

Бн/86

3455

**ЭВМ В ОБРАЗОВАНИИ**  
**программное обеспечение**

**сборник научных трудов**  
(МЕЖВУЗОВСКИЙ)

Министерство народного образования Латвийской ССР

Латвийский ордена Трудового Красного Знамени  
государственный университет имени Петра Стучки

Вычислительный центр

Лаборатория проблем школьной и вузовской информатики

**ЭВМ В ОБРАЗОВАНИИ  
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

**СБОРНИК НАУЧНЫХ ТРУДОВ  
(межвузовский)**

Латвийский государственный университет им. П. Стучки  
Рига 1988

**Э В М В О Б Р А З О В А Н И И**  
**П Р О Г Р А М М Н О Е О Б Е С П Е Ч Е Н И Е**

ЭВМ в образовании. Программное обеспечение: Сборник научных трудов (междуазовский) / Под ред. Н.Н.Устинова. Рига: ЛГУ им. П.Стучки. 1988. 182 с.

Предлагаемый сборник содержит 17 статей, посвященных научным и методическим проблемам адаптации, разработки и стандартизации педагогических программных средств (ППС) для применения при изучении информатики и вычислительной техники, предметов естественно-научного цикла, а также разработке некоторых инструментальных средств создания ППС.

Сборник научных трудов предназначен для научных работников, занимающихся исследованиями и разработкой ППС, методическими вопросами применения ППС, а также преподавателям, аспирантам и студентам, применяющим персональные ЭВМ в качестве технического средства обучения или в качестве инструмента в своей профессиональной или учебной деятельности.

Рис. 42. табл. 22. библиогр. 93 назв.

**РЕДКОЛЛЕГИЯ;**

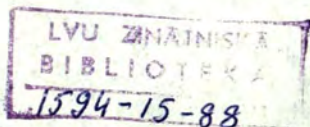
М.В.Витиньш, В.Л.Цилевич,  
А.С.Корхонена, С.И.Павлов,  
Н.Н.Устинов (отв.ред.)

30502-078y

9 ----- 22.88.2405000000

М 812(11)-88

(С) Латвийский  
государственный  
университет  
им.П.Стучки, 1988



## В В Е Д Е Н И Е

В настоящее время активно развиваются работы, связанные с компьютеризацией образования. Актуальными направлениями исследований являются: разработка научно-методической базы нового школьного общеобразовательного курса "Основы информатики и вычислительной техники" ("ОИиВТ"); определение состава и содержания программно-методического обеспечения для изучения информатики и предметов естественно-научного цикла (математики, физики, химии, астрономии и др.); разработка педагогических программных средств (ППС); создание инструментального программного обеспечения для разработки ППС; рассмотрение вопросов, связанных со стандартизацией программных средств.

Научные исследования по компьютеризации образования и разработка перспективных ППС координируются Академией педагогических наук СССР в рамках комплексной программы "ЭВМ в школе". В Латвийской ССР разрабатывается Республиканская целевая комплексная программа "Электронизация народного хозяйства".

Настоящий сборник вносит определенный вклад в разработку проблем компьютеризации образования. Основу сборника составляют исследования, проведенные в 1985-1987 г.г. в лаборатории проблем школьной и вузовской информатики Вычислительного центра при ЛГУ им. П. Стучки, научная тематика которой координируется названными программами исследований.

Статьи сборника разделены на три раздела (см. также содержание). В разделе "Базовое и инструментальное программное обеспечение" проводится сравнение [ 1 ] возможностей вывода информации с использованием средств основного языка программирования (БЕЙСИК, ФОКАЛ) соответствующих персональных ЭВМ (ПЭВМ), применяемых в средних учебных заведениях; обсуждаются требования [ 5 ] к школьным ПЭВМ и комплектам учебной вычислительной техники (КУВТ), сформулированные на основе опыта работы с большинством КУВТ, применяемых в настоящее время в обучении. Рассматриваются некоторые возможности [ 4 ] применения в учебном процессе системного и прикладного програм-

много обеспечения ПЗВМ "ДВК", входящей в КУВТ-86 - наиболее распространенного комплекта в школах Латвийской ССР. Представлены оригинальные результаты разработок: кодовые таблицы символов [ 2 ], включающие национальный (латышский) алфавит, а также аппаратно-программные инструментальные средства [ 3 ] для разработки программного обеспечения для микроЭВМ "Электроника БК-0010".

В разделе "Программно-методическое обеспечение обучения информатике" излагается подход к обучению информатике студентов и школьников [ 7 ], развитый авторами конкурсного учебника по курсу "ОИИВТ". Рассматривается опыт организации и отбора материала для учебных телевизионных передач [ 6 ] (во время которых проводилась передача программ по звуковому каналу) по курсу "ОИИВТ", а также для кружковых занятий [ 8 ] со школьниками. Представляются некоторые разработанные ими программы. Обсуждается опыт разработки [ 9 ] конкретных программ для применения в обучении информатике. Подробно рассмотрены [ 10 ] вопросы организации процесса разработки ППС с участием учителей, а также внедрение ППС в школах Шотландии.

В разделе "Программное обеспечение предметов естественно-научного цикла" рассматриваются некоторые разработанные ППС по вычислительной математике [ 12,13 ], химии [ 14 ], астрономии [ 15 ], физике [ 16 ], для синтеза звука [ 17 ], обсуждаются принципы разработки. Проанализированы [ 11 ] свойства ППС на базе последовательного применения метода тематического моделирования.

Широкий спектр рассматриваемых проблем во многом отражает настоящий этап развития компьютеризации образования, возможности средств учебной вычислительной техники, складывающуюся специализацию в проведении научных исследований, связанных с разработкой ППС. С этой точки зрения предлагаемый сборник может представить интерес не только для научных сотрудников, но и для аспирантов и студентов, а также преподавателей педагогических вузов и средних учебных заведений, активно занимающихся внедрением вычислительной техники в учебный процесс.

УДК 681.322.068

Л. П. Климанс, Л. В. Озолиня, А. А. Прикулис, И. А. Шкерстена  
Вычислительный центр при ЛГУ им. П. Стучки

ОСОБЕННОСТИ ВЫВОДА ИНФОРМАЦИИ ДЛЯ МИКРОЭВМ "ACORN",  
"YAMANA", "КОРВЕТ", "АГАТ", "ЭЛЕКТРОНИКА ВК-0010"

При разработке программ для микрокомпьютеров (в т. ч. для обучения программированию) широко используется вывод на экран текстовой и графической информации. Применение компьютерной графики не только позволяет наглядно представлять результаты работы программы, но и делает процесс обучения более увлекательным.

В настоящее время средние учебные заведения располагают микроЭВМ различных типов, каждый из которых имеет особенности вывода информации на экран. Материал, систематизированный в статье, дает возможность ориентироваться в различных организациях вывода текстовой и графической информации, облегчает адаптацию программ (в т. ч. разработанных на языках БЕЙСИК и ФОКАЛ) с микроЭВМ одного типа на другой.

#### 1. Графический и текстовый режимы экрана

Графическая информация выводится на экран, который можно представить в виде прямоугольника, состоящего из точек. Для разных типов микрокомпьютеров количество точек по горизонтали и по вертикали экрана различно, для одного типа микрокомпьютера может быть несколько режимов экрана с различным разрешением.

Каждый микрокомпьютер имеет определенное количество цветов, в которые может быть окрашена любая точка экрана. Для определенного режима экрана доступный набор цветов меньше, чем полный набор цветов.

Графическая информация на экране является отображением данных из экранной памяти. Структура данных экранной памяти различна для разных типов микроЭВМ. Операторы вывода графики формируют данные экранной памяти, отображаемые на экране в определенном виде, например, точки или прямой. Программируя на языках высокого уровня, таких, как БЕЙСИК, ФОКАЛ, пользователь может не интересоваться структурой данных экранной памяти.

С точки зрения вывода текстовой информации, экран также можно представить в виде прямоугольника, разделенного на блоки. Каждый блок экрана состоит из определенного количества точек (в большинстве случаев 8x8 точек). В одном блоке (позиции) можно отобразить один символ. Вывод символа означает закрашивание точек определенного блока (позиции) соответственно графическому изображению символа. На рис. 1 показаны закрашенные точки (изображены символом "#") в позиции для вывода буквы "к".

