсомола не проявили достаточной активности.

Мы будем совершенствовать положение о конкурсе. Сейчас оценка игровых и учебных программ не дифференцируется. Имеет смысл в будущем оценивать их раздельно. Они очень разные по задачам. Другое дело — обучающие программы, построенные в игровой форме...

Хотелось бы видеть на конкурсе работы профессиональных, мобильных коллективов.

**В. БУЛДАКОВ,
А. СУББОТИН,**
студенты ф-та журналистики МГУ

От редакции. Заинтересованные лица могут получить более подробную информацию по тел. 939-18-80 от Илоны Владимировны Горячей.

Самое прямое отношение

Будущее — за компьютерами на сверхпроводниках. Теперь, с открытием сверхпроводимости при «азотных» температурах, это стало окончательно ясно...

Японские школьники на уроках физики могут проводить эксперименты со сверхпроводящей керамикой. Фирма, занимающаяся ее выпуском, придумала применение остаткам от производства товарных изделий —

изготавливает из них наборы для школьных лабораторий. Эти наглядные пособия довольно дороги, но пользуются таким спросом, что фирма не успевает выполнять поступающие заказы.

А в Англии школьники проявили инициативу и изготовили сверхпроводник самостоятельно. Учительница химии помогла добыть нужные ингредиенты, их перемололи, перемешали, спекли

в муфельной печи — и убедились, что последнее достижение науки может быть повторено, как говорится, «на колене».

На какое отношение все это имеет к школьной информатике? Самое прямое. Эти два случая — яркий пример того, что школа не только должна, но и может идти в ногу с веком. Грядет время компьютеров на сверхпроводниках...

Публикуя слайды студента МГУ В. Булдакова, «Молодежная инициатива» приглашает ребят, увлекающихся фотоделом, прислать нам свои работы. Обязательные условия: «компьютерная» тема, оригинальность сюжета. Размер черно-белой фотографии не менее 13×18 на глянцевой



бумаге, слайды желательно 6×6. Необходимо указать, где и когда проводилась съемка, кто изображен на снимке. Сообщите также свои имя, отчество и фамилию, возраст, а также почтовый адрес. Лучшие снимки будут опубликованы.

МЫСЛИ И ФРАЗЫ...

● Если вы не умеете программировать — не отчаивайтесь, скорее всего, это не самый худший из ваших недостатков.

● ТЗ для программиста: пойти туда, не знаю куда, найти то, не знаю что.

● Программист не стареет, — устаревают его программы...

● Если твои близкие не программируют, стыдись! Ты позоришь звание программиста!

● Диагностика: лучший способ еще раз убедиться, что во всем виновата машина...

● Программисты, как и ЭВМ, бывают 1, 2, 3, 4-го и т. д. поколений...

● Программист пишет программу для ЭВМ, а его начальник — для программиста...

● Если с твоей программой может свободно работать совершенно посторонний человек, то либо ты альтруист, либо тебе есть еще чему поучиться в жизни...

● В чем разница между системным программистом и программистом вообще? Первый, как правило, работает на машину, а второй заставляет машину работать на себя.

● С юмором написанная программа зачастую лишь подтверждает отсутствие чувства юмора у заказчика.

● Каждая новая программа — это, как правило, хорошо забытая старая.

СОВЕТЫ НАЧИНАЮЩИМ...

● Не давай далеко заплывать плавающей запятой!

● Помни: войти в подпрограмму, как правило, гораздо легче, чем из нее выйти!

● Смело плети компьютерную сеть: в конце концов кто-нибудь из заказчиков в нее да попадется!

● Бойся начальника, ТЗ приносящего!

● Учти: вложенные циклы не должны скрипеть при работе...

● На каждые тридцать правильных операторов записывай один неправильный — это делает жизнь разнообразнее!

● В день получения программируй, не отходя от кассы!

Ю. ШЕХОНИН





И. АНДРОПОВ, Е. ГУСЕВ, М. ЗВЯГИН

Для кого написано методическое письмо?

87

Если я прикажу какому-нибудь генералу порхать бабочкой с цветка на цветок... и генерал не выполнит приказ, кто будет в этом виноват — он или я?

*А. де Сент Экзюпери.
«Маленький принц»*

«В настоящее время в Москве в целом сложилась возможность преподавания курса ОИВТ как в IX, так и в X классах по машинному варианту». Эту победную реляцию можно прочесть в методическом письме «О преподавании курса ОИВТ в IX и X классах школ Москвы в 1987/88 учебном году», подготовленном кабинетом информатики и вычислительной техники МГИУУ.

Мы не будем останавливаться на наиболее часто обсуждаемых проблемах, порожденных бесконечным разнообразием и крайне низкой надежностью большей части имеющейся вычислительной техники, почти полным отсутствием необходимого программного обеспечения, недостатком квалифицированных кадров и т. д. Скажем несколько слов об упомянутом письме.

Прежде всего, этот документ должен представлять собой тезисы курса, где несколькими предложениями намечается урок, методика, подход. В документах такого рода каждое слово должно быть выверено и максимально информативно. Поэтому особенно грустно встречать здесь неадекватные выражения вроде «тип информации» (дважды!) или «тип алгоритма».

В разделе «Основные требования к уроку информатики» подчеркивается, что «в процессе изучения теоретической части курса необходимо в максимально возможной степени использовать учебные пособия «Основы информатики и вычислительной тех-

ники» части 1 и 2». Там же настоятельно рекомендуется изучение учебного алгоритмического языка, причем «...алгоритмы сначала записываются на учебном алгоритмическом языке или в виде блок-схем...». Подобное заявление содержит в себе, на наш взгляд, глубочайшее противоречие, ибо приводит к одному знаменателю два принципиально различных подхода к обучению, что дезориентирует не слишком искушенного в программировании учителя. Все это ставит под вопрос само название предложенного письма как методического.

Методическое письмо должно дать учителю главное — «модель» урока, структура которого, особенно с учетом использования ВТ и сомнительных санитарных норм, весьма специфична. Как показали многочисленные беседы с учителями, большую трудность вызывает именно самостоятельное построение урока. И еще одна сложность, так сказать, правового характера, — а что записывать в классный журнал?

Что касается поурочного планирования, то, наверное, естественно ожидать от методического документа такого уровня соблюдения одного из основных дидактических принципов — от простого к сложному. Однако...

В теме «Основы алгоритмизации», как правило, речь идет о графике. Стоит ли так драматизировать эту возможность ЭВМ, тем более что многие ее не имеют? Но даже если она есть, то, следуя предложенному планированию, надо посвятить уроки 5—10 IX класса использованию преимущественно графического редактора. Бесспорно, он зажигает детские глаза, ученики готовы рисовать картинки на экране. Однако уже через 4 (а у большинства

школьников — через 2) урока интерес к предмету вообще, а не только к этому занятию, резко падает, что, по нашему мнению, объясняется непониманием смысла выполняемой работы. Необходимую же мотивировку почему-то предполагается дать значительно позднее.

При этом в течение 11—14-го уроков предлагается ввести сразу: «Язык программирования. Алфавит, величины, операторы, функции, программный и непосредственный режим работы на ЭВМ. Графические возможности языка программирования». Отработку всего этого колоссального объема теоретического материала предлагается проводить с использованием «...алгоритмов построения простых изображений на экране дисплея...». Таким образом, даже здесь, в наименее спорном разделе курса — «Язык программирования» — мы опять занимаемся рисованием.

88 Почему-то лишь в 15—16-м уроках вводятся понятия переменной и присваивания. Не поздно ли после двух месяцев алгоритмизации?

И только к середине X класса, после полутора лет изучения программирования и информатики, предлагается начать изучение массивов (да и то одномерных!); понятие же многомерных массивов отсутствует вовсе!

Относительно темы 5 — «Принципы устройства и работы ЭВМ» — хотелось бы отметить, что последовательность изложения представляется нам хаотичной. Здесь на учащихся обрушивается лавина трудных для усвоения разнородных понятий: процессоры, файлы, операционные системы, двоичное кодирование, опять процессоры, АЛУ и оперативная память, счетчик команд, магистраль, триггеры и внешние устройства ЭВМ. Все эти сведения сообщаются ученику после полутора лет работы на ЭВМ, выполнения долговременных заданий, регулярного использования готовых программных продуктов.

Кстати о заданиях. Учителю предлагается

использовать вычислительные задачи из математики и физики. Хотя эти задачи менее всего демонстрируют возможности компьютера.

Тема 6 («Роль ЭВМ в современном обществе») — нужнейшая, это мотивировка всего курса. Вообще-то методическое письмо предлагает планирование, несколько отличное от утвержденной программы, однако в данном случае следует ей (не вполне разумно) и помещает эту тему в самом конце (кстати, еще более ошибочно следование программе в содержании и размещении темы «Решение задач с использованием ЭВМ»). На нее отводится 23 ч, несмотря на то, что освещаются лишь традиционные виды программного обеспечения. Не только преподаватель школы, но и высококвалифицированный специалист едва ли сможет интересно говорить по этому поводу (без технических подробностей) в течение 23 ч, тем более не имея возможности демонстрации соответствующего обеспечения.

Итак, попробуем подвести итоги: для кого и зачем написано это методическое письмо, столь напоминающее многочисленные инструкции, которые никто не выполняет (чего и не требуется)? Если для учителя грамотного, владеющего предметом, то, разумеется, отмеченные противоречия он найдет возможность устранить самостоятельно. А может быть (чаще всего так и бывает), предложит учащимся свой вариант изучения курса, основываясь на программе машинного варианта, опубликованной в печати. Если же для учителя, слабо владеющего предметом, то вряд ли удастся таким образом организовать обучение.

По-видимому, чтобы создать действительно полезный для учителя вариант планирования курса ОИВТ, следовало бы решать этот вопрос на демократических основах, а именно — привлечь к разработке и обсуждению столь важного и нужного документа учителей Москвы, и в первую очередь тех, кому удастся успешно преподавать этот предмет.

«Как бы есть...»

(обобщение двухлетнего опыта преподавания информатики на базе ПЭВМ «Агат»)

1. Для разумного преподавания курса информатики (как и для любого другого курса) необходимо иметь научно обоснованную, практически проверенную программу. На данный момент не существует не только *научной*, но и просто *нормальной* программы (нормальной назовем программу, согласующуюся со здравым смыслом).

2. «Железо». Есть некоторая разница между плохо работающей ЭВМ, хорошо работающей ЭВМ и совсем не работающей ЭВМ. Однако пока этой разницы «как бы нет», что приводит к печальным результатам.

3. «Железо» все-таки не главное. Утверждение, что непосредственно работает на пользователя не ЭВМ, а программа, выглядит банальностью. К сожалению, до производителей ПЭВМ это еще не дошло, а в результате — катастрофические последствия в первую очередь для школы.

Непосредственным толчком для создания этих заметок послужило появление методического письма «О преподавании курса «Основы информатики и вычислительной техники» в IX и X классах Москвы в 1987/88 учебном году». Оно, как я понял, является одним из последних вариантов программы по информатике, так что прочтем его внимательно. Итак, первая трудность.

Программа. IX класс. На изучение графических возможностей языка отводится 6 ч, на изучение «всего остального» отводится 10 ч. При этом изучение графических возможностей начинается до изучения основных структур языка. Не слишком ли много времени отдано освещению частной возможности языка; не слишком ли мало — самому языку? Правильно ли изучать сначала частную возможность вывода результатов программы (причем так долго), а лишь затем объяснять, из чего может состоять сама программа?

После 10 ч изучения языка на протяжении 8 ч предлагается решать «Задачи из курса математики и физики». Очень хорошо. Но какие конкретно задачи имеются в виду? После 5 ч программирования (так как всего на Бейсик отведено 10 ч, из них половина — теория (как диктуют медицинские нормы)) ничего сложнее навязшего

на зубах квадратного уравнения средний ученик средней школы решить не сможет. Или, по мнению составителей письма, 1 ч (а именно столько выделено ими) достаточно для того, чтобы ученик смог построить математическую модель для задачи «ненулевой» сложности? И разве построению математической модели следует учить лишь в конце года?

Одним словом, я просто не представляю себе, что это за задачи *из школьного курса*, которые можно решать после 5 ч практики программирования (если, конечно, здесь не подразумевается, например, рисование на экране квадрата).

X класс. С «Алгоритмизацией и программированием» та же история: 8 ч на работу с графикой, 6 ч на символичные переменные (т. е. на частные возможности языка) и в полтора раза меньше (9 ч) на само программирование. Причем что страдает: массивы и подпрограммы! Опять смещение акцента, по меньшей мере неоправданное.

Часы 27—28 и 29—30: составление учащимися по индивидуальным заданиям программ, имитирующих объекты и процессы, контролирующих программ и программ-тренажеров. Я проработал на ЭВМ 7 лет и не могу назвать эти планы иначе, чем блефом.

За 1 ч работы на машине (два урока — 2 ч, по медицинским нормам пополам — 1 ч) никакой ученик не сможет отладить программу, имитирующую объект или процесс и, тем более, программу-тренажер.

«Принципы устройства и работы ЭВМ» — это 13 ч, отведенных на сложнейшую техническую и идейную информацию.

Архитектура ЭВМ, программное обеспечение ЭВМ (операционная система), двоичное кодирование, процессор, физические принципы работы... Опять-таки некоторая бессистемность. Операционная система — это ведь не вещь в себе! По программе получается: сначала мы работаем с операционной системой, затем расскажем, как работает и для чего нужна некая абстрактная операционная система, и лишь затем выясним, что это такое. На практике проверено — это ведет к полной путанице. Ученик должен хотя бы приблизительно

понимать, что происходит в машине в то время, когда он на ней работает (например, что при включении происходит загрузка операционной системы, курсор на экран выдает операционная система, у машины есть память, куда помещается программа, клавиатура не «печатает буквы на экране», а общается с памятью и т. д.). Мало того: 11—14-й уроки IX класса говорят о программном и командном режимах работы; и при этом ни слова об операционной системе, ни слова о памяти?!

Одним словом, эту тему нужно раскрывать не так и не тогда.

«Роль ЭВМ в современном обществе». 23 ч рассказа об ЭВМ, сопровождаемые общением с богатым программным обеспечением. Увы, на данный момент это полная утопия. Аргументация будет приведена ниже.

90 Что же в итоге? Несбыточная мечта: иметь программу, а) четко определяющую, чему нужно научить каждого школьника, а чему энтузиаста (не всем, на мой взгляд, нужно знать физические основы ЭВМ на уровне конструкции сумматора); б) стройную и логичную; в) учитывающую реальную математическую подготовку современного ученика.

Ни одним из этих качеств программа, изложенная в методическом письме, не обладает.

Борьба с железом. Состояние электронной техники весьма прискорбно, и это, в частности, делает неприемлемыми некоторые положения методического письма.

Мне довелось работать с двумя комплектами ПЭВМ «Агат», об «Агате» я и буду писать. Впрочем, многое будет верно и в отношении КУВТ-86 — если верить тем его пользователям, с которыми мне приходилось общаться.

Итак, 33—34-й уроки X класса. По методическому письму предполагается активная работа с локальной сетью.

Нет сейчас на «Агате» локальной сети, а когда будет — нельзя будет с ней работать! Мучаются люди с сетью на КУВТ-86. Сбоит она там непрерывно. Объединенные же в сеть «Агаты» представляются мне сплошным адом: сам по себе каждый «Агат» раз в час вылетает из операционной системы; что же будет, когда они будут влиять друг на друга?!

Активная работа с программным обеспечением предполагается при изучении темы «Роль ЭВМ в современном обществе». По сети программы пересылать нельзя (см. выше). Последовательно загружать с одной дискеты все машины? Дискета прослужит не больше недели (спасибо, если больше

одного урока) — проверено двумя годами работы с болгарскими дисководами. И как работать с пакетами? Ну а для работы с «Агат-автором» (и не только с ним) дискета нужна непрерывно.

Можно дать на каждую машину по комплекту дискет — и тогда «запилим» их все очень быстро. Да, собственно, где же их взять? На машину дается 10 штук, а накопленные программы занимают уже все 20 (а резерв? и т. д.).

Итак, работа с программным обеспечением (а это 25—29 ч по программе) сильно затруднена низким качеством дисководов и никакими улучшениями (связанных с сетью) не будет!

Выхода нет никакого: ремонтники не считают себя обязанными приводить машины (например, дисководы) в нормальное рабочее состояние, у них и на приведение машин в «хоть как-нибудь рабочее» состояние сил не хватает. Вот и считается ПЭВМ работающей, если проходят тест памяти и тест комплексный, хотя всем прекрасно известно, что это не только успешной, но даже нормальной работы не гарантирует. Набрать 5-строчную программку и запустить ее с грехом пополам на «работающей» машине можно, но записать на дискету — уже далеко не всегда. А прочитать дискету на другой машине — фокус, который удается редко. Плюс очень неудобные клавиатуры, плюс вредные для глаз телевизоры, плюс ужасный, невыключаемый динамик. Стрессовые ситуации при работе на «Агате» возникают каждую минуту! 40—50 % времени учителя тратится на изнуряющую борьбу с халтурой. Какая локальная сеть?! Какая база данных?! Добиться бы, чтобы клавиши не западали, чтобы магнитофонный вывод работал не у двух, а у всех 13 машин, чтобы дисководы не портили дискет, телевизоры не горели! (У меня горели.)

Еще одна несбыточная мечта: работать на безотказной технике (по определению американцев, «персональная ЭВМ — надежность военной техники, стоимость бытовой электроники». «Агат» стоит 4000 рублей, но более ненадежного устройства я в жизни не видел).

И все же главное — не железо. Все вышеперечисленные проблемы связаны с недостатком опыта преподавания и низким уровнем развития технологии. В той или иной степени они со временем сгладятся. Проблема же программного обеспечения, которая, насколько я могу судить, существует не только в школе, — проблема принципиальная. На сегодняшний день труд программиста не считается трудом. Нет четкого механизма «написал программу — про-

дал», нет заинтересованности в распространении и качестве программы, нет хорошей документации. Создание программного обеспечения стало делом бескорыстного энтузиазма (что приводит, в частности, к незавершенности программ, их некультурности, неудобству в эксплуатации, ненужному дублированию). Распространение программного обеспечения — это обмен, выпрашивание и даже воровство. В итоге — полная неразбериха, хаос. На «Агате» из всего «обеспечения» достойна так называться лишь десятая часть программ. (Программное обеспечение «как бы есть», но его на самом деле нет, так же как ПЭВМ, прошедшая тест памяти, как бы работает, а на самом деле «пилит» дискеты.)

А теперь конкретно о программном обеспечении «Агата». Система «Школьница», возможно, хороша, но я на ней не работаю, так как: а) для учебного процесса она неудобна; б) определить ее сбой во многих случаях невозможно; в) программы, написанные в одной ее версии, могут не идти в другой, а описание есть только на первую версию; г) дискета должна всегда быть в дисковде (см. выше). Наконец, самое главное — подавляющее большинство пользователей работает на Бейсике, и основное обеспечение написано на нем, а давать и Бейсик, и Рапиру одновременно невозможно, и так времени мало.

Поэтому для меня недоступны: единственная существующая (хоть и плохонькая) база данных (она написана в «Школьнице»); программы-исполнители (по той же причине) и т. д. Некоторое число уроков оказывается без поддержки. Так что «база данных как бы есть».

Система «Бейсик-Агат». В настоящее время существует, как минимум, пять версий. Ни на одной из них спокойно работать нельзя. Наиболее надежная («штатный ДОС») работает очень медленно, создает трудности при редактировании, лишена мно-

гих нужных возможностей, реализованных в других версиях.