Операционная система, организуя вывод символа на экран, кроме значения кода символа, использует таблицу, в которой хранятся числовые значения, задающие графическое изображение символов. На рис. 1 показаны эти значения для буквы "к" (в двоичной, десятичной и шестнадцатеричной формах).

Некоторые операционные системы предусматривают возможность внесения изменений в таблицы графических изображений символов (ПЭЕМ "Acorn", "Yamaha"). В таком случае пользователь может создать свой алфавит или шрифт для выводимого на экран текста. Другие микроЭВМ требуют более глубокого изучения работы операционной системы для модификации таблиц графических изображений символов.

Различные режимы экрана накладывают определенные ограничения на вывод графической и текстовой информации. Можно выделить три вида режимов:

1) текстовый режим экрана, допускающий вывод только текстовой информации, например, режимы экрана 3, 6, 7 для микроЭВМ "Acorn", режимы экрана 0, 1 для микроЭВМ "Yamaha";

2) объединенный текстовый и графический режим экрана, например, режимы экрана 0, 1, 2, 4, 5 на микроЭВМ "Acorn", режимы экрана 2, 3 на микроЭВМ "Yamaha", режим экрана микроЭВМ

Рис. 1. Определение графического изображения символа.

.***.**,	01100110	102	66
.**,**,	01101100	108	6C
.****,	01111000	120	78
.***...	01110000	112	70
.****...	01111000	120	78
.**,**,	01101100	108	6C
.**..**.	01100110	102	66
.....	00000000	0	0

Рис. 2. Отображение графика функции в окне экрана W.

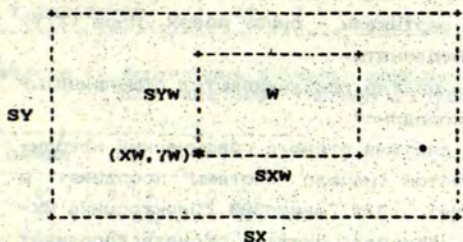


Рис. 3. Определение символа спрайта.

...***.	00011100	28	1C
..***.	00011100	28	1C
.*...*	01001000	72	48
..****.	00111110	62	3E
...*.*	00001001	9	9
..***.	00111000	56	38
..*...*	00101110	46	2E
.*.....	00100000	32	20

"Корвет":

3) графический режим экрана, допускающий только вывод графической информации.

## 2. Геометрия экрана

Точку на плоскости обычно представляют парой чисел  $(X, Y)$ . Эта пара указывает на расстояние точки от неких осей координат. Число  $X$  задает расстояние по горизонтали,  $Y$  - по вертикали. Обычно расстояние по горизонтали принято отсчитывать слева направо, а по вертикали - снизу вверх. Пара  $(X, Y)$  известна как декартовы координаты.

На экране монитора не всегда придерживаются общепринятых правил отсчета значений координат.

Для микроЭВМ "Acorn" система отсчета графических координат совпадает с общепринятой (начало системы координат в нижнем левом углу экрана). Для микроЭВМ "Электроника БК-0010", "Агат", "Yamaha", "Корвет" система отсчета координат начинается с верхнего левого угла экрана, координата  $X$  отсчитывается слева направо, координата  $Y$  - сверху вниз. Позиция экрана для вывода определяется парой координат  $(X, Y)$ .

Система отсчета текстовых координат для всех рассматриваемых микроЭВМ начинается с верхнего левого угла экрана. Допустимые значения координат для текстового режима экрана разных микроЭВМ см. в табл. 1, 6, 7, 8. В тех режимах экрана, где допускается вывод одновременно текстовой и графической информации, у некоторых типов микроЭВМ имеется возможность вывода текста по графическим координатам экрана, например, у микроЭВМ "Acorn" и "Yamaha". Координаты точки определяют положение верхнего левого угла позиции символа.

## 3. Построение графиков функций

Построить график функции - это значит для каждого значения  $x$  из определенной области вычислить значение функции  $y$  и закрасить точку с координатами  $(X, Y)$  на экране монитора или провести линию от предыдущей точки. При построении графиков функций мы имеем дело, с одной стороны, с областью изменения

Таблица 1. Характеристики экранных режимов ПЭВМ "Асорт"

режим	цвета	разрешимость	текст	память экрана
0 текстовый и графический	2	640x256	80x32	20 К
1 ""	4	320x256	40x32	20 К
2 ""	16	160x256	20x32	20 К
4 ""	2	320x256	40x32	10 К
5 ""	4	160x256	20x32	10 К
3 текстовый	2	-	80x25	10 К
6 ""	2	-	40x25	8 К
7 телетекст	16	-	40x25	1 К

Таблица 2. Допустимые значения координат X и Y для текстовых экранов на ПЭВМ "Асорт"

режим	X	Y
0	0-79	0-31
1	0-39	0-31
2	0-19	0-31
3	0-79	0-24
4	0-39	0-31
5	0-19	0-31
6	0-39	0-24
7	0-39	0-24

Таблица 3. Графические режимы 0 и 4 (2 цвета) ПЭВМ "Асорт"

лог. номер	передний план	лог. номер	фон
0	черный	128	черный
1	белый	129	белый

Таблица 4. Графические режимы 1 и 5 (4 цвета) ПЭВМ "Асорт"

лог. номер	передний план	лог. номер	фон
0	черный	128	черный
1	красный	129	красный
2	желтый	130	желтый
3	белый	131	белый

Таблица 5. Текстовые режимы 3 и 6 (2 цвета) ПЭВМ "Acorn"

лог. номер	передний план	лог. номер	фон
0	черный	128	черный
1	белый	129	белый

Таблица 6. Характеристики экранных режимов ПЭВМ "Электроника БК-0010"

режим (симв. в строке)	цвета	разрешимость	текст	память экрана
32	4	256x240	32x24	16 К
64	2	512x240	64x24	16 К

Таблица 7. Характеристики экранных режимов ПЭВМ "Yamaha"

режим	цвета	разрешимость	текст	память экрана
0 текстовый	16	-	40x24	16 К
1 текстовый со спрайтами	16	-	32x24	16 К
2 тонкая графика, текст, спрайты	16	192x256	32x24	16 К
3 грубая графика и спрайты	16	48x64	8x3	16 К

Таблица 8. Характеристика экранных режимов ПЭВМ "Agat"

режим	цвета	разрешимость	память экрана
Графический высокого разрешения	2	256x256	8 К
Графический среднего разрешения	16	128x128	8 К
Графический низкого разрешения	16	64x64	2 К
Алфавитно-цифровая	2	64x32	2 К
Алфавитно-цифровая	16	32x32	2 К

значений  $x$  и  $y$  для функции и, с другой стороны, с областью значений координат  $X$  и  $Y$  на экране, определяемой количеством точек на экране для конкретного микрокомпьютера.

В случае полного использования экрана монитора для изображения графика функции из заданной области значений порядок действий в программе может быть следующим.

1. Определяется (приблизительно) область значений функции для  $x$  из области аргумента, допустим

$$A < x < B, \text{ в } F_{\min} < f(x) < F_{\max} \text{ для } x \in ]A, B[.$$

2. Уточняется количество точек экрана для конкретного компьютера и используемого режима экрана. Допустим, что

$$S_X - \text{количество точек по горизонтали,}$$

$$S_Y - \text{количество точек по вертикали.}$$

3. Определяется масштаб отображения графика функции на экране

$$\text{по горизонтали} - M_X = S_X / (B - A),$$

$$\text{по вертикали} - M_Y = S_Y / (F_{\max} - F_{\min}).$$

4. Определяются координаты  $(X, Y)$  на экране для точки функции  $(x, f(x))$ , а именно

$$X = M_X * (x - A),$$

$$Y = M_Y * (f(x) - F_{\min}).$$

При отсчете координат на экране сверху вниз, для  $Y$  соотношение имеет вид

$$Y = S_Y - M_Y * (f(x) - F_{\min}).$$

Для отображения графика функции в окне экрана  $W$  (см. рис. 2) прежде всего уточняется область значений для функции.

Допустим, что координаты левого нижнего угла окна -  $(X_W, Y_W)$ , размер окна -  $S_{XW}$  точек по горизонтали и  $S_{YW}$  точек по вертикали.

Масштаб

$$M_X = S_{XW} / (B - A),$$

$$M_Y = S_{YW} / (F_{\max} - F_{\min}).$$

Значения координат на экране для соответствующих точек функции  $(x, f(x))$  вычисляются из соотношения

$$X = X_W + S_{XW} * (x - A)$$

и

$$Y = Y_W + S_{YW} * (f(x) - F_{\min}), \text{ если отсчет снизу вверх,}$$

$$Y = S_{YW} - (Y_W + S_{YW} * (f(x) - F_{\min})), \text{ если отсчет сверху}$$

ЕНИА.

График может быть построен и в текстовом режиме. В этом случае вместо закрашивания точек графического экрана используется вывод символа в позиции. Масштаб отображения графика функции в текстовом режиме экрана определяется по описанным выше соотношениям, только величины SX, SY, SXW, SYW указывают количество текстовых позиций по горизонтали и по вертикали для всего экрана и в окне экрана.

#### 4. Вывод информации на экран для микроЭВМ "Acorn", "Yamaha", "Корвет", "Агат", "Электроника БК-0010"

##### 4.1. Организация экрана.

"Acorn". Имеется 8 экранных режимов (см. табл. 1).

"Электроника БК-0010". Имеются 2 экранных режима для вывода текстовой и графической информации одновременно (см. табл. 6).

"Yamaha". Имеется 4 экранных режима (см. табл. 7). В графических режимах каждые 8 точек по горизонтали могут иметь цвет фона или цвет переднего плана. Поэтому, при пересечении двух разноцветных линий возможно искажение цвета.

"Агат". Имеется 5 экранных режимов (см. табл. 8).

"Корвет" Имеется 1 экранный режим для вывода текстовой и графической информации.

##### 4.2. Вывод графической информации на экран.

###### 4.2.1. Установка графического режима экрана.

"Acorn". Графический режим экрана в БЕЙСИКе устанавливается оператором MODE N или VDU 22,N (где N=0,1,2,4 или 5).

"Электроника БК-0010". В ФОКАЛе графический режим устанавливается оператором X PCHR(155).

"Yamaha". Графический режим экрана устанавливается оператором SCREEN <режим> [, <величина спрайтов>]. Здесь <режим> - 2 или 3, <величина спрайтов> - 0,1,2 или 3.

"Агат". Графический режим экрана в БЕЙСИКе устанавливается следующими операторами:

HGR=N - включение графики высокого разрешения, N - номер страницы размером 8 К (1 <= N <= 15);

MGR=N - включение графики среднего разрешения, N - номер стра-

ницы размером 8 К ( $1 \leq N \leq 15$ ):

GR=N - включение графики низкого разрешения. N - номер страницы размером 2 К ( $1 \leq N \leq 63$ ).

"Корвет". Нет необходимости в специальной установке режима.

#### 4.2.2. Очистка графического экрана.

"Acorn". Для очистки графического экрана в БЕЙСИКЕ используется оператор CLR или VDU 16.

Курсор снимается оператором VDU 23,1,0;0;0;0; и восстанавливается оператором VDU 23,1,1;0;0;0;.

"Электроника БК-0010". В ФОКАЛе экран очищается оператором X FCLR(12).

Курсор снимается/восстанавливается оператором X FCLR(15)

"Yamaha". Для очистки графического экрана применяется оператор SCREEN (см. 4.2.1) или оператор LINE (чаще для очистки части экрана) (см. 4.2.6).

"Агат". Очистку экрана с помещением курсора в верхнем левом углу экрана производит оператор HOME или PRINT CHR(12).

"Корвет". Оператор PCLS стирает графическую информацию.

#### 4.2.3. Определение цвета.

"Acorn". В БЕЙСИКЕ имеются операторы определения рабочего набора цветов из 16 доступных, а также операторы выбора цвета из рабочего набора для вывода графической информации.

Каждый режим экрана имеет стандартный рабочий набор цветов для фона и переднего плана (см. табл. 9). В режиме 2 (16 цветов), рабочий набор совпадает с полным набором цветов. Для изменения рабочего набора цветов используется оператор VDU 19,N,M,0,0,0 (где N - логический номер цвета, M - номер цвета).

Выбрать цвет для переднего плана или фона из рабочего набора цветов можно оператором GCOL 0,N (N - логический номер цвета).

"Электроника БК-0010". В ФОКАЛе цвет определяется передачей драйверу экрана кодов управления оператором X FSKR(K) (K - код цвета, см. табл. 9).

"Yamaha". Цвет устанавливается с помощью оператора COLOR [<цвет текста>] [<цвет фона>] [<цвет рамки>].

Из 256 возможных цветов доступен набор из 16 цветов (стандартный набор см. табл. 9). Для переопределения этого

Таблица 9. Кодировка цветов для микрокомпьютеров

цвет	номер цвета					ЕК-0010 (код драйвера ТВ)
	Асорг		Yamaha	Агат	Корвет	
	передн. план	фон				
черный	0	128	1	0: 8	0	188
бордовый	-	-	6	-	-	-
красный	1	129	8	1: 9	4	185*
розовый	-	-	9	-	-	-
свальный	-	-	3	-	-	-
зеленый	2	130	2	2:10	2	186**
темно-зеленый	-	-	12	-	-	-
светло-желтый	-	-	11	-	-	-
желтый	3	131	10	3:11	-	-
бирюзовый	6	136	7	-	-	-
голубой	-	-	5	6:14	3	-
синий	4	132	4	4:12	1	187**
фиолетовый	5	133	13	5:13	5	-
коричневый	-	-	-	-	6	-
серый	-	-	14	-	-	-
белый	7	135	15	7:15	7	-
мерцающий чер- ный/белый	8	136	-	-	-	-
мерцающий крас- ный/бирюзовый	9	137	-	-	-	-
мерцающий зеле- ный/фиолетовый	10	138	-	-	-	-
мерцающий жел- тый/синий	11	139	-	-	-	-
мерцающий си- ний/желтый	12	140	-	-	-	-
мерцающий фио- летовый/зеленый	13	141	-	-	-	-
мерцающий бирю- зовый/красный	14	142	-	-	-	-
мерцающий бе- лый/черный	15	143	-	-	-	-

\* - белый цвет на черно-белом мониторе

\*\* - полутон на черно-белом мониторе

набора необходимо изменить содержимое регистров памяти видео-процессора.

**"Агат"**. Один из возможных 16 цветов для последующих графических операторов (см. табл. 9) устанавливается оператором COLOR=X, где X - номер цвета.

**"Корвет"**. Цвета, используемые в графических операциях, задаются оператором COLOR [цвет текста] [цвет фона], значения параметров см. в табл. 9.

В операторе COLOR указываются логические номера цветов. Для изменения соответствия логических и физических номеров цветов используется оператор LUT (XX(1)). Элементы массива с индексами от 1 до 1+15 содержат физические номера цветов, в их порядковый номер соответствует логическому номеру. До использования оператора LUT, логические номера 0-7 соответствуют цветам графики, а 8-15 - цветам графики с расположенной поверх графической информации текстовой информацией.

#### 4.2.4. Системы экранных координат графического вывода.

**"Acorn"**. Во всех режимах графический экран имеет следующие пределы координат: X - от 0 до 1279; Y - от 0 до 1623.

Оператор VDU 29.X:Y; позволит перенести начало системы координат в точку (X,Y).

**"Электроника БК-0010"**. Координаты в графическом режиме экрана отсчитываются слева направо (X) и сверху вниз (Y). В режиме 32 символа в строке координата X меняется от 0 до 255, в режиме 64 символа в строке - от 0 до 511. Координата Y в обоих режимах меняется от 0 до 239.

**"Yamaha"**. Координата графического экрана X принимает значения от 0 до 255 и отсчитывается слева направо, Y - от 0 до 191 и отсчитывается сверху вниз.

**"Агат"**. Координаты в графическом режиме экрана отсчитываются слева направо (X) и сверху вниз (Y): В режиме HGR координаты меняются от 0 до 255, в режиме MGR - от 0 до 128, и в режиме GR - от 0 до 63.