Все версии страдают несоответствиями описаниям, данным в документации к ПЭВМ, у многих есть индивидуальные недостатки: в одной потеряна вырезка из символьной переменной, в другой потерян магнитофонный вывод и т. д. Плюс ко всему некоторые программы (включая системные), написанные в одной версии, не идут в другой.

Итак, интерпретатор Бейсика на «Агате» «как бы есть», но нормально на нем не поработаешь.

Ну и так далее: клавиатурный тренажер «как бы есть»; графический редактор «как бы есть» (увы, он не является редактором-исполнителем, который требуется в соответствии с методическим письмом); текстовый редактор «как бы есть» (но освоить «Агат-автор» за 2 ч, как того требует методическое письмо, невозможно); динамические таблицы «как бы есть» (за 2 ч их не освоить); пакеты прикладных программ «как бы есть» (частично на «Школьнице», а то, что есть на Бейсике, не отвечает минимальным требованиям культуры программирования).

Я прошу учесть, что часть сведений этого раздела может быть ошибочной. Может быть, где-то уже давно существует какая-нибудь бейсиковая база данных, описание отличий «Школьницы» 3.1 от «Школьницы» 2.1, безошибочно работающий компилятор Бейсика и т. д. Но я, пользователь, о них ничего не знаю, а если и узнаю, то не достану, а если и достану, то только «левым» путем. Так что даже если это все и есть, оно «как бы есть».

Последняя несбыточная мечта: иметь возможность законным путем приобретать необходимое программное обеспечение (безотказное, культурное, сопровождаемое удобочитаемой документацией).

Несбыточная мечта в квадрате: возможность делать заказы на обеспечение.

Всё.

ВНИМАНИЮ ЧИТАТЕЛЕЙ!

Оргкомитет Всесоюзного научно-практического совещания «Электронно-вычислительная техника в общеобразовательной школе» приглашает всех заинтересованных лиц принять участие в работе совещания, которое состоится 19—21 октября 1988 г. в г. Новосибирске в НИИ ИВТ АПН СССР. Тезисы докладов и выступлений объемом 2—3 с. должны быть представлены в 2 экз. до 1 июня 1988 г. по адресу: 630098, Новосибирск, ул. Приморская, 22, НИИ ИВТ АПН СССР, Оргкомитет. Справки по тел.: 45-01-08, 45-02-67.

О программе курса ОИВТ: глядя из вуза

В последние два-три года активно обсуждаются причины резкого увеличения числа абитуриентов в гуманитарных вузах. Выдвигается несколько причин для объяснения этого явления, но, по моему мнению, главная — в том, что именно в эти годы мы пожинаем плоды «усиленного» математического просвещения.

92 Во время первой реформы математического образования было произведено «упрощение» школьной математики, и сейчас в вузы приходят абитуриенты, прошедшие этот курс, основной недостаток которого — нарушение главного принципа преподавания математики не только в школе, но и на младших курсах вузов: от простого к сложному, от частного к общему, от конкретного к абстрактному. Отказ от индуктивного метода преподавания математики привел к тому, что большинство школьников (7:1) приобретает стойкое отвращение к математике. А имеющие свободу выбора абитуриенты соглашаются на любой вуз «антиматематического» направления.

Не знаю, насколько лучше обстоит дело с преподаванием математики в школе сейчас; пока что заметного изменения отношения школьников к математике не видно.

Все это сказано для того, чтобы подчеркнуть подобную опасность, грозящую школьникам, начинающим изучать «Основы информатики и вычислительной техники» по пробному учебному пособию (далее — пособие).

Принято считать, что одна из основных задач нового курса — снять так называемый психологический барьер между школьником и ЭВМ. Казалось бы, логика движения от простого к сложному требует, чтобы вначале была показана простота (элементарность) вычислительных функций ЭВМ; в пособии же принята обратная логика. Уже во введении психологический барьер не только не ослабляется, а, наоборот, возводится до космических масштабов: нагромождается масса непонятных для непосвященного вещей (например, моделирование плазмы в ядерных реакторах, построение компьютерной томограммы и т. п.). И если наивный школьник интуитивно предполагал, что ЭВМ — сложная вещь, но питал некоторые надежды, что он в ней разберется, то после такой «арт-подготовки» у него от этих надежд ничего не останется.

Далее, вместо того чтобы с помощью простейших примеров (типа $1+1=10$) снять ореол непознаваемости с двоичной системы счисления и тем самым показать, что в основе работы ЭВМ лежат простейшие принципы, пособие переходит к изложению алгоритмов, которое, естественно, начинается с определения понятия алгоритма. В последнем совершенно верно утверждается, что одним из важнейших свойств алгоритма является его понятность исполнителю. И если далее идет изложение правил составления алгоритмов именно для ЭВМ, то, двигаясь от простого к сложному, предварительно нужно было бы описать возможности исполнителя (ЭВМ), т. е. изложить, что именно ЭВМ умеет делать и что она понимает. Без такой предварительной подготовки изложение алгоритмического языка становится формальным (и это на целый учебный год). Поэтому на вопросы школьников, почему данную конструкцию алгоритма нужно писать так, а не иначе, приходится отвечать почти по Чехову: так надо потому, что так надо всегда.

Формальный подход к изложению алгоритмического языка приводит к тому, что такие очень важные и трудные для восприятия понятия, как ввод и вывод информации, остаются вне поля зрения учеников.

Авторы пособия практически отказываются от такого мощного методического приема, как наглядное графическое изображение алгоритма с помощью блок-схем*. Это сделано во имя принципов модного сейчас среди профессионалов структурного программирования. По этой же причине в пособии запрещен оператор перехода. И это в то время, когда буквально все методисты, как у нас, так и за рубежом, считают, что начальное обучение основам программирования должно производиться с помощью блок-схем, в которых стрелки переходов от одной части алгоритма к другой неизбежны.

Более того, даже сами создатели структурного программирования говорят о том, что преимущества последнего проявляются в больших программах — объемом 500 операторов и более, т. е. структурное программирование очень верно названо промышленным программированием. Применение же правил структурного программирования при началь-

* По существу, блок-схема типового алгоритма является его опорным сигналом.

ном ознакомлении с основами алгоритмизации искусственно вуалирует простейшую логику выполнения алгоритма машиной. Короче, излагать идеи структурного программирования без предварительного ознакомления с элементарными операциями машинного алгоритма все равно, что заставлять читать «Войну и мир» без предварительного обучения азбуке.

В вопросе о целях школьного обучения информатике существуют две точки зрения: нужно научить школьника «кнопчному» использованию ЭВМ — электронные игры, применение ЭВМ как обучающей или информационной системы и т. п.;

нужно научить программному использованию ЭВМ — решению нестандартных, своих собственных, задач, т. е., по сути, научить творческому использованию ЭВМ, сделав ее орудием повседневного интеллектуального труда.

Для реализации первого подхода учебный курс по информатике в школе вводить не нужно. Достаточно иметь машину и знать, какую кнопку следует нажать первой. После этого ЭВМ сама сообщит ученику, что нужно делать дальше: то ли играть, если программа игровая, то ли учиться, если программа обучающая, и т. д. Можно с уверенностью сказать, что никакого психологического барьера между ЭВМ и учеником в этом случае не возникает. Подтверждением сказанному служит широкое распространение игровых автоматов, являющихся, по существу, микро-ЭВМ. Родителям приходится не разрушать психологический, а, наоборот, создавать физические барьеры между школьниками и автоматами, чтобы первые учили уроки, а не гоняли электронный зайчик по экрану. Нечто подобное будет и с ЭВМ, если мы сделаем упор на их игровое применение (в Японии с этим уже столкнулись).

Вторая точка зрения на цели обучения школьников информатике конкретизируется в следующей цитате: школа и вуз «должны выпускать специалистов, в совершенстве владеющих методами работы с ЭВМ в соответствии с триадой «модель — алгоритм — программа»... Недостаточно их ознакомить с основами программирования. Более того, вредно сводить проблему использования вычислительной техники к проблеме обучения программированию (как это иногда делается). Нужно сформировать у специалиста математическую манеру мышления, которая позволит ему строить удачные модели изучаемых явлений и исследовать их при помощи вычислительных экспериментов**.

Это — идеальная схема подготовки специалиста любого профиля, использующего вычислительную технику для решения своих задач. Неясно только, как этого добиться, вернее, с чего начать.

Математическая манера мышления формируется в течение всего 10-летнего обучения в школе и 5-летнего — в вузе, для чего, естественно, отводится громадное количество учебных часов. О моделировании же изучаемых явлений на ЭВМ, а тем более об их алгоритмизации можно говорить только после того, как отчетливо уяснишь себе, что именно может машина, как она решает задачу; другими словами, нужно понимать, что такое программа для ЭВМ. Без такого понимания моделирование и алгоритмизация конкретных задач превращаются в схоластические упражнения.

Следовательно, основной задачей школьной информатики должно быть изучение элементарных основ программирования: ввод, вывод и пересылка (передача) информации, выполнение элементарных арифметических и логических операций, порядок выполнения программы (в том числе безусловные и условные переходы). Эти простейшие понятия, доступные ученику V класса, нужно объяснить на самых простых примерах: вычисление процента, расчет среднего арифметического и т. п. Даже алгоритм нахождения наибольшего общего делителя оказывается слишком сложным для начального ознакомления.

Все эти принципы работы любой цифровой ЭВМ прекрасно демонстрируются на программируемых микрокалькуляторах (например, БЗ-34, снабдить которыми школы можно было бы еще в 1985 г.). После такой «машинной» подготовки переход к алгоритмическому языку происходит без всяких затруднений, поскольку он становится осмысленным.

На основе 16-летнего личного опыта преподавания основ программирования на первом курсе экономического*** вуза, а также опыта других преподавателей можно с уверенностью утверждать, что школьный курс информатики обязательно должен включать

системы счисления (краткий обзор на уровне V класса);

программирование простейших задач для микрокалькулятора;

программирование на алгоритмическом языке для микро-ЭВМ. Изучить весь этот ма-

*** Экономические вузы, как известно, относят к гуманитарным, поэтому к нам приходят абитуриенты преимущественно с четверкой по математике.

** Самарский А. А. ЭВМ сегодня и завтра // Наука и жизнь. 1985. № 6.

териал, а также провести практические занятия по нему можно в течение одного года (IX класс), занимаясь по 1 ч в неделю, но лучше за 1/2 года (по 2 ч в неделю). В X классе можно заниматься моделированием, алгоритмизацией, а если будут в школе микро-ЭВМ, то и решением задач по математике, физике и другим предметам.

Если абитуриенты придут в вуз с такой подготовкой, т. е. будут иметь четкое представление о том, что такое программа для ЭВМ, как ее составить, ввести в машину и получить решение, то дальнейшее обучение их использованию ЭВМ для решения задач по избранной специальности (моделирование, применение пакетов прикладных программ и т. п.) значительно упростится.

Что касается предложенной Минпросом

программы конкурсного учебника по информатике, то мечты о том, что указанный в ней материал может быть продуктивно усвоен школьниками в отведенное время, несбыточны (в вузе на эти цели отводится 800 ч). Обилие материала при сокращенном времени неизбежно приведет к поверхностному описательному изложению, к скороговорке, и тем самым для школьников информатика превратится из руководства к действию в скучную описательную науку.

Прежде чем объявить конкурс на новое учебное пособие, нужно провести конкурс на программу этого курса и привлечь к нему в числе прочих и преподавателей вузов, преподающих основы программирования в течение 15—20 лет студентам I курса, которые ровно на 2 мес. старше десятиклассников.

Новое содержание — новая форма

В ходе конкурса на лучший учебник по информатике мне удалось познакомиться с тремя рукописями из девяти. Увы, не одну из них нельзя назвать учебником. И дело тут не в отдельных методических просчетах, которые можно исправить. Эти рукописи при всей их полезности и познавательности никогда не станут настоящим учебником, ибо не несут самого главного — методической идеи. Методики в них нет. Читая рукописи, я, учитель информатики, не мог понять, как же работать по ним, как давать уроки. Если одна из них станет учебником, это принесет тысячам учителей муки, заставляя их выкраивать часы и дни для создания методики нового предмета. Не отрицая полезности этой работы, заметим, что ОИВТ преподают в основном молодые учителя и инженеры, т. е. люди, методически наиболее слабые; тем самым будет обусловлен низкий уровень знаний учащихся. Это может стать стратегической ошибкой на пути внедрения ЭВМ в школу.

Причина создавшегося положения в следующем: ОИВТ — не просто новый предмет в школьном расписании. Он кардинально отличается от всех остальных предметов, ибо предполагает совершенно новые способы познания. Но новое содержание требует и новых форм, а мы пытаемся заковать его в старые рамки. И вопрос об учебнике встает особенно остро. Зачем учебник? Как с ним должен работать ученик и как — учитель?

Раньше учебник был основным носителем информации. Учитель обеспечивал качественное прочтение книги, многократную проработку, заучивание и выполнение упражнений. На это ориентирована и вся методика классического урока: обязательный опрос, объяснение (т. е. прочтение или пересказ) очередной микродозы материала, решение серии упражнений «для всех». Менялись времена, менялось содержание, а формы оставались прежними. Чехарда учебников последних лет наглядно это подтверждает! И вот мы пытаемся перетянуть в совершенно новый предмет весь этот груз. А он не пролезает!

Учебник перестал быть основным источником информации, игнорировать это положение дальше просто невозможно. Основные источники информации — учитель, ЭВМ, книги по предмету, телевидение и т. д. Учебник должен быть «документом сопровождения», ориентиром в безбрежном мире новых знаний, вот тогда он действительно станет помощником и учителю, и ученику. Сразу оговоримся: учебник ни в коем случае не должен предназначаться для самостоятельного изучения предмета — для этого необходимы совершенно другие пособия.

Главным компонентом учебника должен быть не текст — он устаревает каждые 2—3 года, он произносится учителем на уроке, он записан в тысяче других книг — зачем его печатать каждому ученику индивидуаль-

но?! А вот опорный конспект материала напечатать необходимо (лучше всего в образной форме), включив в него тот минимум знаний, которым должен обладать каждый ученик. Разбивка материала на крупные блоки поможет ограничиться 10—12 страницами конспектов.

Второй обязательный компонент учебника — материалы к зачетам. Это крайне важный элемент методики. Именно он поможет обеспечить дифференцированный подход к учащимся. В этот раздел входят список литературы и все вопросы к зачетам, разбитые на три уровня сложности. Первый охватывает материал опорных конспектов, второй — вопросы по обязательной литературе, третий — по дополнительной, для глубоко интересующихся предметом ребят.

Третий компонент — упражнения и примеры. Они тоже разбиты на блоки и уровни. Три уровня обеспечат все категории учащихся. Обязательно наличие в учебнике разнообразных примеров и образцов записи упражнений. Задачи должны быть не только иллюстрацией к теории, но и составляющими системы, обеспечивающей развитие определенного навыка.

Развитием этого раздела служит четвертая часть — примерные темы творческих заданий, рассчитанные на выполнение в течение года, требующие комплекса навыков и умений. Эти задания выполняются индивидуально или бригадами — по желанию и в зависимости от трудоемкости. Учебник должен максимально помочь в этой работе — списком литературы по проблеме, указаниями по планированию последовательности этапов, способам проверки результатов.

Пятый компонент — темы докладов, рефератов и сообщений с примерным перечнем необходимой литературы. Темы разбиты на разделы в соответствии с блоками теории.

Заканчиваться учебник должен кратким словарем терминов и другими справочными материалами.

А текст теории нужно издать отдельной книгой. Она может быть довольно толстой, ведь на школу достаточно 2—3 экземпляров, поэтому тираж небольшой и ограничение на количество печатных листов для авторов можно снять. Такое пособие должно содержать сверхполный материал для теоретических занятий.

Предлагаемая структура учебных пособий уже содержит весь необходимый методический материал и дает методику проведения занятий. Если пойти этим путем, тысячи учителей получат реальную помощь, а не эрзац-учебник с гипотетическим дидактическим приложением. К тому же работа над пособием упростится — легко будет организовать сотрудничество большого коллектива авторов: педагогов, ученых, программистов, художников. Легко будет и совершенствовать учебник, по мере необходимости печатая в журнале «Информатика и образование» новый опорный конспект или систему упражнений.

Сейчас мы закладываем базу для работы через 5 лет. Очень хочется не повторять старых ошибок. А они неминуемы, если мы не пересмотрим назначение учебника, его роль в современной технологии обучения.

95

В. ЛЕОНАС

«Тихое бульканье»

Следствие 17: некомпетентность не знает преград ни во времени, ни в пространстве.

Л. Питер. Почему дела идут кривь и вкось, или Еще раз о принципе Питера // Иностранная литература. 1987. № 2

Недавно мне довелось в качестве рецензента прочесть несколько рукописей учебника по курсу «Основы информатики и вычислительной техники» для IX—X классов средней общеобразовательной школы (чуть было

не написал общей среднеобразовательной, что, к сожалению, ближе к истине), представленных на конкурс, объявленный Минпросом СССР. Ощущения, возникшие при чтении, трудно описуемы; именно они побудили меня написать сию статью (честно признаюсь, что рукопись, победившую на конкурсе, я не видел).

Впрочем, начать, пожалуй, следует с того, что хорошие учебники по информатике существуют, правда, не у нас.

В течение восьми лет в НРБ ведется ши-

рокомасштабный эксперимент, охватывающий около 5 % всех школ страны, проведением которого занимается Проблемная группа по образованию (ПГО). Не вдаваясь в подробности, замечу лишь, что существенную роль в этом эксперименте играют специально разработанные учебники. Помимо того, что они отпечатаны на хорошей бумаге с большим числом цветных (и очень интересных!) иллюстраций, в них собран чрезвычайно интересный материал. Эти книжки приятно взять в руки (и взрослым, и детям), их интересно читать (и взрослым, и детям). Когда в 1986 г. ПГО подарила АН СССР комплект учебников для I—VII классов экспериментальных школ, он на одну ночь попал ко мне домой. Моя дочь (ей тогда было 10 лет) не легла спать, пока не просмотрела все учебники до единого. На следующий день она умоляла либо не отдавать их вообще, либо достать ей такие же (напоминаю, учебники были болгарские и написаны по-болгарски!). Смее уверить читателя, что реакция моей дочери на экспериментальные учебники ПГО не уникальна.

Ну а теперь вернемся к конкурсу. К сожалению, как следует из текста на обороте титульного листа каждой из рукописей конкурсного учебника, «рисунки к данному учебнику воспроизвести не удалось...». Отсутствие рисунков не только ухудшает восприятие рукописи, но и в целом ряде случаев не дает возможности понять, что хотел сказать автор. Так, например, в одной из рукописей сказано следующее: «Отличие последовательности решения задач по схеме искусственного интеллекта от традиционного представлено на рис. 22». Ужасно жаль, что рис. 22 не удалось воспроизвести — было бы очень любопытно познакомиться с тем, как решаются задачи «по схеме искусственного интеллекта».

Общим и весьма серьезным недостатком всех прочитанных мной рукописей является терминологическая каша: «компьютер», «ЭВМ», «машина», «цифровая вычислительная машина» — все эти термины используются равноправно. Вот яркий пример — название одного из параграфов: «Общий вид ЭВМ, основные устройства *компьютера*, их функции и взаимосвязь в процессе работы *машины*» (курсив мой. — В. Л.).