**"Корвет"**. Координата графического экрана X принимает значения от 0 до 511 и отсчитывается слева направо, Y - от 0 до 255 и отсчитывается сверху вниз.

Оператор RELOC (X,Y) устанавливает начало системы координат при работе с графическими операторами в точку (X,Y).

#### 4.2.5. Отображение точки.

"Acorn". В БЕЙСИКЕ точка отображается оператором PLOT 69, X, Y. Точку без рисования можно установить оператором MOVE X, Y или PLOT 4, X, Y.

"Электроника БК-0010". В ФОКАЛЕ точка отображается оператором X FT(N, X, Y), где N=0 для гашения, N=1 для закрашивания точки. X и Y - экранные координаты точки.

"Yamaha". Точка с координатами (X, Y) отображается оператором PSET (X, Y), <цвет>. Оператор PRESET (X, Y) окрашивает точку в цвет фона.

"Agat". Оператор PLOT X, Y помещает точку текущего цвета в позицию с координатами (X, Y).

"Корвет". Операторы аналогичны операторам ПЗВМ "Yamaha".

#### 4.2.6. Отображение линии.

"Acorn". Линия от предыдущей точки до точки с координатами (X, Y) проводится оператором DRAW X, Y или PLOT 5, X, Y.

"Электроника БК-0010". В ФОКАЛЕ оператор X FV(N, X, Y) рисует вектор от предыдущей точки к точке с координатами (X, Y), если N=1, или стирает вектор, если N=0.

"Yamaha". Линия из точки (X0, Y0) в точку (X1, Y1) рисуется оператором

LINE [STEP] [(X0, Y0)] - [STEP] (X1, Y1) [,<цвет>] [,B[F]]

Наличие STEP указывает на относительные координаты. При отсутствии первой пары координат используется текущее положение графического курсора. При использовании указателя в место линии рисуется прямоугольник, где (X0, Y0) - координаты верхнего левого угла прямоугольника, (X1, Y1) - нижнего правого угла. При BF рисуется закрашенный прямоугольник.

"Agat". Оператор PLOT [X0, Y0] TO X1, Y1 рисует линию текущего цвета от точки (X0, Y0) к точке (X1, Y1). Если координаты начальной точки не указаны, используется конечная точка последнего выполненного оператора PLOT.

"Корвет". Используется оператор LINE в таком же формате как на ПЗВМ "Yamaha".

#### 4.2.7. Закрашивание треугольника.

"Acorn". В БЕЙСИКЕ треугольник закрашивается оператором PLOT 85, X, Y, где используются, кроме указанной, две предыдущие позиции графического курсора.

Для ПЭВМ "Электроника БК-0010", "Yamcha", "Агат" и "Корвет" соответствующий оператор отсутствует.

4.2.8. Отображение и закрашивание прямоугольника.

Для ПЭВМ "Асоgn", "Электроника БК-0010" и "Агат" соответствующий оператор отсутствует.

Для ПЭВМ "Yamaha", "Корвет" используется оператор LINE (см. 4.2.6).

4.2.9. Отображение окружности (эллипса).

Для ПЭВМ "Асоgn", "Электроника БК-0010" и "Агат" соответствующий оператор отсутствует.

"Yamaha". Окружность, эллипс или дуга с центром (X,Y) и заданным радиусом рисуется оператором CIRCLE [STEP] (X,Y), <радиус> [, <цвет>] [, <начало>, <конец>] [, <отношение осей>].

где <начало>, <конец> - начальный и конечный углы дуги в радианах, <отношение осей> - отношение длины отрезка, заданного на оси X в эллипсе, к длине отрезка, заданного на оси Y. STEP указывает на относительные координаты.

"Корвет". Используется оператор CIRCLE, аналогично, как на ПЭВМ "Yamaha".

4.2.10. Закрашивание замкнутой части экрана.

Для ПЭВМ "Асоgn", "Электроника БК-0010" и "Агат" соответствующий оператор отсутствует.

"Yamaha". Закрашивание замкнутой фигуры, в пределах которой находится точка (X,Y), проводится оператором PAINT [STEP] (X,Y) [, <цвет> [, <цвет границы>]]

Параметр STEP указывает на относительные координаты. Цвет границы закрашиваемой области задается в режиме 3 в том случае, если он отличается от цвета закрашивания.

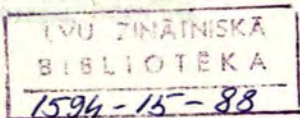
"Корвет". Используется оператор PAINT так же, как на ПЭВМ "Yamaha".

4.3. Особенности вывода текстовой информации.

4.3.1. Установка режима вывода текстовой информации.

"Асоgn". Во всех режимах экрана имеется возможность вывода текста (см. 4.1, 4.2.1).

"Электроника БК-0010". В ФОКАЛе режим 32 или 64 символов в строке переключается оператором X FCHR(155).



"Yamaha". Текстовый режим устанавливается оператором SCREEN N (где N=0 или 1). При включении микроЭВМ автоматически устанавливается режим 0.

"Агат". Текстовый режим устанавливает оператор TEXT=N, где N - номер страницы (от 2 до 63). Страницы от 2 до 31 - цветные 32x32 символа, от 32 до 63 - черно-белые 32x64 символа.

Существуют три режима вывода текстовой информации для режима 32x32 символа. Оператор:

NORMAL включает прямой (светлый по черному) режим;

INVERSE включает инверсный (черный по светлому) режим;

FLASH включает мерцающий режим вывода текста.

"Корвет". Нет необходимости в специальной установке режима.

#### 4.3.2. Очистка текстового экрана.

"Acorn". В БЕЙСИКЕ для очистки экрана используется оператор CLS или VDU 17.

Курсор снимается оператором VDU 23,1,0;0;0;0; и восстанавливается оператором VDU 23,1,1;0;0;0;.

"Электроника БК-0010". Так же, как в графическом режиме (см. 4.2.2).

"Yamaha". Очистка экрана проводится оператором CLS.

"Агат". Очистку экрана см. в 4.2.2. Текстовое окно, начиная с позиции курсора до конца окна, очищает оператор PRINT CHR\$(30). Строка, начиная с позиции курсора до правой границы текстового окна, очищается оператором PRINT CHR\$(13).

"Корвет". Текстовая информация стирается оператором CLS, курсор помещается в позицию экрана (1,1).

#### 4.3.3. Определение цвета.

"Acorn". Во всех режимах экрана, кроме режима телетекста, цвет фона и переднего плана определяется оператором COLOUR N (N - логический номер цвета фона или переднего плана - см. табл. 5). В режиме телетекста (7) цвет для группы символов в строке определяет управляющий код перед этой группой в операторе PRINT. В режиме телетекста - 16 цветов.

"Агат". Цвет символов в текстовом режиме устанавливается оператором RIBBON=X, где X - номер цвета.

"Корвет". Для изменения цвета выводимой текстовой информации необходимо использовать оператор LUT.

Для ПЭВМ "Электроника БК-0010" и "Yamaha" нет отличий от

графического режима (см. 4.2.3).

#### 4.3.4. Определение места вывода текста.

**"Acorn"**. Допустимые значения координат X и Y см. в табл. 2. Позицию вывода текста определяет функция TAB(X,Y) в операторе PRINT. Оператор PRINT TAB(X,Y): эквивалентен оператору VDU 31,Y,Y.

В графических режимах экрана оператор VDU 5 устанавливает вывод по графической координате. Оператор VDU 4 устанавливает вывод по текстовой позиции.

**"Электроника БК-0010"**. В ФОКАЛе для управления курсором используется оператор X FK(X0,Y0), где X0 позиция в строке (от 0 до 31 или 63), Y0 - номер строки (от 0 до 23). Для вывода текста с новой строки в операторе TYPE перед нужным текстом надо указать символ "!".

**"Yamaha"**. Курсор в нужной позиции устанавливается оператором LOCATE X,Y [,<курсor>]. В режиме 0 - значение X от 0 до 39, в режиме 1 - от 0 до 31, значение Y всегда от 0 до 23. Если <курсor>=0, курсор исчезает, при других значениях - виден.

**"Агат"**. Оператор VTAB Y устанавливает курсор на строку экрана с номером Y (от 0 до 31). Оператор HTAB X устанавливает курсор на X-ю позицию текущей строки. В режиме 32x32 - X от 1 до 32, в режиме 32x64 - X от 1 до 64.

Аналогом оператора HTAB является функция TAB(X) в операторе PRINT. Если X меньше значения текущей горизонтальной позиции курсора, функция игнорируется.

**"Корвет"**. Текстовый курсор устанавливается оператором LOCATE [X] [[,Y] [,<курсor>]]. где X - от 1 до 64, Y - от 1 до 16. <курсor> используется для включения (1) и стlкючения (0) изображения курсора.

В операторе PRINT можно использовать функцию TAB(N), где N в пределах от 1 до 255, для установки курсора в N-ю позицию.

#### 4.3.5. Вывод отдельных символов, заданных значением кода.

Для ПЗВМ **"Acorn"**, **"Yamaha"**, **"Агат"** и **"Корвет"** вывод символа, заданного кодом ASCII (**"Acorn"**, **"Yamaha"**) или КОИ-8 (**"Агат"**, **"Корвет"**) используется функция CHR\$(K) в операторе PRINT, где K - значение кода.

**"Электроника БК-0010"**. В ФОКАЛе для вывода на экран символа по указанному коду используется оператор X FCHR(K), где K -

значению кода.

"Yamaha". При этом часть графических символов имеет составной код:  $CHR\$(1)+CHR\$(n+64)$  (n от 1 до 31).

#### 4.3.6. Вывод текстовых строк.

Для ПЭВМ "Acorn", "Yamaha", "Агат", "Корвет" в БЕЙСИКЕ для вывода текстовой строки в списке оператора PRINT указываются символьные переменные и символьные константы.

"Электроника БК-0010". В ФОКАЛе для вывода текстовых строк используется оператор T "<текст>".

"Yamaha". Вывод текста или чисел в графических режимах возможен только через устройство GRP. Устройство GRP необходимо открыть для вывода оператором OPEN "GRP:" FOR OUTPUT AS #1. Вывод текста производится по графическим координатам: PRINT #1, <символьная переменная или константа>.

#### 4.3.7. Вывод чисел.

"Acorn". Для вывода чисел в списке оператора PRINT указываются числовые переменные, числовые константы. Формат выводимых значений задается переменной %X, где 1-й байт задает форму (с фиксированной (2) или с плавающей точкой (1)), 2-й байт - количество цифр в дробной части, 3-й байт - длину зоны. Например, %X=220309 задает формат вывода числа в зоне длиной 9 позиций с фиксированной точкой и с тремя цифрами в дробной части ("0" означает шестнадцатеричную цифру). Стандартный формат вывода: %X=10.

"Электроника БК-0010". В ФОКАЛе для вывода чисел используется оператор T <число или числовая переменная>.

"Yamaha". Для вывода чисел в списке оператора PRINT указываются числовые переменные или константы. Формат выводимых значений задается символьной строкой или переменной. Например, оператор PRINT USING "###.###":N будет печатать число N с 3 цифрами перед и 3 после десятичной точки.

"Агат". Для вывода чисел в списке оператора PRINT указываются числовые переменные, числовые константы.

"Корвет". Вывод чисел, а также установление формата, производится так же, как на ПЭВМ "Yamaha".

#### 4.4. Дополнительные возможности.

##### 4.4.1. Режим экранной графики "Электроники БК-0010".

Для включения режима необходимо:

1. Нажать клавиши <НР> и <РЕД>. В служебной строке появляется сообщение "РЕД".
2. Нажать клавишу <ГРАФ>. В служебной строке появляется сообщение "ГРАФ", и на экране - графический курсор "+".
3. Графическим курсором можно управлять с помощью клавиш управления курсором. Курсор перемещается с шагом в одну точку, не оставляет следа, если включен режим ГРАФ.
4. Для рисования необходимо включить режим ЗАП, для стирания - СТИР. В служебной строке появляются соответствующие сообщения - "ЗАП" или "СТИР".
5. Если включить режим ВЛР РЕД, символы управления и направления перемещения курсора можно записать в строке программы в явном виде.
6. Если перед символом перемещения курсора записать десятичное число, курсор в данном направлении будет перемещаться на соответствующее количество точек.
7. Для редактирования строки необходимо перед выводом строки на экран включить режим ВЛР РЕД, а перед редактированием отключить.

Например, оператор T "Г---35:5--5:5--5Г" нарисует квадрат, состоящий из символов "\*" со стороны в 5 знаков.

##### 4.4.2. Графические объекты на микроЭВМ "Агат".

Интерпретатор языка БЕЙСИК имеет 4 оператора, позволяющие манипулировать графическими объектами в режиме графики высокого разрешения - DRAW, XDRAW, ROT, SCALE. Однако прежде чем эти операторы могут быть исполнены, графический объект должен быть описан с помощью так называемого "описателя формы". Описатель формы состоит из последовательности закодированных векторов, определяющих форму объекта. Один или несколько описателей формы вместе с указателями составляют "таблицу образов". Коды этой таблицы могут быть набраны с блока клавиатуры и записаны на внешнем носителе для последующего использования.

#### 4.4.3. Макроязык графики и спрайты на микроЭВМ "Yamaha".

На микроЭВМ "Yamaha" в БЕЙСИКе существует оператор, который выполняет сразу последовательность графических команд, рисуя на экране дисплея сложную линию или фигуру:

`DRAW` <символьная константа или переменная>

В качестве параметра для оператора задается последовательность команд макроязыка графики. Макроязык содержит команды, указывающие в каком направлении рисовать отрезок линии и какой длины, с какой точки начать линию, команды устанавливающие цвет рисунка. Используя команды макроязыка, можно указать ориентацию рисуемой фигуры на экране или масштаб изображения. Возможно выполнение графических подпрограмм с включением в список команд имени символьной переменной, которая в свою очередь является последовательностью команд макроязыка.

Другая часто используемая графическая возможность Бейсика - спрайты. Это объекты 8x8 точек, которые используются для создания движущихся рисунков на экране дисплея.

Имеется 32 изображаемых спрайта. Каждый характеризуют четыре величины: цвет, координаты X и Y, изображаемый символ. Кроме того, существуют два общих параметра: размер спрайта (нормальный или в два раза больший) и длина символа спрайта (8 для 8x8 точек или 32 для 16x16 точек). Эти параметры устанавливает оператор `SCREEN` (см. 4.2.1).

Каждый спрайт может использовать только один цвет. На рис. 3 ("\*" - закрасенные точки, "." - прозрачные точки) показано, как вычисляются значения для определения символа спрайта в двоичной, десятичной и шестнадцатеричной формах.

Данные можно поместить в таблицу изображений с помощью псевдопеременной `SPRITE$(NI)`, где NI - номер символа спрайта в пределах 0-255, если `SCREEN,0` или `SCREEN,1`, и 0-63, если `SCREEN,2` или `SCREEN,3`, например, в данном случае

```
SPRITE$(0) = CHR$(28) + CHR$(28) + CHR$(72) + CHR$(62) +  
CHR$(9) + CHR$(56) + CHR$(46) + CHR$(32).
```

Определенный символ можно выводить на экран с помощью оператора `PUT SPRITE NS [, [STEP] (X,Y)] [, <цвет>] [, NI]`, где NS - номер спрайта 0-31.

При выводе на экран можно поместить несколько спрайтов в

одну позицию и получить один разноцветный спрайт или поместить рядом, получив один большой спрайт. Однако в одной строке можно разместить только 4 спрайта. Для движения спрайтов используется оператор цикла.

## В ы в о д ы

Возможности языка ВЕЙСИК на разных микроЭВМ различны. Разработка общей для всех микроЭВМ программы представляется чрезвычайно затруднительной.

Графические средства языка существенно зависят от характеристик аппаратуры, в первую очередь от разрешающей способности дисплея. Предложение [ 1 ] принять в качестве стандартного дисплей с разрешающей способностью 256x192 точек (отсчет координат с левого верхнего угла экрана) не решает проблему переносимости программ ввиду существенного отличия списка графических операторов. К тому же на микроЭВМ с более совершенными дисплеями изображение будет занимать только часть экрана.