Перечисление подобных перлов заняло бы слишком много места, но еще об одном следует упомянуть: использование в качестве антонима для слова «истина» слова «фальшь». Удивительно, что, используя синонимичные термины, авторы ни одной из рукописей не удосужились сообщить читателям об их синонимичности. Использование синонимов требует явного указания на это. Но, вооб-

ще говоря, использование синонимов в данном случае нецелесообразно. Конечно, детям необходимо сказать о том, что «ЭВМ» и «компьютер» — это одно и то же, но далее в тексте необходимо пользоваться лишь каким-то одним из этих терминов.

Говоря о терминологии, нельзя ограничиться лишь использованием синонимов. Большое значение имеет корректность определенных вводимых терминов с точки зрения современных научных представлений и их понятность детям. К сожалению, как правило, это условие не соблюдается. Вот пара примеров: «Совокупность восьмью триггеров образует регистр байта», «Совокупность декларативных и нормативных данных образуют образ проблемной ситуации — фрейм». Не станем анализировать оборот «образуют образ»; более важно то, что ни до, ни после не объясняется, что собой представляют «декларативные и нормативные данные» (которые почему-то ранее были названы «декларативными знаниями» и «нормативными знаниями») и «проблемная ситуация».

Завершая разговор о терминологии, коснусь проблемы словотворчества. Чего, например, стоит такой термин, как «дбит», обозначающий не что иное как два бита, или термин «подзвучка», поясняемый следующей цитатой: «Нажимать клавиши БК-0010 следует точно и уверенно. Признаком ввода символа в соответствующую позицию строки экрана монитора является слабая подзвучка нажатия клавиши. Звук ввода напоминает тихое бульканье, поэтому следует добиваться этого звука при нажатии почти всех клавиш».

Комментарии излишни.

Читая первую из попавших ко мне рукописей конкурсного учебника (надо сказать, что она оказалась самой ужасной), я неоднократно хотел бросить эту работу, ограничившись написанием отрицательной рецензии по уже прочитанной части. При этом мне в голову приходили мысли типа: «А что, если все остальные рукописи такие же?» Ведь преступно не то что пытаться учить детей по такому учебнику, но даже подавать его на рассмотрение. Я не знаю его автора или авторов (все рукописи попадали к рецензентам под шифром), но сам постеснялся бы поставить свою подпись под таким текстом. Складывается впечатление, что помимо двух областей, в которых большинство людей традиционно считают себя специалистами, независимо от полученного (или не полученного) образования (я имею в виду воспитание и медицину), появилась еще одна. Человек, написавший пять программ на Бейсике, суммарная длина которых не превышает 500 строк, уже начинает считать себя специалистом в области вычислительной техники и

программирования, способным взяться за написание учебника для детей. Как же можно браться за это, не владея терминологией, историей вопроса, наконец, хотя бы элементарной грамотностью в этой области?! И хотя мое утверждение о «пяти программках на Бейсике» не адресовано конкретному автору, должен заметить, что, судя по приводимым в рукописях примерам программ, ни к одной из них не «приложил руку» профессиональный программист.

Быть может, я злоупотребляю цитатами, причина этого — мучительный вопрос: кто же создает рукописи такого рода?! И так, еще несколько перлов:

«...единственная причина трудностей взаимодействия человека с компьютером кроется именно в этой огромной разнице в скорости выполнения операций по преобразованию информации человеком и компьютером».

«Программы компьютеров позволяют материализовать человеческую мысль и увеличить кругооборот знаний».

«Информации без жизни не существует».

«Применение фреймов придает реляционным банкам свойства адаптивности...»

«Системные средства БК-0010 позволяют обращаться к ассемблеру микросхемы К1801, а точнее — программировать в кодах».

«Преобразуются и САПР, сейчас их стали называть супермашинами логического вывода, или интеллектуальными машинами САПР».

«Большая задача образования состоит именно в смещении центра ее (чего? — В. Л.) на решение задач оптимизационного и информационно-поискового характера с целью повышения экономической пользы от ее (чего? — В. Л.) применения».

«Логическое развитие логики».

«БК-0010 имеет возможность запоминать информацию как в виде программы, так и в виде рассчитанных величин на магнитной ленте. Для этого (чего? — В. Л.) необходимо выполнить специальные операторы (какие? — В. Л.) обращения к магнитофону, с помощью которого произойдет запоминание».

«ЕС-1066 имеет большой пульт управления, в котором размещается оперативная память, процессор, устройство сопряжения с каналами связи и другие компоненты структуры».

«...устройство сопряжения с каналом связи с локальной сетью».

«Ячейка памяти ЭВМ, по существу, является обобщенным понятием и может содержать любую структурную единицу информации».

«Представление структурных единиц (сло-

во «информации» пропущено.— В. Л.) в двоичном коде позволяет:

...

— увеличить объем используемой информации путем ее представления на специальных носителях, использующих прогрессивные способы записи и считывания информации с большой скоростью (например, магнитную запись)»;

«...правый и левый регистр...»;

«...мягкие копии (наряду с твердыми копиями.— В. Л.)»;

«...Электросвязь...».

Тяжелое впечатление оставляют разделы, посвященные изучению языков программирования. Спектр предлагаемых языков очень показателен: «мертвый» Алмир, «полуживой» Фокал, «живой» Бейсик. В одной из рукописей соответствующая глава — попытка обучения одновременно (!) двум примитивным и весьма похожему друг на друга языкам — Бейсику и Фокалу. В другой — попытка обучения одновременно (опять одновременно!) Алмиру и Бейсику. Такая одновременность, конечно, ни к чему, кроме каши в головах учеников, не приведет. А по поводу Бейсика можно напомнить мнение Дийкстры об оболваненных этим языком студентах.

Очень большие сомнения вызывает встретившееся утверждение о том, что «трудность изучения алгоритмических языков состоит в том, что большинство из них *составлено* (курсив мой.— В. Л.) на английском языке», и о том, что создание «национальных алгоритмических языков... безусловно облегчит их изучение». Дело в том, что исследования американских авторов, проведенные с американскими школьниками и студентами, изучавшими программирование, показывают, что существуют концептуальные ошибки, не зависящие от используемого языка программирования; их возникновение связано с переносом житейского смысла на ключевые слова языков программирования или с нагружением последних не тем смысловым оттенком.

Итак, пора просуммировать впечатления: УЖАСНО, НИКУДА НЕ ГОДИТСЯ.

Неужели объявление конкурса на учебник каждый раз приводит к появлению макулатуры, которую размножают на ротاپринтере и раздают на рецензию в разные организации? Мне кажется, имеет смысл ввести двухступенчатый конкурс, чтобы на первом этапе отсеивать явную макулатуру. Вот только возникает вопрос: «Что делать, если все рукописи конкурсных учебников отсеются на первом этапе?»

Понимаю, что неконструктивно ругать легко. Более того, чем хуже что-то, тем легче это что-то ругать, но ругать именно некон-

97

структивно. И наоборот, чем лучше что-то, тем легче сделать критику конструктивной. Но как, скажите, конструктивно критиковать мыло, которое не мылится, автомобиль, который не ездит, книгу, которую невозможно читать! К сожалению, в данном случае «мыло не мылилось», именно это и определило характер моей критики. А если говорить о позитивных предложениях, то следует обратиться к опыту ПГО (по подготовке и изданию самых разнообразных учебников), к опыту издательства «Педагогика» (редакция словарей и справочников, редакция научно-популярной литературы), обратиться к ведущим ученым и ведущим популяризаторам науки. Хороший учебник, по-моему, может быть создан либо уникальным человеком (например, крупным ученым-энциклопедистом, но ни в коем случае не узким специалистом), либо коллективом, объединяющим специалистов предметной области, педагогов, психологов (когнитивистов) и, наверное, детских писателей. Все они должны обладать широкой эрудицией и работать не ради денег (иначе это будет халтура или будут давить

сроки), а ради получаемого от этой работы удовлетворения (я не максималист, деньги все равно надо будет платить, но не они должны быть стимулом). И надо очень любить детей, иначе ничего не получится.

А завершить мне хотелось бы (да простят меня в очередной раз) еще одной цитатой, имеющей, по моему мнению, непосредственное отношение к рассматриваемому вопросу.

«— Осетрину прислали второй свежести,— сообщил буфетчик.

— Голубчик, это вздор!

— Чего вздор?

— Вторая свежесть — вот что вздор! Свежесть бывает только одна — первая, она же и последняя. А если осетрина второй свежести, то это означает, что она тухлая!

— Я извиняюсь...— начал было опять буфетчик, не зная, как отделаться от придирающегося к нему артиста.

— Извинить не могу,— твердо сказал тот».

(М. А. Булгаков, «Мастер и Маргарита».)

ЭВМ находят подчас неожиданные применения

Так, например, результаты исследований, проводившихся в Англии в течение семи лет, показали, что использование ЭВМ для контроля выращивания телят обеспечивает значительное снижение себестоимости производства телятины. ЭВМ следит за кормлением, обеспечивая распределение корма в таких количествах и через такие промежутки времени, чтобы имитировать естественные условия выпаживания телят матками. Применение ЭВМ позволяет также снизить падеж и заболеваемость животных.

Другой пример — пакет программ для распознавания грибов, разработанный в лаборатории ботаники и лесоводства университета им. Поля Сабатье (Тулуза). В него входит базовый массив данных, содержащий сведения о 275 видах грибов.

ЧТО МОЖЕТ ЭВМ?

Пользователь должен ответить на вопросы о характерных признаках данного гриба. Распознавание осуществляется после ответов на 5—6 вопросов. Систему предполагается дополнить видеодиском, который позволит воспроизводить изображения грибов на экране дисплея.

Фирма «Эй-ай-эм технолоджи» (Австралия) разработала автоматизированный гладильный шкаф, работой которого управляет микро-ЭВМ. В шкаф помещается до сорока предметов одежды, повешенной на плечики.

Они по одному подаются в специальное отделение, расположенное в задней части шкафа. Датчик определяет состав ткани, а ЭВМ подбирает температурный режим, величину давления и количество пара, необходимого для обработки каждого из предметов одежды.

Настольные издательские комплексы на базе персональных ЭВМ позволяют отказаться от громоздкой и дорогой типографской техники, от услуг наборщиков, верстальщиков, технических редакторов.

Печатающее устройство нового поколения — лазерный принтер — позволяет получать высококачественные иллюстрации (графические и полутонные), в большинстве комплексов предусмотрены наборы шрифтов разнообразных начертаний и размеров. «Типография на столе» способна обрабатывать тексты, менять формат полосы, делать иллюстративные вставки и т. д.

Г. ГУТМАН, О. КАРПИЛОВА
г. Куйбышев

Азбука программирования

Муравей учится считать*. С помощью исполнителя Муравей мы научились составлять линейные программы, познакомились с понятиями «цикл» и «ветвление».

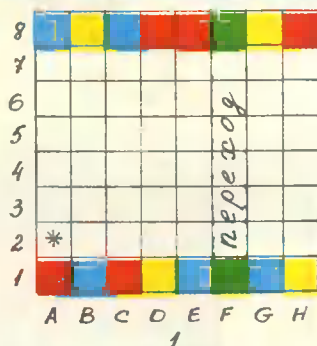
Рассмотрим еще одну конструкцию цикла — цикл «пока».

Цикл «пока». Конструкция этого цикла имеет вид:

пока /условие/ повторять	— заголовок цикла
/серия предписаний/	— тело цикла
кц	— конец цикла

Разберем эту конструкцию на примере следующей задачи (рис. 1).

Муравей должен перейти улицу по переходу. Указателем перехода служит зеленый кубик, расположение которого неизвестно. Остальные кубики (дома, вдоль которых идет Муравей) другого цвета.



Решение. Муравью нужно определить цвет кубика, около которого он стоит, и выполнить ветвление:

* Продолжение. Начало см.: Информатика и образование. 1987. № 6. С. 87—92.

если КУБИК не ЗЕЛЕНый
то ВПРАВО 1
иначе ВВЕРХ 5

все

При этом Муравей либо перейдет улицу, либо сдвинется на одну клетку вправо и должен будет снова повторить те же действия. Так как заранее не известно, сколько раз придется повторять ветвление, то применяется цикл «пока».

начало

пока КУБИК не ЗЕЛЕНый повторять
ВПРАВО 1

кц

ВВЕРХ 5

конец

Исполняя эту программу, Муравей будет двигаться вправо до тех пор, пока не дойдет до зеленого кубика. Как только Муравей обнаружит зеленый кубик, он выйдет из цикла и выполнит следующую за циклом команду ВВЕРХ 5, т. е. перейдет улицу.

Напомним, что граница доски является для Муравья непреодолимым препятствием. Что будет, если, стоя на верхней горизонтали, Муравей получит команду ВВЕРХ 1? Выполнить ее он не сможет, т. е. ему придется «биться головой о стенку». Чтобы избежать этого, научим Муравья проверять условия типа ВВЕРХ МОЖНО; ВВЕРХ НЕЛЬЗЯ, ВПРАВО МОЖНО и т. д.

Теперь по команде
если ВВЕРХ МОЖНО
то ВВЕРХ 1

все

Муравей либо сдвинется на клеточку вверх, либо останется на месте (если он стоит на верхней горизонтали).

Допустим, Муравей находится на произвольном поле. Приведем его на нижнюю го-

ризонгаль. Так как неизвестно, сколько шагов для этого потребуется, написать просто ВНИЗ К мы не можем. Здесь также надо использовать цикл «пока».

начало

пока ВНИЗ МОЖНО повторять
ВНИЗ 1

кц

конец

В результате исполнения этой программы Муравей придет на первую горизонталь.

Приведем теперь Муравья на поле А1 (начальное положение неизвестно). Это легко сделать, дописав предыдущую программу:

начало

пока ВНИЗ МОЖНО повторять
ВНИЗ 1

кц

пока ВЛЕВО МОЖНО повторять
ВЛЕВО 1

кц

конец

100

Предположим теперь, что и на самой доске могут быть непреодолимые препятствия — стенки или столбы, которые Муравей должен обходить. Тогда проверку условий возможности движения надо делать на каждом шаге.

Рассмотрим такую задачу (рис. 2).

Муравей находится в пещере и ищет выход из нее. В стене пещеры (кубиках, стоящих на четвертой горизонтали) один кубик можно сдвинуть и выбраться наружу.

Чтобы найти выход, Муравей движется вдоль стены и проверяет условие ВВЕРХ МОЖНО. Как только это условие выполняется, Муравей по команде ВВЕРХ 2 выталкивает кубик и выходит из пещеры. Получаем программу:

начало

пока ВВЕРХ НЕЛЬЗЯ повторять
ВПРАВО 1

кц

ВВЕРХ 2

конец

В качестве проверяемых условий в цикле «пока» могут быть и составные условия.

Муравей с памятью. Чтобы подготовить школьников к работе с микрокалькулятором, научим Муравья производить простейшие арифметические действия и сравнивать числа между собой, а также снабдим его одной ячейкой памяти (П). В эту ячейку Муравей может записать любое число. Запись каждого нового числа стирает предыдущее. Кроме того, Муравей может прочесть число, которое записано в памяти, и произвести с ним какие-либо действия (например, прибавить к нему единицу или сравнить с заданным числом).

Запись числа в память будем изображать стрелочкой →.

5 → П — записать число в память Муравья.

П + 1 → П — сложить число, находящееся в памяти, с единицей и записать результат в память.

Теперь Муравей может считать.

Пусть он определит номер горизонтали, на которой стоит. Ясно, что для этого достаточно спуститься на первую горизонталь, подсчитывая сделанные шаги. Спускаться вниз с произвольной горизонтали Муравей умеет:

пока ВНИЗ МОЖНО повторять
ВНИЗ 1

кц

Сколько шагов при этом будет сделано? Очевидно, на единицу меньше, чем номер горизонтали, на которой Муравей находится. Поэтому до начала спуска запишем в память Муравья 1 (1 → П) и будем при каждом шаге прибавлять к содержимому памяти 1 (П + 1 → П). Таким образом, программа примет вид:

начало

1 → П

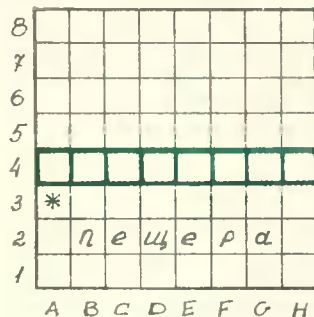
пока ВНИЗ МОЖНО повторять

ВНИЗ 1

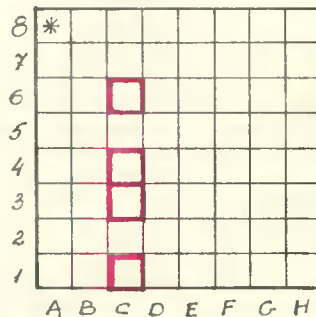
П + 1 → П

кц

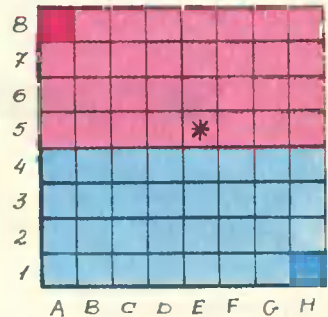
конец



2



3



4

В результате исполнения программы в памяти Муравья будет записан номер горизонтали, на которой он стоял в начальный момент.

Теперь попробуем подсчитать количество кубиков, стоящих на какой-либо вертикали, например на вертикали С (рис. 3).

Поставим Муравья на поле С8. Если по вертикали стоит К кубиков, то, спускаясь вниз, Муравей сможет сделать только 7 К шагов, поэтому для подсчета кубиков достаточно записать в память число 7 и вычитать на каждом шаге 1. Программа принимает вид:

начало
ВПРАВО 2; 7 → П
пока ВНИЗ МОЖНО повторять
П-1 → П

кц
конец
После выполнения этой программы в памяти Муравья будет записано число кубиков, стоящих на вертикали С.

Можно выяснить, сколько среди этих кубиков красных.

Для этого Муравей должен спуститься вниз по соседней вертикали, вычитая из памяти 1 около любого не красного кубика:

начало
ВПРАВО 1
пока ВНИЗ МОЖНО повторять
ВНИЗ 1
если КУБИК не КРАСНЫЙ
то П-1 → П
все

кц
конец
Тогда в результате исполнения программы Муравей спустится на нижнюю горизонталь, а в его памяти будет записано количество красных кубиков.

Дополним список проверяемых условий сравнениями содержимого памяти с заданным числом, т. е. условиями типа: $P=0$; $P \neq 3$; $P \geq 5$ и т. п. Тогда появляются возможности для решения довольно сложных задач. Например.

Привести Муравья на поле А8, если в начальный момент он находится на красной половине доски, и на поле Н1, если на синей половине (рис. 4).

Решение:
начало
1 → П
пока ВНИЗ МОЖНО повторять
ВНИЗ 1
П+1 → П
кц
если $P \leq 4$
то пока ВПРАВО МОЖНО повторять

ВПРАВО 1

кц
иначе ВВЕРХ 7
пока ВЛЕВО МОЖНО повторять
ВЛЕВО 1
кц

все
конец
Дополнительные возможности. Заметим, что при решении задач для исполнителя Муравей кроме освоения основных конструкций — линейных программ, циклов, ветвлений — учащиеся знакомятся попутно и с некоторыми специальными приемами программирования.

Так, подсчитывая число кубиков, стоящих на одной вертикали, мы фактически использовали такой стандартный прием, как организация счетчика. Этот же прием применяется и при подсчете количества кубиков определенного цвета. Легко увидеть сходство представленных задач с задачами из учебника информатики: найти число ненулевых элементов в таблице чисел или найти количество элементов таблицы, удовлетворяющих какому-либо условию.

Задача отыскания номера первого (или последнего) элемента таблицы, удовлетворяющего некоторому условию, формулируется для Муравья следующим образом:

Найти по вертикали А самый нижний красный кубик и определить, на какой горизонтали он стоит (рис. 5).

Решение:
начало
1 → П
пока КУБИК не КРАСНЫЙ и ВВЕРХ
МОЖНО повторять
ВВЕРХ 1
П+1 → П
кц

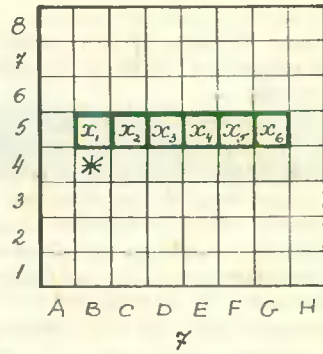
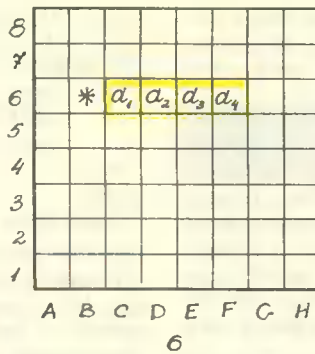
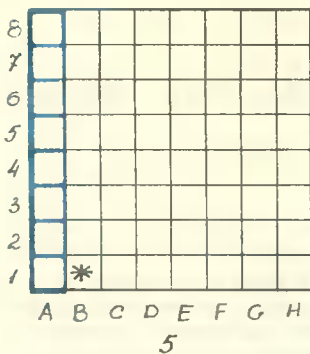
если КУБИК не КРАСНЫЙ
то 0 → П
все
конец

Если в ряду есть красные кубики, то, дойдя до первого из них, Муравей остановится; при этом число, записанное в его памяти, укажет номер соответствующей горизонтали. Если красных кубиков в ряду нет, то Муравей остановится на верхней горизонтали, но в памяти его будет записан 0.

Очень хорошо иллюстрируются с помощью Муравья циклические перестановки линейных таблиц.

Пусть на доске стоит ряд кубиков a_1, a_2, a_3, a_4 (рис. 6). Переставим их в следующем порядке: a_1, a_1, a_2, a_1 .

Предполагая в дальнейшем показать перестановку таблицы, разберем с учащимися



вариант решения, использующий дополнительное поле G6 (выполняющее роль вспомогательной переменной), т. е. предложим следующую программу:

```

начало
ВПРАВО 1; ВВЕРХ 1; ВПРАВО 4; ВНИЗ 1;
ВПРАВО 1; ВНИЗ 1; ВЛЕВО 4; ВНИЗ 1;
ВЛЕВО 1; ВВЕРХ 1
конец
  
```

Если указанную программу использовать как тело цикла «повторить», то можно получить любую циклическую перестановку.

Если научить Муравья понимать то, что написано на кубике (считывать информацию), то возможности исполнителя существенно расширятся.

Можно решить задачу, эквивалентную отысканию максимального элемента таблицы. Записать в память Муравья максимальное из чисел, изображенных на кубиках (рис. 7).

Сначала Муравей запоминает число, стоящее на первом кубике. Затем, двигаясь вдоль ряда кубиков, исполнитель сравнивает содержимое памяти с встречающимися числами. Найдя большее число, Муравей записывает его в память (стирая предыдущее). Таким образом, когда Муравей просмотрит весь ряд, в его памяти будет записано максимальное число.

Обозначим число на кубике, стоящим перед Муравьем, через X. Тогда программа примет вид:

```

начало
X → П
повторить 5 раз
  ВПРАВО 1
  если X > П
    то X → П
  все
кц
конец
  
```

Далее можно решить задачу: упорядочить линейную таблицу из шести элементов по возрастанию их значения.

Покажем, как Муравей может переставить

кубик с максимальным числом в конец ряда (на поле G5 рис. 7).

```

начало
для К от 1 до 5 шаг 1
  выполнить
  X → П
  ВПРАВО 1
  если X < П
    то ВВЕРХ 2; ВПРАВО 1; ВВЕРХ 1;
    ВЛЕВО 1 ВВЕРХ 1; ВЛЕВО 1;
    ВНИЗ 2; ВЛЕВО К ВНИЗ 2;
    ВПРАВО К; ВНИЗ 1; ВПРАВО 1;
    ВВЕРХ 1
все
кц
конец
  
```

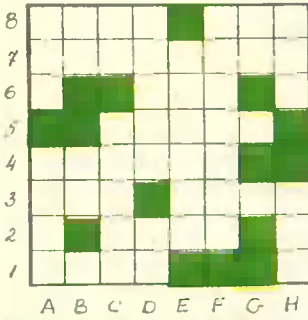
Теперь можно найти кубик с максимальным значением среди первых пяти и поставить его на поле F5; и т. д. Таким образом упорядочивается весь ряд кубиков. Заметим, что эта задача более трудная, так как требует использования вложенных циклов.

Программа упорядочения ряда из шести кубиков:

```

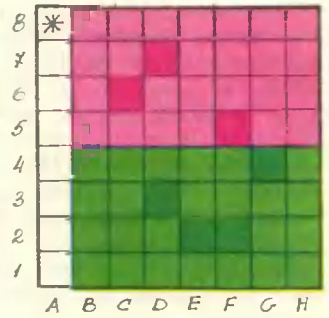
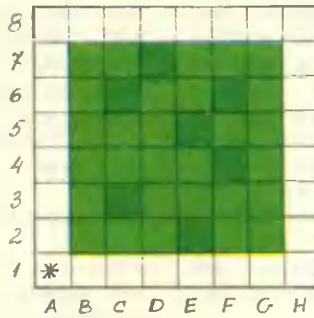
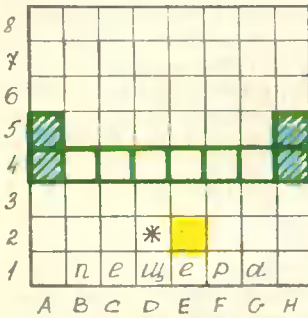
начало
для М от 5 до 1 шаг -1
  выполнить
  для К от 1 до М шаг 1
    выполнить
    X → П
    ВПРАВО 1
    если X < П
      то ВВЕРХ 2; ВПРАВО 1; ВВЕРХ 1;
      ВЛЕВО 1; ВВЕРХ 1; ВЛЕВО 1
      ВНИЗ 2; ВЛЕВО К; ВНИЗ 2
      ВПРАВО К; ВНИЗ 1; ВПРАВО 1
      ВВЕРХ 1
    все
    кц
    ВЛЕВО М
  кц
конец
  
```

Задачи, представленные в последнем разделе, трудны для учащихся IV—V классов, но они будут очень полезны при работе со старшеклассниками.



начало
пока ВВЕРХ МОЖНО или ВПРАВО МОЖНО
 повторять
 если ВВЕРХ МОЖНО
 то ВВЕРХ 1
 иначе ВПРАВО 1
 все
 КЦ
 конец

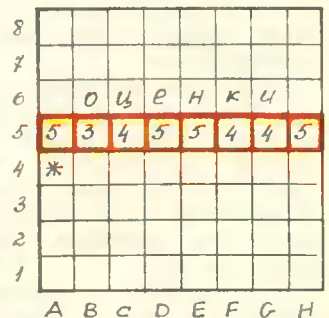
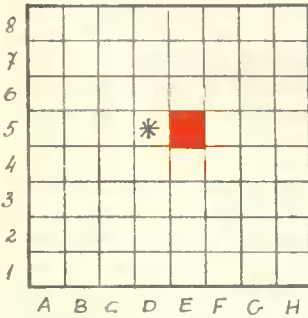
1. Где окажется МУРАВЕЙ, выполняя программу? Придумайте, как расставить препятствия на доске, чтобы МУРАВЕЙ не попал на поле Н8.



2. Вынести из пещеры сундук с золотом.

3. Подсчитать, сколько кубиков стоит на зеленой части доски.

4. Где больше кубиков: на зеленой или на красной части доски?



5. Переставить кубик на горизонталь, номер которой записан в памяти Муравья.

6. Найти на елке самый верхний синий кубик.

7. Сколько пятерок получил ученик за четверть?

Олимпиада юных программистов

С 5 по 18 августа 1987 г. в Друскининкае (Литовская ССР) прошла первая региональная школа юных программистов. В ней приняли участие 80 школьников пяти республик — Эстонии, Латвии, Литвы, Белоруссии и Российской Федерации. Для выявления лучших была проведена олимпиада по программированию: за 4 ч нужно было решить три задачи. Каждое решение — процедура или программа на Паскале, Бейсике или Си (по выбору участника).

Авторство работ было зашифровано. Листы бумаги с заранее написанными шифрами раздавались участникам случайным образом, каждому по два: большой — для записи решения задачи, маленький — «паспорт», на котором следовало написать свою фамилию и хранить. После объявления результатов по маленьким листочкам с шифрами и фамилиями были определены победители.

В задачах олимпиады, которые здесь приводятся, требовалось найти идею решения и алгоритм.

Задача 1. Прямоугольник, стороны которого выражены целыми числами m и n , разделен на квадраты размером 1×1 . Составьте процедуру (программу), которая находит число квадратов, пересекаемых диагональю прямоугольника.

Решение. Возможны два случая:

1. Числа m и n взаимно простые, т. е. не имеют общих делителей. Тогда диагональ не будет проходить через пересечения вертикалей и горизонталей, разделяющих прямоугольник на квадраты. Нетрудно доказать, что диагональ пересечет $m+n-1$ квадратов.

2. Числа m и n не являются взаимно простыми, т. е. имеют общий делитель, отличный от единицы. Пусть их наибольший общий делитель (НОД) равен d . Тогда исходный прямоугольник можно разделить на d^2 меньших прямоугольников со сторонами, равными m/d и n/d . При этом диагональ исходного прямоугольника будет проходить через d новых прямоугольников, в каждом из которых пересечет $m/d+n/d-1$ квадратов. Следовательно, она пересечет

$$x = (m/d + n/d - 1)d = m + n - d$$
 квадратов.

Осталось записать алгоритм — сделаем это на Паскале.

Решая эту задачу, многие вместо нахождения наибольшего общего делителя прове-

ряли, является ли число m делителем числа n или наоборот. Было немало работ, в которых использовались вещественные числа, например определялись длины отрезков вертикалей и горизонталей, которые отсекает диагональ, и проверялось, равны ли эти длины (вещественные числа) целым числам. Очевидно, что такой подход гарантии правильности результатов не дает.

Задача 2. Исходные данные — последовательность двузначных натуральных чисел. Длина последовательности неизвестна. Признаком ее конца является ноль (он уже не принадлежит последовательности). Нужно написать программу, которая бы напечатала все числа, принадлежащие последовательности, в неубывающем порядке. Например, если дано

21 44 21 63 0,

то должно быть напечатано

21 21 44 63.

Решение. На первый взгляд, это обычная задача сортировки. Однако в ее формулировке есть две особенности: усложняющая (длина последовательности неизвестна) и упрощающая (члены последовательности — натуральные двузначные числа). Используя вторую особенность, можно успешно справиться с последовательностью неизвестной длины, сохраняя не сами члены последовательности, а число вхождений каждого члена в последовательность. Приведем программу.

Лишь немногие школьники воспользовались тем, что члены последовательности — двузначные натуральные числа, и написали наиболее рациональные программы. В некоторых решениях для работы с последовательностью неизвестной длины были использованы рекурсия или динамические структуры данных, однако большинство вопреки формулировке задачи ограничило длину последовательности, причем лишь немногие из них написали операторы для вывода предупреждающих сообщений, если длина последовательности оказалась больше, чем заданная в программе.

Задача 3. В игре на отгадывание чисел «Быки и коровы» компьютер играет роль загадывающего. Составьте процедуру, которая по двум четырехзначным натуральным числам («задуманному» компьютером и предлагаемому человеком) определяет, сколько цифр отгадано правильно и сколько цифр

```

1      function f (m, n : integer) : integer;
        function нод (a, b : integer) : integer;
          { наибольший общий делитель }
        begin
          if a = 0 then нод := b
            else нод := нод (b mod a, a)
          end;
        begin
          f := m + n - нод(m, n)
        end;

2      program nnn (input, output);
        const min = 10; {наименьший член последовательности}
              max = 99; {наибольший член последовательности}
        var a : 0..max; {член последовательности или ноль}
            seq : array [min..max] of integer;
                {последовательность в упакованном виде}
            i : min..max;
            j : integer;
        begin
          for i := min to max do seq[i] := 0;
          read(a);
          while a in [min..max] do
            begin
              seq[a] := seq[a] + 1;
              read(a)
            end;
          for i := min to max do
            for j := 1 to seq[i] do
              write(i);
            writeln
          end.

3      procedure игра (a, b : integer; { данные числа }
                      var m, { число отгаданных цифр }
                        n : { и стоящих на своем месте }
                          integer);
        type digits = array [1..4] of 1..9;
        var i, j : 1..4;
            aa, bb : digits;
        procedure перевод (x : integer; var xx : digits);
          { перевод числа x в цифры xx }
          var i : 1..4;
        begin
          for i := 4 downto 1 do
            begin
              xx[i] := x mod 10;
              x := x div 10
            end
          end;
        begin
          перевод (a, aa);
          перевод (b, bb);
          m := 0; n := 0;
          for i := 1 to 4 do
            for j := 1 to 4 do
              if aa[i] = bb[j]
                then begin
                  m := m + 1;
                  if i = j then n := n + 1
                end
            end;
        end;

```

стоит на задуманном месте. Все цифры различны и не равны нулю. Пример: задуманное число 9481, предлагаемое число 3418, результаты 3 и 1.

Решение. Идея его проста: оба числа разлагаются на цифры процедурой *перевод*, а затем цифры попарно сравниваются. С этой задачей успешно справилось большинство участников олимпиады.

Региональная олимпиада проводилась впервые, естественно, в ее организации были и сложности. Вкратце остановимся на двух из них.

Выполнение, проверка и оценка работ производились вручную, без использования ЭВМ. Причина — большое число участников олимпиады и малое число компьютеров (с Паскалем — только 4). Для нормальной работы каждый участник должен иметь компьютер. Очевидно, это трудно осуществимо, поэтому можно предложить проводить олимпиады в два тура: первый — в «безмашин-

ном» варианте, второй — с ЭВМ. Конечно, имеются в виду ПЭВМ с дисками, а не рабочие места школьного КВТ.

Программы разрешалось писать на трех языках: Паскале, Бейсике и Си, поскольку на владение ими указали участники школы при опросе. Использовани Бейсика сильно затрудняло сравнение и оценку работ: для сохранения равных условий для участников нельзя было в достаточной мере учитывать такие важные факторы, как удобочитаемость, структурность, лаконичность программ, так как в Бейсике отсутствуют соответствующие средства. В будущем необходимо пользоваться только сравнимыми языками высокого уровня, сообщая об этом заранее, в информационных материалах о проведении школы. Некоторые дополнительные знания по конкретному языку могут быть получены и во время региональной школы до олимпиады на лекциях, консультациях, семинарах, при практической работе.

106

Оптические диски

Цифровые оптические диски — удобный носитель сверхбольших объемов информации с относительно быстрым доступом. Запись двоичной информации обычно производится лучом мощного лазера, формирующим микроуглубления или пузырьки на поверхности тонкой металлической или полимерной пленки; происходящие при этом локальные изменения необратимы. Чтение информации осуществляется маломощным лазером. Возможность фокусировки лазерного луча обеспечивает высокую плотность записи. Оптические диски с одноразовой записью (ОД) получили широкое распространение в первую очередь благодаря своей большой информационной емкости: сейчас на ОД диаметром 305 мм можно записать 1024М байт, что примерно в 1,5 тыс. раз больше, чем вмещается на обычную дискету. По стоимости оборудования в расчете на 1 М байт

ЧТО? МОЖЕТ ЭВМ!

ОД обходится в 11 раз дешевле сменного магнитного диска и в 3,5 раза дешевле винчестерского диска. Долговечность записи на ОД — около 10 лет, а для лучших экземпляров — свыше 30 лет, тогда как информацию на МД приходится перезаписывать каждые 2—3 года.

В настоящее время разрабатываются две основные технологии оптической записи без разрушения носителя, что позволяет выпускать ОД с перезаписью. Первая основана на том, что записывающий лазерный луч вызы-

вает обратимый переход из кристаллического в аморфное состояние в окисной теллуриевой пленке. Во второй используется намагниченный поверхностный слой, а локальное повышение температуры, вызываемое записывающим лазером, используется для изменения направления намагничивания. Фирма «Вербатим» объявила о начале выпуска с 1988 г. термомагнитооптических дисков диаметром 3,5 дюйма с емкостью 40М байт.

ОД уже нашли широкое применение в самых различных областях издательской деятельности и рассматриваются как самостоятельный вид публикации, так как этот способ является экономичным и удобным для распространения больших объемов информации. Ожидается, что к 1988 г. около четверти информации, записываемой в двоичном виде, будет «попадать» на ОД, а дисководы для них будут встраиваться даже в ПЭВМ.



М. ДРАГАНОВ

Цели и задачи курса информатики в средних школах Болгарии

Обучить молодежь работать с вычислительной техникой — одна из стратегических задач системы образования. Наша партия и правительство создают все условия для ее решения. Сегодня каждая средняя школа у нас располагает в среднем 15 ПЭВМ (Правец-82, Правец-8М и др.).

Как в СССР, так и в НРБ использование компьютеров в процессе обучения рассматривается в двух аспектах: компьютеры в качестве средства обучения и компьютеры, как цель обучения. Формулировка второго аспекта не совсем точная и не отражает полностью его содержание, но принята из-за своей краткости. Вопросы, связанные с этим аспектом, включены в предмет «Информатика».

В наше время в условиях информационного бума умение отобрать информацию превращается в особенно ценное качество. То же относится и к умению выявлять и извлекать внутренние иерархические связи между информационными единицами в информационных массивах. Разнообразные виды деятельности, связанные с информацией, могут быть объединены в четыре основные группы: сбор, переработка, хранение и распространение. Комбинации этих видов деятельности определяют различные информационные процессы. Необходимость их автоматизации привела к разработке соответствующих методов и теоретических средств и к формированию информатики, как самостоятельной, фундаментальной научной дисциплины. Основным практическим средством автоматизации информационных процессов стали компьютерные и микропроцессорные системы. Среди множества информационных процессов выделим один фундаментальный процесс — управление: основа всякой дея-

тельности. Компьютерные технологии управления машинами, целыми производствами стали характерной особенностью нашего времени. Микропроцессорные системы, компьютерные технологии управления все шире входят во все сферы деятельности, в быт, становятся постоянным окружением. Отсюда необходимость в том, чтобы молодое поколение было подготовлено к труду и жизни в этих условиях. Молодым людям необходимо овладеть основами информатики, методами и средствами реализации информационной деятельности и процессов, а также приобрести навыки практической работы с вычислительной техникой. Все это привело к тому, что в средних школах был введен обязательный общеобразовательный предмет «Информатика».

Исходя из причин, вызвавших введение этого предмета, можно сформулировать его цели и задачи.

Первая, основная цель — дать учащимся знания о свойствах информации, методах и средствах ее сбора, переработки, хранения и распространения. Каждая из этих видов деятельности представляет собой ветвь информатики со своей базой и инструментарием, куда включены знания об алгоритмах и алгоритмических структурах. Целью является формирование у учащихся алгоритмического мышления. Для достижения этой цели, в принципе, компьютер не обязателен. Однако после изучения информационных процессов и функциональных особенностей компьютеров, как информационных машин, возникает необходимость в изучении алгоритмов, моделирующих эти процессы, а следовательно и необходимость в изучении методов и языков программирования.