В большинстве случаев в графических построениях не используются возможности операторов закрасивания. Если обойтись рисованием точек и линий, то различия в программировании вывода графики не так существенны, и адаптация таких программ на другой тип микрокомпьютера не требует больших затрат труда. Константы и операторы, зависящие от типа микрокомпьютера, желательно локализовать в определенных блоках программы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Александров А. Л., Башмакова Е. С., Гуткин М. Л., Либеров А. В. О Стандарте языка Вейсик // Микропроцессорные средства и системы. 1986. # 5. С. 25-30.
2. Математическое обеспечение микроЭВМ "Электроника БК-0010". Язык Фокал. Руководство пользователя. Том 1. Книга 1. 1985. 15 с.
3. Математическое обеспечение микроЭВМ "Электроника БК-0010". Руководство системного программиста. Том 1. Книга 2. 1985. 78 с.
4. Математическое обеспечение микроЭВМ "Электроника БК-0010". Руководство пользователя микроЭВМ по языку "Вейсик - MSX". 1986. 78 с.
5. Машина вычислительная электронная персональная "Агат" Система программирования "Вейсик". Интерпретатор языка "Вейсик". Описание языка. 1985. 28 с.
6. Машина вычислительная электронная персональная "Агат" Система программирования "Вейсик". Интерпретатор языка "Вейсик". Руководство программиста. 1985. 36 с.
7. Базовые программные средства ПК8020. Интерпретатор языка Вейсик. Описание языка. 1988. 116 с.
8. Coll J. User Guide The BBC Microcomputer. London: British Broadcasting Corporation, 1982. P. 519.
9. Yamaha. MSX BASIC Reference Manual. New York: ASCII Corporation, 1985. P. 355.

Межвузовский сборник научных трудов  
ЭВМ В ОБРАЗОВАНИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
1988, Фига: ЛГУ им. П. Стучки. С. 25-37

УДК 681.3.06

О. Н. Елкина, А. А. Курмис, И. Я. Озис,  
С. И. Павлов, А. Р. Пайкенс, М. О. Ёглайс  
Вычислительный центр при ЛГУ им. П. Стучки

#### ПРЕДСТАВЛЕНИЕ НАЦИОНАЛЬНЫХ АЛФАВИТОВ НА ПЕРСОНАЛЬНЫХ ЭВМ И КОДОВЫЕ ТАБЛИЦЫ СИМВОЛОВ

Изобретение и массовое производство персональных ЭВМ (ПЭВМ), укомплектованных высококачественными периферийными устройствами (в т. ч. графическими цветными мониторами, почта-ташными устройствами, накопителями большой информационной емкости и др.), существенно расширяет сферу применения средств вычислительной техники как инструмента профессиональной деятельности в различных отраслях народного хозяйства, научных исследованиях, управлении, а также как технического средства обучения в образовании.

Эффективность использования ПЭВМ в значительной мере зависит от качества системного и прикладного программного обеспечения, средств реализации "дружественной" среды, диалога с ПЭВМ на удобном для пользователя языке.

Для поддержки диалога пользователя и ПЭВМ в зависимости от решаемой задачи необходимы следующие алфавиты:

- русский;
- национальный (латышский, литовский, эстонский, украинский, белорусский и др.);
- иностранный (английский, немецкий, французский и др.);
- греческий.

а также наборы специальных знаков (математических, астрономических, географических и др.).

В научных исследованиях, образовании (особенно по предметам естественно-научного цикла), а также при разработке к

адаптации программного обеспечения перечисленные алфавиты и специальные знаки зачастую нужны одновременно, в пределах одного комплекса программ.

Одновременное использование латинского и русского алфавитов в ПЭВМ обсуждалось в [ 1,2 ]. В настоящей работе рассматривается кодирование изображений символов для дисплея и печатающего устройства, а также формирование кодовой таблицы, включающей символы одновременно трех алфавитов - русского, латышского и латинского. Изложение ведется на примере ПЭВМ "Ассол". Аналогичные проблемы с учетом возможностей ПЭВМ и периферийных устройств решались также для ПЭВМ "Yamaha", "Корвет", "Электроника ЕК-0010".

#### 1. Кодировка символов для отображения на дисплее и печатающем устройстве ПЭВМ

В памяти ПЭВМ хранится таблица с закодированными начертаниями символов, представляющими собой рисунки из 8x8 точек - т.н. матрицы знака. При нажатии клавиши вырабатывается код, и по хранящемуся в памяти начертанию символа строится его изображение на экране ПЭВМ. Системные средства предоставляют возможность переопределения начертаний символов. Для кодирования каждого символа необходимо 9 байт: один байт с кодом символа и по одному байту на каждую из восьми строк матрицы. На рис. 1 показаны матрицы знака для вывода на экран некоторых букв русского и латышского алфавитов, а также кодировка их начертания в двоичной и шестнадцатеричной системе счисления. Синтезированные наборы шрифтов используются для разработки программ с диалогом на русском и латышском языках. На рис. 2 представлены копии двух экранов - "заставок" программы с диалогом на русском (а) и латышском (б) языках для выбора модификаций текстового редактора и режимов работы печатающего устройства [ 3 ].

Для вывода символов на печать применяется знаковосинтезирующий принтер "ТАХАН-КАГА". Принтер предоставляет возможность кроме стандартного набора символов, начертания которых хранятся в ПЗУ, распечатывать наборы символов, начертания

Рис. 1. Кодирование изображений символов для дисплея.


		двоичное значение	шестнадцатеричное значение
1	****	00111100	3C
2		00000000	00
3	****	00111100	3C
4	**	00000110	06
5	*****	00111110	3E
6	** **	01100110	66
7	*****	00111110	3E
8		00000000	00
1	**	00011000	18
2		00000000	00
3	*****	00111110	3E
4	** **	01100110	66
5	** **	01100110	66
6	*****	00111110	3E
7	**	00000110	06
8	****	00111100	3C
1		00000000	00
2		00000000	00
3	***	11100000	E0
4	**	01100000	60
5	*****	01111100	7C
6	** **	01100110	66
7	*****	01111100	7C
8		00000000	00

9U	3C 00 3C 06 3E 66 3E 00
BB	18 00 3E 66 66 3E 06 3C
FF	00 00 E0 60 7C 66 7C 00

Рис. 2. Примеры организации экрана в программах с диалогом на русском (а) и латышском (б) языках.

<b>Лаборатория проблем школьной и вузовской информатики</b>	<b>ЛГУ</b> $\Sigma$ <b>ВУ</b>
<b>ТЕКСТОВЫЙ РЕДАКТОР</b>	
1. Русско-латинский	
2. Русско-латинско-латышский	
<b>Нажмите:</b> 1 $\overline{L}$ 2 - latviešu valoda номер раздела	

 <b>Rīga</b>	<b>Skolu un augstskolu informātikas problēmu laboratorija</b>	<b>LOU</b> $\Sigma$ <b>SC</b>
<b>TEKSTU REDAKTORS</b>		
1. Latviešu-latīņu		
2. Latviešu-latīņu-krievu		
<b>Nospiediet:</b> 1 $\overline{R}$ 2 - русский язык numurs		

которых переопределяются пользователем. Переопределенные наборы символов загружаются в принтер с помощью специальных программ [ 3 ] и хранятся в дополнительном ОЗУ.

Изображение символа, выводимого на печать, представляет собой рисунок из 8x6 точек (8 строк, 6 столбцов). Головка принтера одновременно выводит изображение одной вертикальной позиции знака. Так как возможно смещение головки по горизонтали на пол-иглы, то фактически получается 11 вертикальных позиций для одного знака. Для кодирования каждого символа в матрице необходимо 13 байт: один байт - код символа, один байт - сдвиг (опустить или нет изображение на толщину иглы), 11 байт - 11 вертикальных позиций знака. На рис. 3 показаны изображения некоторых символов русского и латвийского алфавитов в матрице 8x11 и их представление в кодированной форме в двоично-шестнадцатеричной системе счисления.

В меню принтера имеется ряд контрольных кодов, позволяющих выбирать режимы печати: изменять расположение символов и строк на листе; на основе заданных матриц символов синтезировать разнообразные шрифты, различающиеся шириной и высотой букв. Возможности принтера использованы в программе диалога по выбору режимов печатающего устройства [ 3 ].