Овладение основами программирования — вторая основная цель обучения по «Информатике». Но это не значит, что каждый учащийся должен стать программистом. Речь идет об основах знаний и умений. Принципы и основы программирования усваиваются легче с использованием конкретного языка. Мы выбрали для этой цели Бейсик, который изучить сравнительно легко. Изучая программирование, школьники лучше усваивают методы и средства информатики. Их знания становятся более глубокими и прочными. При составлении программ лучше воспринимаются алгоритмические и информационные структуры; формируются навыки правильной постановки практических задач, формализации проблем; знания о методах моделирования и умение составлять простые математические модели и др. Устраняется психологический барьер при работе с компьютером. Учащийся приобретает уверенность в своих силах. При овладении основами программирования и получении практических навыков учащийся сможет использовать компьютер при изучении других дисциплин. И прежде всего — математических и технических (в рамках профессионального обучения). Речь идет об использовании компьютера при решении задач вычислительного характера, при моделировании и исследовании процессов и явлений, при автоматизации некоторых инженерных задач и т. д. Для решения большинства имеются готовые пакеты программ.

Приобретение умения использовать готовое прикладное программное обеспечение — третья основная цель обучения информатике. Кроме умения работать с готовыми программами необходимо умение правильной интерпретации полученных результатов и их использования на практике. Достижение третьей цели — очень важное требование, так

как на практике использование готовых программ и работа с управляющими вычислительными системами (также работающими с готовыми программами) встречается чаще всего. Многие специалисты считают, что обучение по информатике следует ограничить этими рамками. Это мнение крайнее, но оно показывает, насколько важно приобретение учащимися практических умений работы с некоторыми из самых распространенных программных средств:

- текстовыми редакторами и системами текстообработки;
- электронными таблицами;
- базами данных;
- системами для автоматизации проектирования и конструирования и др.

Например, вся машинописная деятельность через несколько лет будет выполняться в основном с помощью текстообработывающих систем на базе универсальных микрокомпьютеров. Следовательно, учащиеся уже в школе должны овладеть практическими умениями в этом плане. Одновременно с тремя основными целями следует добиваться достижения двух дополнительных целей:

- приобретения общего представления о назначении и функциях операционных систем и практического умения пользоваться самыми важными функциями основной операционной системы школьных компьютеров;
- приобретение общего представления об устройстве микрокомпьютеров и микропроцессорных систем и принципах их работы.

Жизнь подтверждает, что компьютеризация производства и других видов деятельности — необратимый процесс. Обучение информатике в болгарских средних школах является первой ступенью. Следующая — компьютеризация всей системы образования.

П. АСЕНОВА

Состояние и перспективы обучения информатике в Болгарии

В последние годы в единой средней политехнической школе (ЕСПШ) Болгарии началось изучение основ информатики. Главные направления развития школьной информатики основываются на методической концепции, разработанной под руководством Министерства народного просвещения НРБ. Предусматривается формирование у учащихся

системы знаний и приобретение практических навыков в области информатики и вычислительной техники, необходимых как в процессе обучения, так и в дальнейшей трудовой деятельности.

Формы обучения различны: обязательная, свободно выбираемая подготовка (математика и информатика) и факультативная под-

готовка. Далее мы будем говорить о содержании обязательного обучения информатике.

Элементы информатики начинают изучать с VIII класса в курсе математики, а самостоятельная учебная дисциплина включается в учебный план с X класса (II курса техникумов и средних профессиональных технических школ (СПТШ)).

Изучение элементов информатики в курсе математики предполагает знакомство с алгоритмами решения задач (алгоритмами приближенного вычисления, решения уравнений и систем уравнений, решение задач на прогрессии, пределы, производные); применение микрокомпьютера в качестве калькулятора; применение компьютера для построения графиков и при изучении геометрических преобразований; ознакомление с линейным программированием, системами счисления, элементами булевой алгебры и др.

На уроках информатики в X классе (II курс техникума и СПТШ) изучаются три группы понятий.

Основные понятия информатики: информация, сообщение, информационная деятельность, информационный процесс.

ЭВМ как средство автоматизации информационных процессов.

Алгоритмы и средства их описания.

Курс информатики для XI класса (III курс техникумов и СПТШ) охватывает следующие темы:

Компьютеры и их математическое обеспечение: математические основы ЭВМ; архитектура ЭВМ; классификация компьютеров; операционные системы.

Структуры данных и средства их обработки: структуры данных; обработка текстовой и графической информации, базы данных.

Информатика: наука и практика: модели и моделирование, системы информационного обслуживания; системы управления; системы связи; системы проектирования; системы обучения; системы автоматизации научных исследований.

В XII классе предусматривается изучение математического моделирования.

Углубленное изучение информатики по профилю «оператор-программист» осуществляется в учебно-производственных комплексах.

В X классе (II курс техникумов и СПТШ) информатика изучается в объеме 34 часов по трем учебникам.

Самое широкое распространение получил учебник авторского коллектива в составе Бърнева П., Азълова П., Добрева Д., Бистерова Ц. По нему ведется обучение в ЕСПШ и техникумах. По другому учебнику (Ангелов А., Гъров К., Гавраилов О.) занимаются учащиеся СПТШ. Третий учебник (Да-

ковски Л., Сапунджиев Г., Радева С.) не получил широкого распространения. Для каждого учебника разработаны методические пособие и дискета с программами. Обучение осуществляется в двух вариантах: машинном и безмашинном. Каждый учебник обеспечивает оба варианта путем варьирования содержания и выбора соответствующей методики.

В учебнике П. Бърнева и других на фоне основных понятий информатики формируется представление об информатике как науке. *Информатика* — это наука об информации (рассматриваемой независимо от ее содержания), о ее структуре и свойствах, о разнообразных видах информационной деятельности, о методах и средствах автоматизации такой деятельности. Информатика рассматривается как наука, которая по своей универсальности и обобщенности стоит близко к философии.

С 1979 года Проблемная группа образования (ПГО) при Академии наук Болгарии и Министерстве народного просвещения проводит экспериментальное обучение по новой учебной программе с применением новых методов обучения. Информатика как самостоятельный учебный предмет изучается с V класса. Используются игровые формы обучения, дети знакомятся с языком программирования Лого (уже два года обучение ведется на адаптированной версии этого языка с болгарской лексикой). Осуществляется взаимное проникновение информатики и других учебных дисциплин. Эксперимент ПГО охватывает небольшое количество школ. К настоящему времени накоплен интересный опыт обучения информатике младших школьников.

Большое значение в эксперименте отводится учебной среде, которая включает комплекс учебных средств: компьютерную технику, информационную базу, языковые и операционные средства программирования, учебные программные средства, учебную и учебно-методическую литературу. Серьезное внимание уделяется повышению квалификации учителей.

На основе отечественного и зарубежного опыта намечены перспективы развития школьной информатики на ближайшее будущее.

Выделим *основные понятия*, усвоение которых предусмотрено содержанием перспективного курса школьной информатики: информация и сообщение, информационная деятельность и информационный процесс, системы и управление системами, модели и моделирование (информационное моделирование), язык, алгоритм, технология. Учащиеся познакомятся также с различными

информационными средствами: связи, информационно-множительными, сенсорными и регистрирующими устройствами, средствами для хранения и поиска информации, средствами вычисления, универсальными средствами информатики (компьютерами), информационными средствами со встроенными микропроцессорами; *с основными информационными службами*: библиотеками, архивами, почтами, справочными и рекламными службами, вычислительными центрами и др. Раздел «Методы и формы работы с информацией» предполагает изучение способов накопления, хранения и поиска, обработки и распространения информации. В содержание курса информатики включены вопросы, посвященные проблемам общения. В специализированных школах предлагается изучать программирование и основы математического обеспечения. Предусматривается знакомство школьников с философскими, социальными, правовыми, морально-этическими и гигиеническими вопросами информатики.

Реализация новой перспективной программы требует разработки систематического курса информатики с первого до последнего класса ЕСПШ. Необходимо содержательное взаимодействие между информатикой и другими учебными дисциплинами. Остро ставится вопрос о разработке новой методики преподавания информатики.

К настоящему времени основные трудности в обучении информатике состоят в следующем. Предмет этой науки во многом неоднозначен и дискуссионен. Это неизбежно отражается на содержании данной учебной дисциплины и на методике ее преподавания. В мире существует множество концепций обучения информатике, различны представления о содержании, методах и средствах ее преподавания. Практика покажет, что является целесообразным. Преподавательские кадры, обучающие информатике в школе, не обладают достаточной квалификацией. Проблемы существуют и в области материальной базы (техника, учебники, учебные пособия, дополнительная научно-популярная литература).

Для преодоления этих трудностей в НРБ создана специальная сеть учреждений и организаций. При Министерстве народного просвещения создана центральная учебно-методическая комиссия по информатике, а в округах — окружные комиссии. Главная их задача — наблюдать за обучением и оказывать методическую помощь школе. В Министерстве просвещения начал изда-

ваться бюллетень «Обучението по информатика». В журналах «Обучението по математика» и «Народна просвета» публикуются методические материалы. Для учащихся издаются журналы «Математика», «Компьютер за вас» и др. Работу по повышению квалификации педагогических кадров проводят факультет математики и механики Софийского университета, Центральный институт усовершенствования учителей, Высший машинный электротехнический институт и другие вузы. Союз математиков Болгарии организует ежемесячные встречи учителей по обмену передовым опытом. Разработка программных продуктов осуществляется в Лаборатории автоматизированных систем факультета математики и механики Софийского университета и др. В планах государственных издательств «Народна просвета», «Техника», «Наука и изкуство» и других предусмотрен выпуск учебных методических пособий и научно-популярной литературы по информатике.

Болгарские специалисты сотрудничают с советскими, а также со специалистами других стран, участвуют в выполнении комплексной программы по компьютеризации обучения в рамках СЭВ.

Таким образом, в Болгарии пытаются комплексно решать многообразные и сложные проблемы внедрения и преподавания информатики.

Л и т е р а т у р а

1. *Бърнев П.* Информация и управление. София: Народна просвета, 1985.
2. *Бърнев П., Азълов П., Бърнева П., Добрев Д., Керпеджиев Ст., Попратилов Ст.* Перспективна програма за преподаване на информатика в ЕСПУ // Обучението по математика. 1986. № 3.
3. *Бърнев П., Азълов П., Бърнева П., Добрев Д., Керпеджиев Ст., Попратилов Ст.* Проект за перспективна учебна програма по информатика за ЕСПУ // Обучението по математика. 1987. № 1.
4. *Николов Р.* Учебна среда по информатика. Дис. к. м. н. 1987.
5. *Николов Р., Сендова Е.* Информатика: Учебник: В 2 ч. София: ПГО, 1985.
6. Система на учебно-възпитателна работа в 8—12 клас на ЕСПТУ: Проект. София: МНП, 1986.
7. Учебники и учебные пособия по информатике для X класса.

Перспективные устройства ввода

Самый трудоемкий процесс в цепи автоматизированной обработки данных — ввод информации в ЭВМ. Этот дорогостоящий и длительный этап порождает вдобавок наибольшее число ошибок, поэтому понятна многочисленность попыток решить эту проблему.

Одно из направлений таких работ — создание систем речевого ввода. В 70-е гг. в США был создан ряд специализированных ЭВМ, способных воспринимать речь при словарном запасе в 1000 слов. Наиболее удачной считалась ЭВМ «Гарми», разработанная в университете Карнеги—Меллона (Питсбург). Эта машина воспринимала в тихом помещении речь трех мужчин и двух женщин со средним уровнем ошибок 2%.

В фирме ИБМ разработано устройство распознавания речи на базе персональной ЭВМ с двумя дополнительными платами. Оно распознает 5000 слов, автоматически настраиваясь на индивидуальные речевые особенности говорящего за 20 мин начитывания им определенного текста.

ЭВМ уже способны оперировать 20 000 слов, а это около 98% слов, используемых в переписке. Общий уровень ошибок при распознавании в этом случае — 6,3%.

Интересный подход развивает французская фирма «Анатекс». Там разработан графический планшет с соответствующим программным обеспечением. При подключении к ПЭВМ «Макинтош» планшет позволяет вводить в компьютер рукописный текст. Для предварительного обучения достаточно написать все буквы алфавита, и устройство запомнит

почерк. Такой комплекс удобно применять при обработке текстов и проведении финансовых расчетов.

«Левши» от электроники

Страны Юго-Восточной Азии в течение последних лет переживают «бум» компьютерной техники. В Бангкоке, Джакарте, Маниле, Сингапуре сформировались даже своеобразные «толкучки», где в небольших магазинчиках, по ценам значительно ниже предлагаемых в каталогах известных японских и американских фирм, средней руки бизнесмены или специалист может приобрести компьютер с цветным дисплеем, печатающее устройство, блоки памяти и все остальное. Разумеется, это подделки, которые изготавливаются чуть ли не на глазах потребителя.

В общем-то это лицензионное «пиратство». Однако потребители серьезно утверждают, что качество у подделок выше оригиналов, хотя значительная часть компонентов местного производства. Работа-то штучная. Начался же «бум» из-за того, что во многих городах Юго-Восточной Азии по ряду причин, прежде всего климатических, скачет напряжение в электросети. Дорогая техника выходит из строя,

импортные запчасти не по карману потребителям по обычным ценам. Спрос в итоге породил предложение.

Интересно, что «левши» от электроники в Юго-Восточной Азии сами становятся предметом купли-продажи. Сотни (а ряд фирм считает, что и тысячи) их, подмеченные специальными агентствами, без шума выезжают на работу в Австралию, Новую Зеландию, Канаду, Западную Европу, даже в США. Как носители «практических» знаний японской компьютерной технологии, они используются в конкурирующих производствах. Дальновидные политики здесь начинают бить тревогу. Обучение в технических институтах за рубежом одного человека обходится, например, в Индонезии в 60 тыс. долларов. Когда такого человека сманивают, потери очевидны: втуне остаются истраченные средства на его образование, а знания и опыт послужат только увеличению отставания по сравнению с промышленно развитыми странами.

(По материалам «Правды».)

«Самопроверка»

За рубежом создан сравнительно недорогой и простой в эксплуатации программируемый тестер для проверки входных и выходных цепей интегральных схем. Встроенное программное обеспечение позволяет быстро проверить сложные схемы более 1500 различных устройств, включая блоки памяти. Результаты проверки высвечиваются на индикаторе в форме pass (норма) или fail (нет нормы). Подобная информация может быть выведена на печать по желанию пользователя. Так электроника проверяет электронику.

ЧТО? МОЖЕТ ЭВМ

Педагогические кадры

Психолого-педагогическая подготовка специалистов народного хозяйства для преподавания в школе

(Примерные учебно-тематический план и программа специального курса психолого-педагогической подготовки специалистов народного хозяйства, привлекаемых для преподавания в средних общеобразовательных школах курса «Основы информатики и вычислительной техники»)*

112

Введение в средних школах, ПТУ и средних специальных учебных заведениях страны общеобразовательного предмета «Основы информатики и вычислительной техники» потребовало в ряде территорий привлечения к преподаванию названного курса инженерных работников, не имеющих специальной психолого-педагогической подготовки.

Практика свидетельствует, что значительная часть привлекаемых к работе с учащимися специалистов народного хозяйства остро нуждается в приобретении знаний в области общей, возрастной и педагогической психологии, педагогики и методики преподавания предмета.

Для оказания учреждениям методической службы практической помощи в организации работы с указанной категорией работников предлагаются настоящие учебно-тематический план и программа специального курса, на изучение которого отводится 72 ч. При этом на лекции учебным планом отводится 45 ч, а на семинарские занятия — 27 ч. Зна-

чительный объем лекционных занятий объясняется тем, что слушателям необходимо дать максимум новой информации, необходимой им для последующего самостоятельного образования.

Предполагается, что в результате обучения по настоящей программе слушатели получат минимум знаний по психологическим основам обучения, вопросам дидактики, общим и частным вопросам методики преподавания основ информатики и вычислительной техники. Кроме того, научно-методический анализ курса «Основы информатики и вычислительной техники» предусматривает обзор всех тем, предлагаемых для девяти- и десятиклассников, а также более углубленное изучение разделов, которые, по мнению практиков, представляют наибольшую трудность для учащихся.

По всем разделам учебного плана необходимо широко использовать материалы XXVII съезда КПСС, документы школьной реформы, а также разработки уроков, опубликованных в «Учительской газете».

При подготовке раздела «Общие и частные вопросы методики преподавания курса «Основы информатики и вычислительной техники» были использованы материалы, разработанные лабораторией обучения информатике НИИ СиМО АПН СССР.

* Авторы-составители: *Лесов Л. Д.* — заведующий учебно-методическим кабинетом руководящих кадров Гомельского областного института усовершенствования учителей; *Каплунович И. Я.* — доцент Мозырского педагогического института, кандидат психологических наук.

Учебно-тематический план

Наименования тем	Всего часов	Из них:	
		лекции	практические и семинарские занятия
1	2	3	4
I. Психологические основы обучения школьников	6	6	
1. Психологические основы процесса обучения	4	4	
2. Возрастные и индивидуальные особенности познавательной деятельности учащихся	2	2	

1	2	3	4
II. Вопросы дидактики	20	14	6
1. Принципы дидактики	2	2	
2. Обучение, воспитание и развитие как единый процесс	2	2	
3. Методы обучения и пути их совершенствования	4	2	2
4. Современный урок: требования к нему и пути его совершенствования	6	4	2
5. Педагогическое руководство формированием общих умений и навыков	2	2	
6. Управление самостоятельной работой школьников и подготовка их к самообразованию	4	2	2
III. Научная организация педагогического труда	7	4	3
1. Принципы научной организации педагогического труда	2	2	
2. Психологические аспекты деятельности учителя	5	2	3
IV. Общие и частные вопросы методики преподавания курса «Основы информатики и вычислительной техники»	39	21	18
1. Методика преподавания информатики	1	1	
2. Структура и содержание курса информатики	2	2	
3. Межпредметные связи курса основ информатики и вычислительной техники	2	1	1
4. Кабинет вычислительной техники в школе	3	1	2
5. Организация учебных занятий	4	2	2
6. Контроль и оценка результатов обучения	2	2	
7. Особенности использования ТСО при обучении школьников основам информатики и вычислительной техники	2	2	
8. Содержание и организация внеклассной работы по информатике	2	2	
9. Методика изучения тем курса IX класса	8	4	4
10. Методика изучения тем курса X класса	8	4	4
11. Практикум по составлению программ и их выполнение на ЭВМ	5		5
Итого:	72	45	27

113

ПРОГРАММА

I Психологические основы обучения школьников

1. Психологические основы процесса обучения

а) Проблема обучения и развития.

Ленинская теория познания и процесс обучения. Психологический анализ процесса познания. Сущность процесса учения, его особенности и отличие от научного познания. Соотношение обучения и развития. Основной закон функционирования высших психических функций. Уровень актуального и зона ближайшего развития. Необходимость опережающего обучения в психическом развитии личности. Понятие о развивающем обучении, его основные проблемы.

б) Характеристика процесса усвоения знаний.

Понятие об усвоении знаний. Усвоение знаний и умственное развитие учащихся. Процесс и этапы усвоения знаний. Виды внимания, их проявление и формирование в учебном процессе. Учет объема и распределения внимания при усвоении знаний. Внимание и самоконтроль. Оптимальные условия восприятия учебного материала (словесного и наглядного). Наблюдение и его роль в познании. Психологические требования к работе с наглядными пособиями. Роль и место памяти в познании. Закономерность запоминания, сохранения и воспроизведения учебного материала.