Конструкция принтера предусматривает возможность генерации и вывода на печать NLQ-шрифта, по качеству близкого к пишущей машинке (Near Letter Quality). Изображение знака представляется в матрице 16x23, а для кодирования одного символа необходимо 49 байт (46 байт для 23 позиций головки принтера). На рис. 4 показано кодирование изображения символов для NLQ-шрифта. Пример распечатки - текст настоящего сборника.

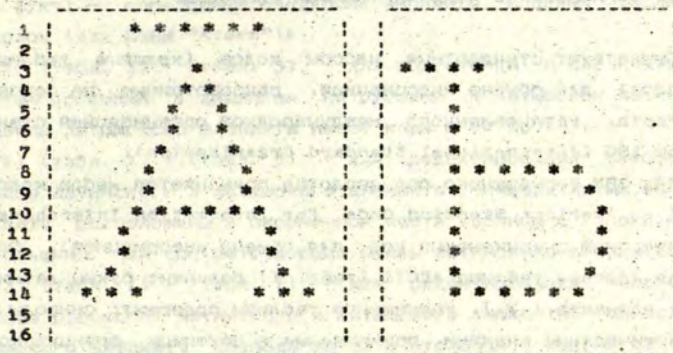
Кропотливая работа по кодированию изображений символов проводится с помощью специально разработанных программ, позволяющих в диалоговом режиме изменять начертания символов на экране в матрице 8x8, 8x11 или 16x23, а затем получать закодированную таблицу в специальной области памяти или файле на диске, предназначенного для загрузки информации в ОЗУ принтера.

Рис. 3. Кодирование изображения символа для принтера.

	-1-2-3-4-5-6-7-8-		
1		00000000	00
2		00000000	00
3		00000010	02
4	* * * *	10011100	9C
5	* * * *	00100000	20
6	* * * *	10001010	8A
7	* * * *	00100000	20
8	* * * *	10001010	8A
9	* * * *	00100000	20
10	* * * *	10001010	8A
11		00000100	04
-----			
1		00000000	00
2		00000000	00
3		00000000	00
4	* * * * *	01111110	7E
5	* * * * *	00000001	01
6	* * * * *	01000100	44
7	* * * * *	00000001	01
8	* * * * *	01000100	44
9	* * * * *	00000001	01
10	* * * * *	01000100	44
11		00111000	38
-----			
1		00000000	00
2		00000100	04
3		00001010	0A
4		00000000	00
5		00001010	0A
6		00000000	00
7		00001010	0A
8		00000000	00
9	* * * * *	00111110	3E
10		00000000	00
11	*	00100000	20

-1-2-3-4-5-6-7-8-

Рис. 4. Кодирование изображения для шрифта MLQ.



0000080809290808000000  
 000000012C0000C210000000  
 000003484040404048430C000  
 0040C04000000000040C0400

020203020000000000000000  
 00000F010101010100000000  
 00000F000000000008700000  
 00000C04040404040480000000

BE 80 17

FF 80 17

00 00 00 00 00 04 00 00  
 00 0C 00 30 80 44 01 80  
 82 40 0C 00 90 40 20 00  
 90 40 0C 00 82 40 01 80  
 80 44 00 00 30 00 0C 00 00  
 00 04 00 00 00 00

00 00 20 00 00 00 20 00  
 00 00 3F FC 00 00 21 04  
 00 00 01 04 00 00 01 04  
 00 00 01 04 00 00 01 04  
 00 88 00 70 00 00 00 00  
 00 00 00 00 00 00

## 2. Формирование кодовой таблицы символов

Существуют стандартные наборы кодов (кодовые таблицы символов) для обмена информацией, разработанные на основе стандарта, установленного международной организацией стандартов ISO (International Standard Organization).

Для ЭВМ зарубежного производства применяется набор кодов ASCII - American Standard Code For Information Interchange (Стандартный американский код для обмена информацией). Основная кодовая таблица ASCII (табл. 1) содержит буквы латинского алфавита [ 2 ]. Расширение таблицы содержит: символы с диакритическими знаками, применяемые в немецком, французском и других языках, алфавиты которых построены на базе латинского; символы табличной графики, специальные символы и знаки.

Для отечественных ЭВМ стандартные коды определены ГОСТом 13052-74, который устанавливает два набора алфавитно-цифровых и управляющих символов 7-битного кода - КОИ-7Н(0) и КОИ-7Н(1). Набор КОИ-7Н(0) включает строчные и прописные буквы латинского алфавита (табл. 1), набор КОИ-7Н(1) - строчные и прописные буквы русского алфавита (табл. 2). Оба набора содержат также цифры, математические и разделительные знаки и символы для управления передачей данных. Набор кодов КОИ-8 по существу является объединением наборов КОИ-7Н(0) и КОИ-7Н(1).

ГОСТ 19767-74 определяет набор латинских и русских прописных букв, не содержащий строчных букв, т.е. представляет собой совмещение наборов КОИ-7Н(0) и КОИ-7Н(1) с исключением строчных букв.

Наборы кодов, приведенные в упомянутых стандартах, не содержат букв национальных алфавитов. Это относится также к вариантам переменной части кодовой таблицы (см. табл. 9, символы табличной графики и другие специальные знаки не показаны), предложенным в [ 1,2 ] в качестве основы стандарта для ПЭВМ.

Для одновременной обработки информации, включающей символы трех алфавитов - русского (по 32 прописных и строчных

буквы), латинского (по 26 прописных и строчных букв) и латышского (дополнительно к латинскому по 11 прописных и строчных букв), - опробованы следующие варианты кодовых таблиц символов (для ПЭВМ "Acorn"):

1) (табл. 1) + (табл. 3) - для подготовки и распечатки текстов программ с диалогом на русском и латышском языках (символы латышского алфавита имеют коды с 80 по 9F);

2) (табл. 1) + (табл. 5) - для редактирования текстов на базе латинского и латышского алфавита (символы латышского алфавита расположены в переменной части таблицы с кодами на 7F большими, чем соответствующие буквы латинского алфавита);

3) (табл. 1) + (табл. 7) - для редактирования текстов на базе русского, латинского и латышского алфавитов (символы латышского алфавита располагаются в столбцах с кодами от 40 до 9F).

Выбор того или иного варианта кодовой таблицы определяется как по функциональному признаку, так и с учетом ограничений, вносимых применяемым и разрабатываемым программным обеспечением, а также потребностью управления (с помощью контрольных кодов) работой периферийных устройств, например, принтером.

Привязка к клавиатуре, маркированной для латинских букв в стандарте QWERTY, а для русских букв в стандарте ЦУКЕН, для дополнительных к латинскому букв латышского алфавита, осуществлена к соответствующим клавишам без диакритических знаков при одновременном нажатии специальной клавиши <CTRL>.

Аналогичные проблемы решаются также и для других ПЭВМ. Например, для "ВЭФ-микро" кодовая таблица символов является объединением (табл. 1) + (табл. 6), а для ПЭВМ "Robotron 1715" - (табл. 4) + (табл. 2).

Для введения стандартизации программного обеспечения с возможностью диалога на латышском языке необходима единая кодовая таблица символов для различных типов ПЭВМ. С учетом [ 1 ] предлагается следующий вариант, включающий в переменной части кодовой таблицы одновременно символы русского и латышского алфавитов:

Таблицы 1-9. Варианты кодовых таблиц символов.

Табл. 1.

	0	1	2	3	4	5	6	7
0		0	0	F	р			
1	!	1	A	Q	a	q		
2	"	2	B	R	b	r		
3	#	3	C	S	c	s		
4	\$	4	D	T	d	t		
5	%	5	E	U	e	u		
6	&	6	F	V	f	v		
7	'	7	G	W	g	w		
8	(	8	H	X	h	x		
9	)	9	I	Y	i	y		
A	*	A	J	Z	j	z		
B	+	B	K	[	k	{		
C	<	C	L	\	l			
D	=	D	M	]	m	}		
E	>	E	N	^	n	~		
F	/	F	O	_	o			
	?							

Табл. 2.