Мышление — высшая ступень познания. Приемы усвоения, их особенности и роль в познании. Усвое-

ние знаний и формирование интеллектуальных умений — единый процесс. Абстрактное и конкретное, теоретическое и эмпирическое, репродуктивное и творческое мышление при усвоении знаний. Особенности и структура мышления. Роль речи в усвоении знаний.

Знания, умения, навыки: их содержание и взаимосвязь, этапы овладения ими. Обучение знаниям, умениям и навыкам как цель и как средство формирования способов умственной деятельности, учебной работы старшеклассников.

Овладение способами умственной деятельности и умственное развитие как единый процесс. Влияние содержания усваиваемого материала и методики его изложения на умственное развитие. Конкретно-практический (от конкретного — к абстрактному) и абстрактно-логический (от абстрактного — к логическому) пути обучения. Методы управления процессом усвоения.

в) Активизация мыслительной деятельности учащихся.

Сущность познавательной активности. Усвоение знаний как активный процесс. Зависимость качества усвоения от степени интеллектуальной активности учащихся, условий обобщения. Учет психологических закономерностей познавательных процессов — основа активизации учащихся на уроке. Учебно-познавательная и учебно-исследовательская деятельность школьников, их соотношение.

Создание проблемных ситуаций как путь активизации мышления учащихся. Психологическая структура и классификация основных типов проб-

лемных ситуаций. Правила создания и управления активизацией в проблемных ситуациях.

Межпредметные связи и умственная активность учащихся.

Значение программированного обучения в активизации познавательной деятельности. Переработка информации учащимися в условиях обычного, проблемного и программированного обучения.

Оптимизация обучения как фактор развития самостоятельности и активности учащихся. Условия оптимизации: учет уровня актуального и зоны ближайшего развития, сензитивности к определенным сферам деятельности, склонности к систематическому занятию ими.

г) Мотивация учения.

Учебная мотивация как совокупность побуждения, определяющая направленность школьника на различные стороны учебной деятельности. Источники учебной мотивации: внешние (учебные ситуации), внутренние (потребности, идеалы и т. д.), личностные (успех, самоутверждение и т. д.).

Познавательные и социальные мотивы учения, их особенности. Мотивы «понимаемые» и реально действующие, их взаимодействие и влияние на отношение к учению. Побуждающие и смыслообразующие мотивы.

Интерес как познавательно-динамическое отношение ученика к учебной деятельности. Познавательный интерес как мотив учения, определяющий познавательную направленность и активность ученика. Мотив самооценки и саморазвития в общей структуре мотивации учения старшеклассников.

Формирование мотивации учебной деятельности. Влияние содержания учебного предмета на отношение к нему учащихся. Организация учебной деятельности как условие формирования положительных мотивов учения. Роль социальных факторов (семьи, учителя, внеклассных форм работы) в развитии учебных мотивов.

2. *Возрастные и индивидуальные особенности познавательной деятельности старшеклассников*

а) Общая характеристика возрастных особенностей и учебной деятельности старшеклассников.

Социальная ситуация развития, ведущая деятельность и основные психические новообразования в ранней юности. Психологические особенности учебной деятельности и ее структуры. Учебно-познавательные мотивы. Особенности познавательной деятельности старшеклассников: стремление к научным способам познания, к самостоятельному преобразованию учебных задач, к исследовательскому поиску по типу эксперимента. Специфические особенности умственной деятельности старшеклассников.

Основные формы и методы изучения возрастных особенностей учащихся.

б) Индивидуальные особенности.

Проблема дифференциации учащихся, учет их индивидуальных различий. Психологические различия в содержании, характере, способах учебной работы и умственной деятельности старшеклассников. Объективная сложность и субъективная трудность усвоения содержания материала, решения задачи. Типология трудностей в формировании знаний, умений и навыков.

Психологическая характеристика основных типов неуспевающих; методы диагностики. Психоло-

гические основы индивидуализации обучения. Адаптивные программы обучения.

Методы определения индивидуальных особенностей учащихся.

II Вопросы дидактики

1. Принципы дидактики

Цель коммунистического воспитания — всестороннее развитие личности в процессе обучения. Целенаправленное формирование у учащихся диалектико-материалистического мировоззрения и коммунистической нравственности, способствующих всестороннему развитию личности каждого школьника: его мышления, воли, характера, духовных потребностей, способностей. Их воспитание в процессе обучения.

Принципы дидактики как общие положения, определяющие содержание, методы и формы обучения. Факторы, обуславливающие принципы дидактики. Связь дидактических принципов с целями, поставленными обществом перед школой, с закономерностями развивающего обучения.

Принцип научности. Изучение прочно установленных в науке положений. Воспитание у учащихся диалектического подхода к изучаемым предметам и явлениям. Раскрытие новейших достижений науки. Ознакомление учащихся с формами и методами научной организации труда.

Принцип сознательности и творческой активности учащихся при руководящей роли учителя. Необходимость обеспечения оптимального сочетания педагогического руководства и самостоятельности учащихся в обучении. Подведение учащихся к ясному пониманию цели и задач предстоящей работы. Обучение учащихся приемам творческой деятельности, обеспечивающим активность в усвоении новых знаний.

Принцип наглядности в обучении. Роль и место наглядности в обучении. Виды наглядности, используемые в учебном процессе.

Принцип системности: усвоение учащимися знаний в их логической связи, преемственности и последовательности.

Принцип связи обучения с жизнью, с практикой коммунистического строительства. Образование как активная сила преобразования человека. Изложение основ наук как обобщенных и систематизированных результатов общественно-исторической практики. Связь изложения теории с личной практикой учащихся. Производительный труд и общественно полезная деятельность как важнейшие виды практики школьников.

Принцип прочности результатов обучения и развития познавательных возможностей учащихся. Принцип положительного эмоционального фона обучения. Принцип коллективного характера обучения и учета индивидуальных особенностей.

Взаимосвязь принципов дидактики.

2. *Обучение, воспитание и развитие как единый процесс*

Психолого-педагогическое содержание понятий «обучение», «воспитание», «развитие». Источники, условия и движущие силы развития личности. Роль наследственности, среды и воспитания в этом развитии. Ведущая роль обучения и воспитания в формировании человека.

Единство обучения, воспитания и развития как

важнейшая закономерность педагогического процесса. Раскрытие советской дидактикой основ и путей решения проблемы о связи обучения, воспитания и развития.

Объективные условия воспитывающего обучения; содержание, методы и формы организации такого обучения в их взаимосвязи.

Соотношение понятий «всестороннее» и «гармоничное развитие личности». Основная цель коммунистического воспитания и основные направления всестороннего развития личности: идейно-политическое, трудовое, нравственное, эстетическое, физическое воспитание; содержание и задачи основных направлений; методы изучения уровня воспитанности учащихся.

3. Методы обучения и пути их совершенствования

XXVII съезд КПСС, Основные направления реформы общеобразовательной и профессиональной школы о совершенствовании методов обучения и повышения качества знаний.

Понятие методов и приемов обучения.

Пути совершенствования методов изложения материала. Наглядные методы, приемы эффективного их использования. Способы совершенствования практических методов. Основные пути повышения воспитательной эффективности методов обучения.

Факторы, обусловившие необходимость кардинального совершенствования методов обучения в современной школе. Приведение методов обучения в соответствие с содержанием образования, требованиями жизни.

Функции методов обучения: образовательная, воспитательная, развивающая, мотивирующая. Дидактическое единство функций методов обучения.

Классификация методов обучения по источникам знаний, по дидактическим целям, по уровню познавательной самостоятельности учащихся. Основные методы проблемного, программированного, развивающего обучения. Типизация методов интенсивного обучения, основу которого составляет деятельный подход; методы организации учебно-познавательной деятельности, ее стимулирования и мотивации; методы контроля и самоконтроля учебной деятельности.

Диалектичность каждого метода, его сильные и слабые стороны.

Необходимость выбора в каждом отдельном случае оптимального сочетания методов обучения, исходя из целей урока, его содержания, реальных учебных возможностей как ученического коллектива, так и отдельных учащихся, состояния учебно-материального оснащения, личных качеств (психологических особенностей) учителя, а также других факторов.

Рациональные приемы организации самостоятельной работы учащихся и активизация их познавательной деятельности, используемые учителями-новаторами (Ш. А. Амонашвили, С. Н. Лысенковой, В. Ф. Шаталовым, М. П. Щетининым и др.).

Причины недопустимости абсолютизации отдельных методов обучения.

4. Современный урок. Требования к нему и пути его совершенствования

Изменения в организации учебного процесса в свете требований Основных направлений реформы общеобразовательной и профессиональной школы.

Урок как основная форма обучения и воспитания учащихся. Общие требования к уроку и пути его совершенствования в свете требований реформы школы.

Цели и задачи урока. Требования к отбору учебного материала. Реализация идей и учет особенностей учебной программы. Выбор методов обучения на уроке.

Различные виды и типы уроков, их особенности; структура урока.

Идеологические требования к современному уроку. Коммунистическая идейность, партийность урока. Обеспечение неразрывного единства обучения и воспитания; формирование у учащихся диалектико-материалистического мировоззрения; воспитание высоких моральных качеств строителя коммунизма, привитие школьникам умения распознавать и бороться против буржуазной идеологии и морали. Тесная связь обучения с жизнью; привитие учащимся любви к труду и людям труда.

Дидактические требования. Реализация на уроке принципов дидактики. Использование новейших данных педагогики и психологии, достижений науки, основы которой изучает данный предмет. Обеспечение высокой познавательной активности учащихся, оптимальное сочетание изложения учителем программного материала и самостоятельной работы учащихся, выполнение ими проблемных заданий и решение познавательных задач.

Индивидуализация и дифференциация на уроке, особенности использования дидактических материалов. Целесообразное использование наглядных пособий, технических средств обучения. Выработка у учащихся практических умений и навыков; первоначальная подготовка учащихся к тому или иному виду трудовой деятельности; профессиональная ориентация школьников.

Привитие учащимся общеучебных умений и навыков. Преемственность в обучении, связь данного урока с предыдущим, осуществление внутри- и межпредметных связей.

Применение эффективных методов учета знаний. Соблюдение на уроке единого орфографического режима.

Требования к оценке знаний учащихся на уроке. Психологические основы оценки. Оценка и отметка. Психолого-педагогические требования к оценке. Основные формы деятельности по формированию умения содержательной оценки: оценочная деятельность учителя, коллективная учебно-познавательная деятельность учащихся, самостоятельная оценочная активность. Критерии оценок как педагогическая, психологическая и дидактическая проблема. Место и роль проверки в оценке знаний на уроке.

Организационные требования. Организационная четкость, рациональное использование учебного времени. Ясная постановка перед учащимися учебных задач. Использование различных типов уроков, вариативность структуры урока. Роль сознательной дисциплины учащихся на уроке. Подведение итогов урока.

Необходимость постоянного присутствия у педагога самообладания, внимательности, самоконтроля. Значение высокой требовательности учителя, сочетающейся с его доброжелательностью, справедливостью, уважением к личности учащихся.

Гигиенические требования: обеспечение чистого

воздуха в классе, температурного режима, освещенности: соблюдение учащимися правильной посадки.

Самоанализ и самооценка учителем урока.
Основные критерии результативности урока.

5. Педагогическое руководство формированием общеучебных умений и навыков

Умения и навыки, их характеристика и отличие. Структура умений и навыков, их компоненты. Критерии их сформированности. Взаимосвязь знаний, умений, навыков.

Виды навыков и умений. Классификация учебных умений и навыков, их характеристика. Место и значение их в учебно-познавательном труде. Возрастная характеристика общих учебных умений. Специальные умения и навыки. Требования современных программ к системе специальных умений и навыков.

Этапы формирования навыков и умений. Объяснение и показ в ходе формирования навыка, их сочетание. Требования к качеству показа и объяснения. Ход формирования навыков, анализ упражнений. Причины неравномерности развития навыка в процессе упражнений. Роль методов обучения в эффективном формировании умений и навыков.

6. Управление самостоятельной работой школьников и подготовка их к самообразованию

Подготовка учащихся к самообразованию как важнейшая социально-педагогическая проблема. Формирование у школьников умений и навыков учебного труда как важнейшее условие и средство повышения качества обучения, преодоления перегрузки.

Сущность самостоятельной работы учащихся. Педагогическое руководство как основное условие выполнения самостоятельной работы. Место самостоятельных работ в процессе обучения. Виды самостоятельных работ. Работа учащихся с учебником. Организация самостоятельной работы учащихся при решении задач. Пути рациональной организации самостоятельной работы школьников. Приемы самостоятельной индивидуальной деятельности.

Классификация общеучебных умений и навыков (учебно-организационные, учебно-информационные и учебно-интеллектуальные). Структура деятельности учителя по формированию у учащихся каждого из основных умений. Научная организация труда — важнейшее условие эффективности самообразовательной работы школьников.

III. Научная организация педагогического труда

Возрастание требований к учебно-воспитательному процессу и качеству знаний учащихся в свете решений XXVII съезда КПСС и реформы общеобразовательной и профессиональной школы. Сущность и принципы научной организации педагогического труда.

Специфика труда преподавателя, основные направления научной организации его труда. Психологические аспекты педагогической деятельности.

Сущность гностических, конструктивных, орга-

низаторских и коммуникативных умений учителя. Формирование этих умений в процессе педагогического труда.

Педагогические способности, их проявление и формирование.

Время преподавателя. Анализ бюджета времени. Личная техника подготовки к занятиям: уточнение темы, четкое определение цели занятия, подбор материалов, составление плана урока, репетиция проведения занятия (отработка содержания, интонации, жесты, определение затрат времени и т. д.). Методика комплектования и использование личной библиотеки.

Организация учителем проведения занятий, воспитательной работы. Совершенствование условий его труда.

Самообразование как важнейший фактор в системе непрерывного повышения педагогического мастерства и профессиональной квалификации. Роль и содержание самообразования в период интенсификации и совершенствования техники и технологии.

IV. Общие и частные вопросы методики преподавания курса

1. Методика преподавания информатики

Информатика как наука и как учебный предмет. Общеобразовательные, воспитательные и развивающие цели, реализуемые в процессе изучения дисциплины. Значение курса в общем образовании.

Общие методические вопросы преподавания. Методические особенности учебных пособий по основам информатики и вычислительной техники; особенности методики преподавания курса.

2. Структура и содержание курса информатики

«Основы информатики и вычислительной техники» — новый учебный предмет общеобразовательной школы. Значение его преподавания в современной школе, необходимость введения основ информатики в содержание общего среднего образования. Структура и содержание школьного курса информатики. Принципы его построения; основные идеи, понятия, положения и методы. Основные цели обучения информатике: формирование компьютерной грамотности школьников, прочных знаний, умений и навыков по основным разделам курса, формирование диалектико-материалистического мировоззрения, воспитание советского патриотизма и интернационализма, идейно-политического воспитания и профессиональная ориентация, развитие алгоритмической культуры и познавательных способностей, привитие учащимся интереса к основам информатики и вычислительной техники. Прикладная и политехническая направленность курса.

Задачи и упражнения в курсе информатики.

3. Межпредметные связи курса основ информатики и вычислительной техники

Место информатики в системе научного знания. Связь обучения информатике с преподаванием других предметов (математики, физики, обществоведения, трудового обучения, родного и иностранного языков и т. д.), их взаимопроникновение.

4. Кабинет вычислительной техники в школе Оборудование кабинета.

Организация и методика проведения занятий в кабинете вычислительной техники.

Правила техники безопасности и производственной санитарии.

5. Организация учебных занятий

Организационная форма учебных занятий по информатике, их краткая характеристика. Требования к современному уроку информатики.

Формы внеклассной работы. Организация и методика проведения экскурсий.

Планирование учебной работы учителя. Тематическое планирование. Конспект и развернутый план проведения урока.

6. Контроль и оценка результатов обучения

Вопросы контроля в учебном процессе. Функции проверки (контролирующая, обучающая, диагностическая, прогностическая, ориентирующая). Принципы проверки (индивидуальная, групповая, фронтальная). Виды проверки (текущая, систематическая, итоговая). Методы проверки (устная, письменно-графическая, практические работы). Средства проверки (вопросы, задачи, задания для выявления знаний, умений и навыков).

Прямая и обратная связь на уроке. Усиление обучающих функций контроля. Контроль с помощью ЭВМ. Контролирующие программы, методика их составления.

Оценка результатов обучения; статистические методы обработки результатов в педагогике.

Оценка эффективности используемых методов и форм обучения.

7. Особенности использования ТСО при обучении основам информатики и вычислительной техники

Роль ТСО в учебном процессе. Виды ТСО: приборы, аппараты, инструменты, печатные и экранные средства. ЭВМ как новое мощное техническое средство обучения; место и границы его применения на уроках информатики. Типы обучающих программ и методика их использования в учебном процессе.

Дидактические возможности различных видов ТСО.

Общие методические подходы к использованию ЭВМ в комплексе с другими средствами обучения. Психолого-педагогические требования к работе на ЭВМ и с ТСО.

8. Организация и содержание внеклассной работы по информатике

Задачи, принципы и содержание внеклассной работы; ее виды и формы. Групповая внеклассная работа, факультативные занятия и их значение. Краткий анализ содержания факультативного курса. Методика проведения факультативных занятий.

Массовая внеклассная работа. Школьные олимпиады, вечера, кружки.

9. Методика изучения тем курса IX класса

Научно-методический анализ и методика изучения тем курса информатики IX клас: а) «Введение», «Алгоритмы. Алгоритмический язык», «Алгоритмы работы с величинами», «Табличные величины. Алгоритмы работы с табличными ве-

личинами», «Вспомогательные алгоритмы», «Этапы решения задач с использованием ЭВМ. Построение алгоритмов для решения задач из курсов математики и физики».

10. Методика изучения тем курса X класса

Научно-методический анализ и методика изучения тем курса информатики X класса: «Устройство ЭВМ», «Знакомство с программированием», «Язык программирования Рапира», «Язык программирования Бейсик», «Роль ЭВМ в современном обществе».

Примерное содержание практических и семинарских занятий

Практическое занятие. Методы обучения и пути их совершенствования.

Цель: отработать умение определять оптимальное сочетание различных методов и приемов обучения.

Содержание: анализ фрагментов уроков, определение использованных методов и приемов; определение методов и приемов, использованных при изучении конкретного вопроса курса ОИВТ; самостоятельное построение хода изучения конкретного вопроса курса ОИВТ с обоснованием выбора используемых методов и приемов.

Практическое занятие. Современный урок, требования к нему и пути его совершенствования.

Цель: получение навыков определения методики проведения урока.

Содержание: анализ посещенного урока, его вид и тип; самостоятельная разработка методики проведения урока с учетом реализации его различных функций.

Практическое занятие. Управление самостоятельной работой школьников и подготовка их к самообразованию.

Цель: закрепить знания по организации эффективной самостоятельной работы учащихся.

Содержание: обсуждение слушателями: а) системы упражнений для организации самостоятельной работы учащихся по данной теме; б) вопросов по тексту учебного пособия по данной теме; в) практических заданий с реализацией на ПЭВМ по данной теме.

Подготовка слушателями: а) упражнений для организации самостоятельной работы учащихся по данной теме; б) вопросов по тексту учебного пособия по данной теме; в) практических заданий с реализацией на ПЭВМ по данной теме; г) списка литературы для учащихся, желающих углубить знания по данной теме.

Семинарское занятие. Психологические аспекты деятельности учителя.

Цель: закрепить полученные знания.

Вопросы для обсуждения: психологические предпосылки эффективности педагогического труда;

педагогическое общение, взаимоотношения учителя и учащихся;

психологические основы педагогического такта и педагогической этики.

Практическое занятие. Психологические аспекты деятельности учителя.

Цель: изучение и анализ педагогической деятельности учителя.

Содержание: проведение анализа педагогиче-

ческой деятельности учителя по следующим компонентам:

- а) познавательная деятельность педагога;
- б) проектировочная деятельность;
- в) конструктивная деятельность;
- г) организаторская деятельность учителя на уроке.

Практическое занятие. Межпредметные связи курса ОИВТ.

Цель: научить определять межпредметные связи на конкретных темах курса ОИВТ.

Содержание: по конкретным темам курса ОИВТ определить связь с другими предметами; изучить литературу по вопросам применения методов и достижений информатики в других науках.

Практическое занятие. Кабинет вычислительной техники в школе.

Цель: познакомить слушателей с работой кабинета ИВТ в школе.

Содержание: обсуждение организационной и содержательной сторон работы кабинета ИВТ. Требования существующих нормативных документов, регламентирующих работу кабинета ИВТ в школе.

Практическое занятие. Организация учебных занятий.

Цель: закрепление знаний по планированию учебных занятий.

Содержание: обсуждение разработанных тематических планов, годового, календарного планов учителя. Самостоятельная подготовка слушателями плана проведения урока по конкретной теме.

Практические занятия. Методика изучения тем курсов ОИВТ IX и X классов.

Цель: закрепление полученных знаний и накопление материалов для преподавательской работы.

Содержание: посещение занятий в школе, СПТУ, ССУЗе, УПК с последующим их анализом.

Разработка отдельных тем курса ОИВТ с обоснованием метода обучения, системы упражнений, межпредметных связей, необходимым дидактическим и раздаточным материалом, применением ТСО, дополнительных заданий по самообразованию (с учетом варианта изучения курса ОИВТ — машинного или безмашинного).

Литература

по разделам I—III учебно-математического плана

О реформе общеобразовательной и профессиональной школы: Сб. документов и материалов. М., 1984.

Бабанский Ю. К. Рациональная организация учебной деятельности. М., 1981.

Бабанский Ю. К. Оптимизация учебно-воспитательного процесса. М., 1982.

Бабанский Ю. К. Методы обучения в современной общеобразовательной школе. М., 1985.

Дидактика средней школы/Под ред. М. Н. Скаткина. М., 1982.

Зотов Ю. Б. Организация современного урока. М., 1984.

Кулько В. А., Цехмистрова Г. Д. Формирование у учащихся умений учиться: Пособие для учителей. М., 1983.

Кухарев В. Н. Педагогическое мастерство учителя: Ч. I—IV. Гомель, 1985.

Лернер И. Я. Дидактические основы методов обучения. М., 1981.

Махмутов М. И. Современный урок. М., 1981.

Махмутов М. И. Организация проблемного обучения в школе. М., 1977.

Отстающие в учении школьники (проблемы психологического развития) / Под ред. З. И. Калмыковой, И. Ю. Кулагиной. М., 1986.

Проблемы общей, возрастной и педагогической психологии/Под ред. В. В. Давыдова. М., 1978.

Психология обучения: Учебное пособие/Под ред. В. В. Давыдова. М., 1978.

Фридман Л. А., Волков К. Н. Психологическая наука — учителю. М., 1985.

Якиманская И. С. Развивающее обучение. М., 1979.

Литература

по общим и частным вопросам методики преподавания курса «Основы информатики и вычислительной техники»

Ершов А. П. Программирование — вторая грамотность // ЭКО. 1982. № 2.

Ершов А. П. Человек и машина. Математика и кибернетика // Знание. 1985. № 4.

Кузнецов А. А., Нурмухамедов Г. М. Компьютер на школьной парте // Физика в школе. 1985. № 4. С. 49—54.

Монахов В. М. Психолого-педагогические проблемы обеспечения компьютерной грамотности // Вопросы психологии. 1985. № 3. С. 14—23.

Монахов В. М. Новый предмет «Основы информатики и вычислительной техники» // Народное образование. 1985. № 4. С. 84—86.

Основы информатики и вычислительной техники: Учебное пособие/Под ред. А. П. Ершова, В. М. Монахова. М., 1985—1986. С. I, II.

Изучение основ информатики и вычислительной техники: Методическое пособие для учителей и преподавателей средних учебных заведений/Под ред. А. П. Ершова, В. М. Монахова. М., 1985—1986. Ч. I, II.

Роберт И. В. Какой должна быть компьютерная обучающая программа // Информатика и образование. 1986. № 2.

Роберт И. В. Наглядные пособия на уроках информатики // Вечерняя средняя школа. 1986. № 4.

Роберт И. В., Кузнецов Э. И. Алгоритмы и алгоритмический язык. Алгоритмы работы с величинами (серия таблиц и методическое руководство по использованию серии таблиц). М., 1985.

Якиманская И. С. Воспитание сенсорной культуры труда. М., 1969.

От школьника до профессора

(опыт работы факультета повышения квалификации)

В сентябре 1985 г. на факультете повышения квалификации преподавателей вузов при Казанском ордена Трудового Красного Знамени химико-технологическом институте им. С. М. Кирова (КХТИ) была открыта специальность «Применение вычислительной техники в учебном процессе и научных исследованиях».

Рабочая программа была составлена на основе типовой программы, рекомендованной Минвузом СССР, содержащей отдельные модули. Проведение занятий по модулям было поручено преподавателям различных кафедр института: прикладной математики, химической кибернетики, автоматизации производственных процессов, процессов и аппаратов химических производств и др. Обучение по этой программе проводилось в течение восьми недель.

В начальный период около половины учебных часов отводилось на изучение алгоритмизации, программирования на языке Бейсик (вариант 3) для микро-ЭВМ «Электроника Д3-28», программирования на языке Фортран IV для ЭВМ ЕС-1033, технологии программирования, ЕСПД. Лекционные, практические и лабораторные занятия по каждому из этих разделов проводились разными преподавателями.

Однако уже в процессе обучения слушателей первого потока стала ясна ошибочность такого подхода, так как параллельное изучение двух языков неподготовленными людьми не давало хорошего результата. Анкетирование слушателей выявило и другие недостатки в преподавании. Например, обучение по близким разделам программы разными, пусть даже высококвалифицированными преподавателями нецелесообразно, так как неизбежны пересечения и неохваченные места на стыках разделов. В соответствии с пожеланиями слушателей ограничили углубленным изучением лишь Бейсика. Постепенно курс оформился как «Технология программирования на языке Бейсик».

В первые два года работы ФПК по вычислительной технике при КХТИ постоянно изменялись требования к преподаванию курса. Помимо преподавателей общеобразовательных, общетехнических и специальных кафедр его прослушали группы профессоров и руководящего состава института, инженеров и аспирантов, преподавателей кафедры иностранных языков. Разумеется, содержание курса менялось в соответствии с контин-

гентом. Неизменным оставалось только то, что все слушатели выполняли выпускную работу, которая представляла собой задачу из предметной области слушателя, решаемую с помощью ЭВМ. Программа должна была состоять не менее чем из 50 операторов. Защита выпускной работы проводилась в присутствии комиссии. На какие только темы не писали слушатели свои выпускные работы! В процессе их выполнения стало ясно, что возможности Бейсика сковывают инициативу слушателей. Особо это выявилось при создании обучающих и контролирующих программ преподавателями кафедры иностранных языков, неорганической и аналитической химии.

В связи с этим было решено разработать новый курс по технологии программирования на языке, допускающем обработку символьной информации. В качестве этого языка была выбрана версия Бейсика + (М), разработанная в Казанском государственном университете. Этот курс и читается в настоящее время. Обучение проходит в течение 17 недель (36 ч лекций, 30 ч практических и 32 ч лабораторных занятий). Кроме этого, читаются и подкрепляются лабораторными занятиями на микро-ЭВМ «Электроника Д3-28» другие курсы: «Принципы устройства и работы ЭВМ», «Статистические и численные методы решения задач на ЭВМ», «Проблемы охраны окружающей среды», «Методы автоматического и оптимального проектирования», «Гибкие автоматизированные производства». Лабораторные работы по курсу АОС слушатели выполняют на ЭВМ ЕС-1045.

На ФПК работают опытные преподаватели. Много времени и энергии они отдают распространению компьютерных знаний среди инженеров предприятий города, а также среди подрастающего поколения. Частые гости в вычислительной лаборатории ФПК — школьники старших классов. В летнее время здесь проводятся занятия летней химической школы «Орбиталь» — подразделения Малого химико-технологического института при КХТИ, коллективу которого присуждена премия Ленинского комсомола 1987 г. в области педагогической деятельности.

Ближайшими задачами работы ФПК являются укрепление технической базы и углубление специализации обучения.

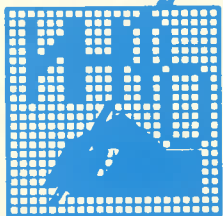
Л. ДМИТРИЕВА, доцент
Г. МАТУШАНСКИЙ, декан ФПК

119

В городе Черновцы 6—9 октября 1987 г. проходил семинар «Теоретические и прикладные проблемы компьютеризации обучения».

Участники семинара познакомились с работами, которые ведутся в стране в этом направлении.

Организационный комитет стремился предоставить всем участникам возможность высказать свою позицию.



Самодельный модем — нелепость или необходимость?

120 Крайне редки стали упоминания на страницах прессы о связи ЭВМ по телефону. В конце XX в. она не более удивительна, чем водопровод или автомобиль. Казалось бы, отсюда можно сделать оптимистический вывод о скором появлении модемов в школьных классах. Однако при внимательном рассмотрении оптимизм исчезает.

Во-первых, отечественная промышленность не выпускает модемы в достаточном количестве, так что острая их нехватка ощущается в сферах, гораздо более важных, чем просвещение.

Во-вторых, из личного опыта каждому известно качество работы нашей телефонной сети.

Этого достаточно, чтобы понять, что широкое внедрение модемов в школьную практику следует отнести «на перспективу до 2000 г.». А кто не хочет ждать, тот может попытаться связать машины по телефону «собственноручно».

Возможно ли это?

Напомним несколько фактов:

1. У каждой ЭВМ, при всем разнообразии конструкций, есть единая стандартная возможность подключения к магнитофону.

2. Магнитофонный код у разных ЭВМ различен, но физически сигнал одинаков — это прямоугольные импульсы. Значения битов 0 и 1, а также вся прочая информация кодируются исключительно длительностью импульсов.

3. Программист, знающий ассемблер данной машины и знакомый с ее конструкцией, без особого труда может написать программу, формирующую на магнитофонном выходе импульсы любой длительности и в любой последовательности.

А теперь вспомним, что физическая сущность телефонной связи состоит главным образом в том, что переменное сопротивление микрофона модулирует постоянный ток, посылаемый телефонной станцией. Легко изготовить устройство размером с пачку сигарет, которое будет делать нечто подобное, но подчиняясь сигналам, поступающим с магнитофонного выхода ЭВМ. В зависимости от высокого или низкого уровня сигнала модулятор (можно смело назвать его так) будет изменять собственное сопротивление постоянному току телефонной линии, тем самым формируя в ней

прямоугольные импульсы. (Разумеется, ряд деталей, в том числе и существенных, опущен, чтобы не затемнять основную идею.)

Правда, после путешествия по телефонной сети прямоугольный импульс неузнаваемо изменится. Но с помощью элементарного устройства, которое способен создать даже юный радиолюбитель, может быть восстановлен исходный сигнал (естественно это устройство назвать демодулятором). После этого сигнал остается лишь подать на магнитофонный вход другой ЭВМ — связь установлена: одна машина «думает», что пишет на магнитофон, другая — что читает, а на самом деле они обмениваются информацией непосредственно друг с другом. И это достигается без всяких переделок машин — требуются лишь пара коробочек, включаемых между магнитофонным выходом ЭВМ и телефонной линией.

Конечно, на пути реализации этой идеи немало сложностей. Вот некоторые из них: выбор конструкции модема (надежно работающей и не содержащей дефицитных элементов); разработка физического и логического стандарта; разработка программного обеспечения; борьба с помехами телефонной сети.

Преодолимы ли они? Конечно, да. Кто будет этим заниматься? На этот вопрос, к сожалению, нельзя дать столь же четкий ответ. Зато легко указать тех, кто, по-видимому, будет этому препятствовать. Например, Министерство связи — запретило всегда спокойнее. Или потенциальные разработчики модемов. В самом деле, если у института есть в плане многотысячный заказ и вдруг обнаруживается, что устройство примерно той же полезности спаял юный радиолюбитель, институт это не обрадует.

Впрочем, всех запретов, препятствий, затруднений, препон не счесть. Но какие же выводы в конце концов можно сделать?

«Нелепая необходимость» — так, пожалуй, следует сказать о самодельном модеме. Одно-единственное предпринятие в рамках производства ширпотреба могло бы засыпать школы модемами. И один программист за год — максимум! — мог бы создать необходимое программное обеспече-

ние. К сожалению, на бездействие солидных организаций и высокопоставленных лиц можно рассчитывать, а на содействие — едва ли; и то, что требуется, легче всего сделать самостоятельно.

В данном случае это вполне реально: можно самому изготовить эти коробочки и написать про-

граммы для диалога двух машин. Автор настоящей заметки уже работает в этом направлении и надеется после ее публикации найти единомышленников.

А. ИГЛИЦКИЙ

«Хотелось бы обсудить...»

В связи с внедрением компьютеризации в учебный процесс хотел бы поделиться одним соображением.

Сейчас оперативность обмена опытом между пользователями, живущими в разных городах, минимальна. Можно, конечно, съездить в другой город, но в учебное время это трудно. Можно попросить прислать текст программы, но что, если она в машинных кодах или имеет длину, допустим, 1000 строк?

Возможна и такая ситуация. В вашем журнале мы прочли, предположим, что некто создал хорошую программу. Пишем автору письма. Он высылает программу. Мы ее получаем, но поздно! Соответствующий раздел уже пройден. Чем дальше от редакции живет пользователь, тем больше этот разрыв между запросом и получением ответа. Посылать программу телеграфом? Слишком дорого!

Между тем способ оперативной связи есть. Это обычный телефон. Однако он не приспособлен для передачи двоичной информации: узка полоса пропускаемых частот, значительны частотные искажения. Человеческое ухо справляется с этими недостатками, но не компьютер.

Тем не менее передачу по телефонной линии двоичной информации можно организовать: для этого достаточно снизить скорость передачи до 300 бод и, кроме того, передавать сигнал в телефонную сеть, минуя микрофон телефонной трубки.

Передачу и прием программы желательно организовать через кассетный магнитофон, который подключается к телефону через некоторое устройство.

К сожалению, к телефонной сети любую аппаратуру без разрешения Министерства связи СССР подключать запрещено.

Поэтому хотелось бы обсудить на страницах журнала возможность создания такого устройства для магнитофона и получения санкции Министерства связи СССР на его использование.

При реализации этого предложения мы получим общесоюзную (можно позвонить в любое место нашей страны) общедоступную (телефоны есть не только в учреждениях, но и в личном пользовании) оперативную (не чета почте) компьютерную связь.

Реализация последовательного интерфейса для телефонной сети, очевидно, несложна.

В заключение можно предложить организовать центральный банк информации по обеспечению учебного процесса, который мог бы собирать программы и рецензировать их на предмет массового тиражирования. В этот банк можно было бы обращаться за любой хранящейся там информацией, в любое удобное время. Вполне очевидно, что программы могут отыскиваться и пересылаться компьютером, установленным в этом банке, без участия человека.

Можно привести еще один аргумент в пользу создания телефонной компьютерной связи.

Пересылкой информации занимается не только пользователь компьютера, но и любая статистическая служба, учреждения, предприятия и т. д. При внедрении телефонной компьютерной связи многие официальные бумаги можно будет пересылать в виде программ для компьютера, который их отпечатает в необходимой для делопроизводства форме. Для равномерной загрузки телефонной линии их можно передавать ночью.

А. ЯНГУЛОВ
г. Абакан

А. БАРИНОВ, А. СЕМИБРАТОВ
Госпрофобр РСФСР

Семинар обобщает опыт

122

При анализе проблем, связанных с использованием компьютеров в учебном процессе, обычно в качестве основной выделяется проблема дефицита ЭВМ. По нашему мнению, важно другое. Сейчас создается ситуация, когда поступление средств вычислительной техники опережает создание для них педагогических программных средств (ППС). В 1990 г. каждое второе профтехучилище будет иметь комплект учебной вычислительной техники, а ППС, принятых экспертным советом Госпрофобра СССР, насчитывается всего лишь несколько десятков — это для более чем 1500 профессий. Остро встает вопрос об эффективном использовании средств ЭВТ.

Объем теоретических знаний, практических умений и навыков работающих увеличивается в связи с усложнением оборудования, внедрением новых технологий. Изменяется само содержание труда современного рабочего, происходит его интеллектуализация. А увеличение сроков подготовки рабочих недопустимо. В настоящее время значительная часть станков с ЧПУ и обрабатывающих центров в стране простаивает из-за нехватки квалифицированных кадров для их обслуживания. Выход — в интенсификации учебного процесса.

Средство новой информационной технологии, которое позволит интенсифицировать процесс профессиональной подготовки за счет активизации познавательной деятельности и усиления индивидуализации обучения, — компьютер. Педагогические программные средства, в свою очередь, являясь содержательной основой компьютерного обучения.

Понимая важность этой проблемы, Госпрофобр РСФСР принял энергичные меры по разработке ППС. Так, в настоящее время на базе опорных профтехучилищ Российской Федерации созданы 17 центров, разрабатывающих педагогические программные средства. Основой их деятельности является координационный план создания ППС на двенадцатую пятилетку.

В соответствии с этим планом ведется разработка ППС по специальным и общетехническим курсам для 14 групп профессий. Среди них «Электротехника», «Автоматизация производства

на основе ЭВТ», курсы специальной технологии по ряду профессий и, в первую очередь, по приоритетным направлениям научно-технического прогресса: электронизации и автоматизации народного хозяйства, новым материалам и технологии их производства, биотехнологии.

Некоторые научные и методические проблемы создания и применения ППС в учебно-воспитательном процессе ПТУ решаются совместно с вузами. Опыт работы 22 зональных центров рассмотрен на совместной коллегии Минвуза РСФСР и Госпрофобра РСФСР в июле 1987 г., где были конкретизированы направления работ, уточнена специализация зональных центров по созданию и внедрению ППС. Принято решение о создании при вузах отраслевых лабораторий Госпрофобра РСФСР по данной проблематике. Одна из таких лабораторий уже создана при Ленинградском институте авиационного приборостроения.

Все программные средства, разрабатываемые в системе профтехобразования, рассматриваются экспертной группой, созданной на базе Центральной научно-исследовательской лаборатории проблем подготовки рабочих широкого профиля (ЦНИЛПРШП) Госпрофобра РСФСР. Экспертной группой проводится не только оценка поступающих ППС, но и ведется консультационная работа с авторами и организациями — разработчиками ППС; выдаются конкретные рекомендации по доработке содержания, сценариев и методики применения программного средства; проводится анализ недостатков представляемых материалов. В частности, было установлено, что авторами при разработке ППС в основном производится анализ недостатков представляемых материалов. В частности, было установлено, что авторами при разработке ППС в основном производится анализ недостатков представляемых материалов. В частности, было установлено, что авторами при разработке ППС в основном производится анализ недостатков представляемых материалов. В частности, было установлено, что авторами при разработке ППС в основном производится анализ недостатков представляемых материалов.

Для подведения первых итогов и в целях обмена опытом разработки и применения ППС в Омске

с 23 по 25 ноября 1987 г. проходил I Всероссийский семинар-практикум по проблемам применения ЭВТ в учебном процессе средних профтехучилищ, в ходе которого решались следующие задачи:

обобщение опыта совместной работы вузов, управления и учебных заведений профтехобразования в разработке педагогических программных средств (ППС) и применению ЭВМ в учебном процессе;

определение научных и методических принципов, разработки и применение ППС;

определение форм организации учебной деятельности с использованием ЭВТ в процессе профессиональной подготовки учащихся СПТУ.

В работе семинара-практикума участвовали 120 человек: преподаватели и мастера производственного обучения большинства областных управлений профтехобразования и зональных центров, на базе которых осуществляется внедрение ЭВТ в учебный процесс СПТУ; ведущие ученые и специалисты ВНИИ профтехобразования, НИИ профтехпедагогики АПН СССР, ЦНИЛПРШП, педагогических и политехнических институтов, а также представители Госпрофобров СССР, РСФСР и некоторых союзных республик.

В ходе семинара были проведены пленарные и секционные заседания по следующим направлениям:

теория и практика использования ЭВТ в профессиональной подготовке рабочих кадров;

организация учебной деятельности учащихся СПТУ в кабинете вычислительной техники;

программное обеспечение средств ВТ, применяемых в СПТУ.

На семинаре работали три секции, организованные по принципу разработки и использования ППС для конкретных типов ЭВМ, а именно: КУВТ-86, ДВК-2М, КУВТ «Ямаха». Работа секций строилась в форме живого диалога между докладчиком и слушателями.

Выступления сопровождалась демонстрацией ППС. Достаточное количество рабочих мест за компьютерами позволило слушателям поработать с представленными ППС. Выказывались замечания, оценки, конкретные предложения по доработке программ, по методике их использования в учебном процессе. Такая форма обслуживания оказалась весьма полезной и для слушателей, и для докладчиков. По сути, участники семинара получили опыт проведения компьютеризованных уроков.

Нынешний семинар собрал много энтузиастов компьютерного обучения. Гостеприимный Дом техники Омского областного управления профтехобразования, где проходил семинар, был открыт с раннего утра и до поздней ночи для дискуссий и обмена разработками в области программного обеспечения.

Важной частью семинара являлась работа экспертной группы Госпрофобра РСФСР по приемке ППС, представленных в ходе работы секций для различных видов техники. Из 42 рассмотренных поурочных ППС 18 были приняты, по восьми авторам были даны конкретные замечания и предложения по доработке, а остальные ППС признаны неудовлетворительными. Оперативность, непосредственный контакт с разработчиками, учет мнения аудитории — достоинства проведения эксперти-

зы параллельно с работой семинара. Такая форма работы экспертной группы оказалась высокоэффективной, полезность ее применения в будущем не вызывает сомнений.

Интересные доклады представили сотрудники ВНИИ профтехобразования, ЦНИЛПРШП, СИПИ, НИИ профтехпедагогики АПН СССР, представителей региональных центров Украины и Азербайджана, Эстонии, а также Куйбышевского, Ростовского, Ленинградского, Новосибирского зональных центров.

Группа ученых НИИ профтехпедагогики АПН СССР под руководством В. А. Беловина и Д. М. Шакировой сделала сообщения о принципах построения моделирующих ППС и изложила основные положения проектирования программно-методического обеспечения для профессиональной школы. Предложенные ими подходы реализованы при разработке ППС по 11 темам различных учебных предметов.

С интересом было встречено выступление сотрудников Свердловского инженерно-педагогического института под руководством В. Н. Ларионова. Им был продемонстрирован целый комплекс ППС для КУВТ-86 и «Ямахи», направленных на решение различного рода задач по курсам спецтехнологии и предметам производственного обучения профессий машиностроительного профиля.

По нашему мнению, актуальность этих работ в том, что программы-тренажеры (а именно они представлены группой) — наиболее эффективные из используемых ППС. Так, в одной из программ моделируется работа прокатного стана, выполняется расчет энергии, расходуемой на прокатку в каждом проходе, и суммарный расход энергии за все проходы.

Пакет программ, моделирующих процесс производственного обучения электромонтажников, продемонстрировали сотрудники Куйбышевского политехнического института. Он предназначается для ДВК-2М с платами графики (КГД). Отличительной чертой этого комплекта программ является интегрированное рассмотрение урока производственного обучения, когда из отведенных 6 ч обучения на компьютере занимает 1 ч и 5 ч — работа в электромонтажной мастерской. Это повышает педагогическую эффективность применения микро-ЭВМ и улучшает качество обучения.

Грамотное использование тех небогатых технических возможностей, которые имеются у «Электроники БК-0010», и творческое сотрудничество педагогов и высококвалифицированных программистов может дать неплохие результаты. Примером этого являются комплекты ППС, разработанные в СПТУ-7 Тулы и СПТУ-77 Свердловска по спецтехнологии операторов станков с ЧПУ, представленные к награждению бронзовой медалью ВДНХ. В них максимально использованы графические, динамические и цветовые возможности Фокала.

Интересный пример ППС для интенсивного обучения ЭВМ. Разработанная МАИ по заказу Госпрофобра РСФСР, эта программа учитывает при выдаче каждого задания время выполнения предыдущего и допущенные ошибки, ведет протокол обучения.

Своими подходами к проектированию ППС поделился руководитель регионального центра при

Школьник, здоровье,

компьютер...

Как функционируют различные органы и системы человеческого организма под влиянием электронно-вычислительной техники? Нет сомнений, что отвечать на поставленный вопрос — ученым. «Приблизительный, неаргументированный ответ на уровне обывательских разговоров («а вы слышали, от компьютеров бывает...») — нас не устраивает.

В лаборатории медико-биологических проблем НИИ школ Минпроса РСФСР претворяется в жизнь программа комплексного исследования здоровья школьников, работающих с ЭВМ. Программа рассчитана до 1991 г.



учебному предмету всегда ставятся задачи частичного или полного пересмотра содержания и структуры программы обучения по данному предмету, создания новых учебных и методических пособий, разработки дидактических средств обучения. Создание отдельных автоматизированных обучающих программ вне комплексного методического обеспечения не может решать задачи повышения эффективности учебного процесса.

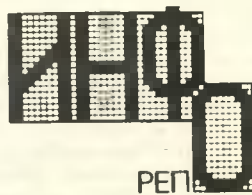
Преимуществами такой технологии разработки ППС являются обеспечение организационного и методологического единства разработок; возможность «распараллеливания» работ; удобство программирования и однозначность толкования сценария. Этот доклад был особенно ценен для практических работников, так как содержал конкретные рекомендации.

Заведующая лабораторией ВНИИ профтехобразования Л. Б. Лебедева рассказала о комплексном подходе к созданию и применению ППС.

Это важно потому, что наряду с задачей создания программного продукта по какому-либо

учебному предмету всегда ставятся задачи частичного или полного пересмотра содержания и структуры программы обучения по данному предмету, создания новых учебных и методических пособий, разработки дидактических средств обучения. Создание отдельных автоматизированных обучающих программ вне комплексного методического обеспечения не может решать задачи повышения эффективности учебного процесса.

Такие семинары будут проводиться ежегодно.



РЕПЕРТАЖ
НОМЕРА

К настоящему времени разработана методика исследования, установлены испытательные стенды отечественного производства, проведены пробные исследования кардиореспираторной системы школьников (фото 1)*.

Уроки информатики ведет молодой преподава-

* См. в этом номере: И. Антипов, С. Колобов, О. Ермолаев, В. Анисимов. О методике психологического исследования адаптации школьников к учебной нагрузке при работе на компьютере.



1. Сотрудница лаборатории Е. Блинова — одна из активных участников эксперимента. Данные по морфофункциональным особенностям организма школьников фиксируются до, во время и после урока. Полианализатор ПА5-01 обладает широкими возможностями.

2. Урок ведет А. Луканкин. Совместно с С. Колобовым они разработали «Тетрадь по информатике», которую вы видите в руках учащихся. Это ценное пособие, позволяющее тренировать полученные навыки, собирать сведения об усвоении учебного материала.



тель А. Луканкин. В поле его научных интересов — педагогические проблемы компьютеризации обучения. Школьники с пониманием относятся к стоящим перед ними задачам (фото 2); вдумчивости и дисциплинированности содействует творческая атмосфера, царящая здесь.

По результатам исследования планируется написать монографию о медико-психологических аспектах адаптации школьников при работе с ЭВТ. Огонек, душу вкладывают в дело все, но «центром притяжения» группы, сплачивающим и вдохновляющим участников эксперимента, является С. Колобов (фото 3).

3. Никтоскоп-01 — так называется прибор, находящийся на первом плане. Он позволяет изучить возможности и особенности зрительного анализатора. С. Колобов (стоит) большое значение придает исследованию влияния работы с компьютерами на детское зрение. Прогнозы, опирающиеся на первые результаты эксперимента, вполне оптимистичные.

Почему так важны для нас, педагогов, родителей, журналистов, пишущих на «компьютерную тему», научно обоснованные, подкрепленные фактами, гигиенические нормы работы с ЭВМ, практические рекомендации, психологически и биологически оправданные? Вряд ли стоит пространно пояснять: здоровье наших детей, всего поколения, которому жить в XXI веке, для нас превыше всего.

Т. ДРАГНЫШ

Новая книга для учителя

Уже сейчас, на начальном этапе применения компьютеров в среднем образовании, накоплен некоторый опыт. Отражением его явилась вышедшая в издательстве «Просвещение» книга для учителя «Вопросы компьютеризации учебного процесса»*. Это сборник статей, подготовленных учителями школ и учебно-производственных комбинатов Москвы.

Составитель, выделив три группы методических проблем, возникающих в процессе компьютеризации учебно-воспитательного процесса, распределил в соответствии с ними материал сборника по следующим трем разделам:

I. Использование вычислительной техники в процессе преподавания курса «Основы информатики и вычислительной техники».

II. Использование вычислительной техники в преподавании физики.

III. Профессиональная подготовка учащихся по основам программирования.

Естественно, такое распределение материала не позволяет охватить весь круг педагогических проблем, связанных с компьютеризацией обучения, но в рамках оговоренной тематики представляется вполне оправданным и позволяет читателю получить достаточно полное представление о содержа-

нии и методах обучения, формах работы с вычислительной техникой в школах и учебно-производственных комбинатах Москвы за последнее пятилетие.

Работа, начатая редакцией математики издательства «Просвещение» по освещению различных аспектов применения вычислительной техники в обучении, несомненно, заслуживает одобрения. Хотелось бы, чтобы в ближайшие годы каждый педагог, применяющий вычислительную технику в своей деятельности, получил возможность ознакомиться с опытом использования компьютеров не только в курсе основ информатики и вычислительной техники, но и в курсе математики. Кроме того, принимая во внимание резко возросший интерес школьников и учителей к изучению вопросов разработки и оптимизации алгоритмов, численных методов «машинной» математики, языков программирования высокого уровня (Паскаль, Си, Фокал, Бейсик, Лого и др.) и практически полное отсутствие доступной по уровню изложения учебной литературы, представляется полезным издание для этой категории читателей небольших по объему брошюр в рамках какой-либо серии, например: «Математика, информатика и вычислительная техника для школьников».

Е. ЛОДАТКО

Канд. пед. наук,

доцент Славянского пединститута,

учитель информатики СШ № 5 г. Славянска

* Вопросы компьютеризации учебного процесса/Сост. Н. Д. Утринович. М.: Просвещение, 1987.

О некоторых книгах, выпущенных в 1987 г.

1. Клейман Г. М. Школы будущего. Компьютеры в процессе обучения: Пер. с англ.— М.: Радио и связь, 1987.

Цель книги — демонстрация различных способов использования персональных компьютеров в системе образования. Рассматриваются различные области применения компьютеров в школе: обработка текстов, игры, разработка программ и т. п. Значительное место уделяется описанию возможностей компьютера как инст-

румента при изучении правописания, математики, иностранного языка, рисования и т. д.

Для всех, кто интересуется применением компьютеров в школе.

2. *Коснёвски Ч.* Занимательная математика и персональный компьютер: Пер. с англ.: М.: Мир, 1987.

В книге в увлекательной форме изложены методы изучения разделов математики с помощью персональных компьютеров. Кратко даются теоретические сведения, предлагаются занимательные задачи, для которых приведены законченные программы с подробными комментариями. Для всех, кто изучает математику с помощью компьютера и осваивает языки программирования.

3. *Паскалев Ж.* Первые шаги в вычислительной технике: Пер. с болг. М.: Радио и связь, 1987.

В книге раскрывается понятие алгоритма, даны основы построения компьютеров, показана роль программного обеспечения в работе ЭВМ, приведен ряд программ.

Для читателей, интересующихся устройством компьютера.

4. Персональные компьютеры: Информатика для всех. М.: Наука, 1987.

Сборник носит обзорный характер и содержит начальные сведения о персональном компьютере. Описываются устройство компьютера, основные приемы программирования, методы решения некоторых прикладных задач.

5. *Хирн Д., Бейкер М.* Микрокомпьютерная графика: Пер. с англ.— М.: Мир, 1987.

В книге приведены алгоритмы и программы машинной графики, ориентированные на использование персональных компьютеров и мини-ЭВМ. Даны рабочие программы на языке Бейсик для создания плоских и пространственных статистических и динамических изображений. Рассмотрены возможности применения компьютеров в управлении, экономике, обучении и в быту. Для читателей, интересующихся методами машинной графики.

Составил Д. ВОЛКОВ

Уважаемые читатели!

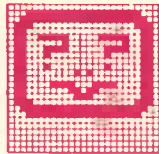
Тем, кто хочет выступить на страницах нашего журнала, поделиться опытом преподавания курса «Основы информатики и вычислительной техники», организации занятий, внеклассной работы, технического творчества школьников в кружках электронного конструирования, сообщаем: ваша статья, обзор, информация должны быть отпечатаны на пишущей машинке в двух экземплярах через два интервала на стандартных листах писчей бумаги. Рукописи, представленные в виде принтерных распечаток, принимаются в работу, если выполнены на белой бумаге обычного формата, размер шрифта близок к стандартному (не мелок) и расстояние между строками не менее двух интервалов. На последней странице укажите фамилию, имя, отчество автора, домашний адрес, место работы, должность, количество детей в семье.

При подготовке рукописи необходимо уточнить используемые термины, сокращения, источники цитат, название предприятий, учреждений, географические названия.

Прилагаемые эскизы, схемы, таблицы должны быть выполнены четко и аккуратно (можно шариковой ручкой на листах плотной бумаги) и снабжены необходимыми пояснениями.

Фотографии, отпечатанные на глянцевой бумаге размером не меньше 9×12 см, должны быть без повреждений, изломов, царапин. Пояснительные надписи к ним выполняются на оборотной стороне только мягким карандашом.

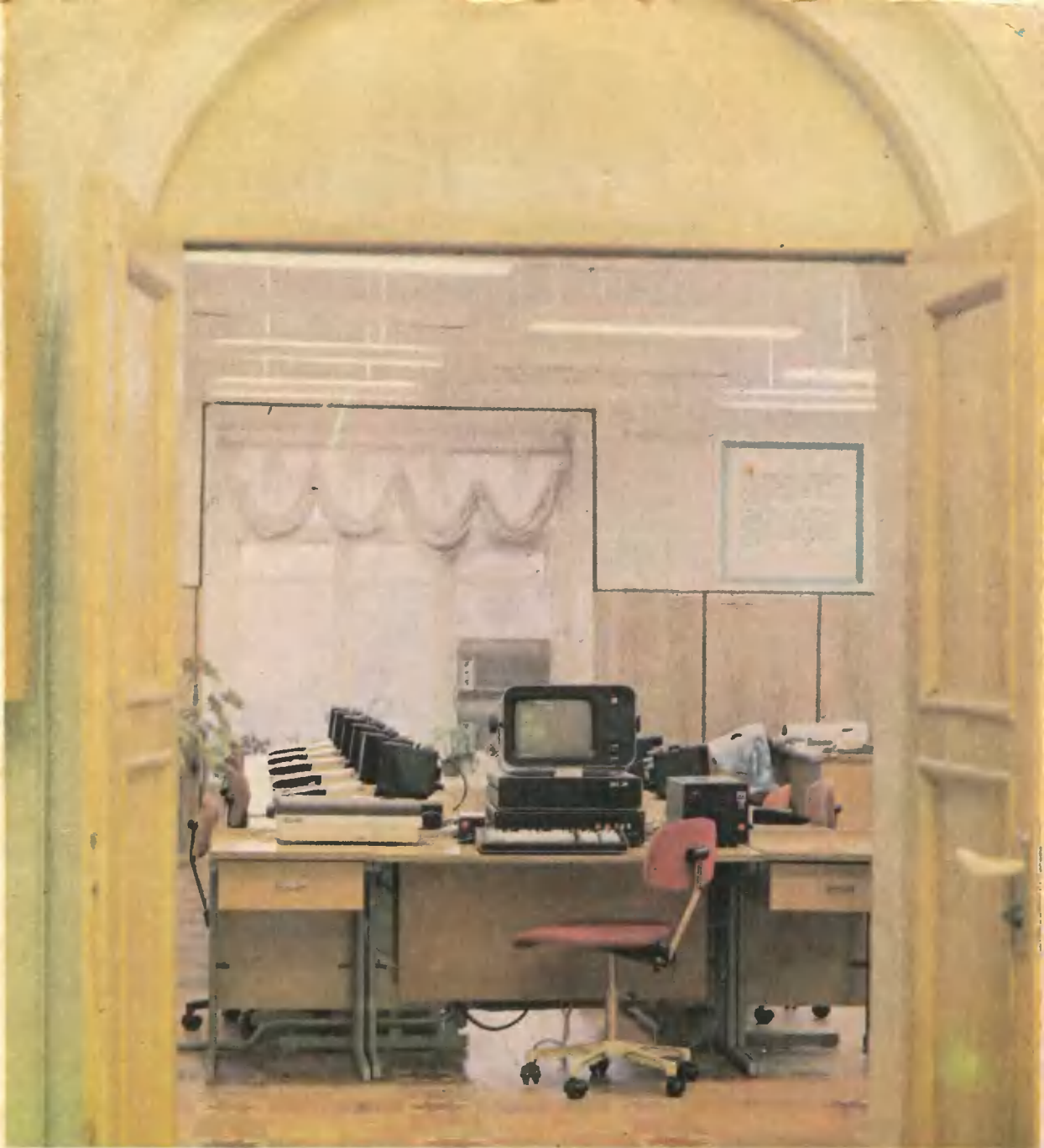
Ждем от вас интересных материалов!



У НАС
В ПРОДАЖЕ
ВСЕГДА:
СВЕЖИЕ
ПРОГРАММНЫЕ
ПРОДУКТЫ
МАГАЗИН "ЭЛЕКТРОНИКА"

ПРЕКРАТИ
СВОИ ГЛУПЫЕ
ИГРЫ!...







Политехнический музей — это не только история техники, но и ее сегодняшний день. В этом зале установлен КУВТ-86, на котором проводятся занятия учебных групп.

На четвертой странице обложки — устройство ввода информации с перфоленты выпуска 1961 г.

Цена 60 коп.
70423

ИИФ 
2 '88 

**ИНФОРМАТИКА
И ОБРАЗОВАНИЕ**