	8	9	A	B	C	D	E	F
0		0	ы	п	ю	п		
1	!	1	я	л	р	я		
2	"	2	б	р	б	р		
3	#	3	ц	с	ц	с		
4	\$	4	д	т	д	т		
5	%	5	е	у	е	у		
6	&	6	ф	ж	ф	ж		
7	'	7	г	в	г	в		
8	(	8	х	ь	х	ь		
9	)	9	и	ы	и	ы		
A	*	A	й	з	й	з		
B	+	B	к	ш	к	ш		
C	<	C	л	щ	л	щ		
D	=	D	м	н	м	н		
E	>	E	н	ч	н	ч		
F	/	F	о	ь	о	ь		
	?							

Табл. 3.

	8	9	A	B	C	D	E	F
0	л	а	о	ю	п	ю	п	
1	с	э	!"	1	а	р	я	
2	е	"	1	2	б	р	б	
3	г	з	#	3	ц	с	ц	
4	к	к	\$	4	д	т	д	
5	л	л	%	5	е	у	е	
6	л	л	&	6	ф	ж	ф	
7	л	л	'	7	г	в	г	
8	л	л	(	8	х	ь	х	
9	л	л	)	9	и	ы	и	
A	л	л	*	A	й	з	й	
B	л	л	+	B	к	ш	к	
C	л	л	<	C	л	щ	л	
D	л	л	=	D	м	н	м	
E	л	л	>	E	н	ч	н	
F	л	л	/	F	о	ь	о	
	л	л	?					

Табл. 4.

	0	1	2	3	4	5	6	7
0	н	0	0	р	р			
1	л	с	!"	1	а	р	я	
2	с	п	"	2	б	р	б	
3	с	а	#	3	ц	с	ц	
4	с	е	\$	4	д	т	д	
5	с	е	%	5	е	у	е	
6	с	к	&	6	ф	ж	ф	
7	с	к	'	7	г	в	г	
8	с	к	(	8	х	ь	х	
9	с	к	)	9	и	ы	и	
A	с	к	*	A	й	з	й	
B	с	к	+	B	к	ш	к	
C	с	к	<	C	л	щ	л	
D	с	к	=	D	м	н	м	
E	с	к	>	E	н	ч	н	
F	с	к	/	F	о	ь	о	
	с	к	?					

Табл. 5.

	8	9	A	B	C	D	E	F
0		0	е					
1	!"	1	л	а				
2	"	2	с	с	с	с	с	
3	#	3	с	с	с	с	с	
4	\$	4	е	е	е	е	е	
5	%	5	е	е	е	е	е	
6	&	6	г	г	г	г	г	
7	'	7	г	г	г	г	г	
8	(	8	и	и	и	и	и	
9	)	9	и	и	и	и	и	
A	*	A	и	и	и	и	и	
B	+	B	к	к	к	к	к	
C	<	C	л	л	л	л	л	
D	=	D	л	л	л	л	л	
E	>	E	н	н	н	н	н	
F	/	F	н	н	н	н	н	
	?							

Табл. 6.

	8	9	A	B	C	D	E	F
0		0		ю	п	ю	п	
1	л	а		я	л	я	л	
2	с	с	с	а	р	с	а	
3	с	с	с	в	ц	с	в	
4	с	с	с	в	д	т	с	
5	с	с	с	в	е	у	с	
6	с	с	с	в	ф	ж	с	
7	с	с	с	в	г	в	с	
8	с	с	с	в	г	в	с	
9	с	с	с	в	х	ь	с	
A	с	с	с	в	и	ы	с	
B	с	с	с	в	й	з	с	
C	с	с	с	в	к	ш	с	
D	с	с	с	в	л	щ	с	
E	с	с	с	в	м	н	с	
F	с	с	с	в	н	ч	с	
	с	с	с	в	о	ь	с	

Табл. 7.

	8	9	А	В	С	Д	Е	Ф
0			Ю	П	Ю	П	Я	
1	А	Р	а	р	А	Р	Я	
2	С	с	б	р	В	Р		
3			ц	с	Ц	С		
4			д	т	Д	Т		
5			е	у	Е	У		
6	Е	е	Ф	Ж	Ф	Ж		
7	Г	г	В	Г	Ь			
8	І	і	х	ь	Х	Ь		
9	К	к	ы	й	Ы	Й		
А	Л	л	я	з	Я	З		
В	У	у	к	ш	К	Ш		
С	С	с	л	э	Л	Э		
Д	У	у	а	м	У	М		
Е	2	2	о	ч	О	Ч		
Ф			о	ь	О	Ь		

Табл. 8.

	8	9	А	Р	С	Д	Е	Ф
0			А	Р	в	Д	Е	
1			В	С	т	С	Е	
2			Э	Т	в	Т		
3			Г	У	Г	У		
4			Д	Е	Ж	Ф		
5	А	С	Ж	Ц	Ж	Ц	а	с
6	С	с	Э	Ч	э	ч	е	с
7	О	И	Ш	Я	ш	я	к	е
8	І	Я	Щ	Я	щ	я	і	к
9	К	Л	ь	к	ь	л	ь	к
А	К	Л	ь	к	ь	л	ь	к
В	Л	М	ь	л	ь	м	ь	л
С	М	Н	ь	м	ь	н	ь	м
Д	С	Н	ь	с	н	ь	д	о
Е	О	Д	о	д	о	я	о	д
Ф	2	П	я	п	я	я	2	п

Табл. 9.

	8	9	А	В	С	Д	Е	Ф
0			А	Р	а	р	Е	
1			В	С	т	с	е	
2			В	Т	в	т		
3			Г	У	г	у		
4			Д	Ф	д	ф		
5			Е	Х	е	х		
6			Ж	Ц	ж	ц		
7			Э	Ч	э	ч		
8			И	Ш	и	ш		
9			И	Ш	и	ш		
А			К	ь	к	ь		
В			Л	ь	л	ь		
С			М	ь	м	ь		
Д			Н	э	н	э		
Е			О	ю	о	ю		
Ф			П	я	п	я		

Табл. 10.

	8	9	А	В	С	Д	Е	Ф
0			А	К	W	а	к	w
1			Л	Л	X	л	л	x
2			Д	Л	Y	д	л	y
3			С	Н	З	с	н	z
4			С	Н	2	с	н	2
5			Д	Н		д	н	
6			Е	О		е	о	
7			Е	Р		е	р	
8			Р	Q		р	q	
9			Q	R		q	r	
А			О	S		о	s	
В			Н	S		н	s	
С			І	T		і	t	
Д			І	U		і	u	
Е			Ј	С		ј	с	
Ф			К	V		к	v	

- (табл. 1) + (табл. 8) или альтернативный вариант, аналогичный [ 1 ].

Для вновь разрабатываемых ПЭВМ или нового поколения программного обеспечения, по-видимому, необходимо предусмотреть возможность одновременного использования наряду с основной частью кодовой таблицы (табл. 1) нескольких вариантов переменной части кодовой таблицы, содержащей соответственно:

- упорядоченный набор букв русского алфавита (табл. 9) [ 1, 2 ]:

- упорядоченный набор букв латышского алфавита (табл. 10); отметим, что аналогично может быть решена проблема кодовой таблицы для других национальных алфавитов: для литовского и эстонского к латинскому алфавиту добавляется соответственно по 10 и по 4 прописные и строчные буквы, а для украинского и белорусского к русскому алфавиту добавляется соответственно по 3 и по 1 прописной и строчной букве:

- упорядоченные наборы букв других алфавитов;
- наборы специальных знаков и символов.

### 3. Перевод файлов с различными кодовыми таблицами

Поскольку пока отсутствует стандарт на кодовую таблицу символов с национальными алфавитами, возникает необходимость в перекодировке разнообразных файлов для обеспечения обмена информацией между различными ПЭВМ.

С этой целью в виде программы [ 3 ] реализован универсальный алгоритм для трансляции файлов из одного набора кодов в другой. Алгоритм "переводит" коды символов исходного файла в соответствии с созданным "словарем". Словарь - это таблица, состоящая из FF байтовых элементов. При трансляции символ из входного файла, имеющий i-й код, заменяется на символ, имеющий код, находящийся в i-м байте словаря. Оттранслированные коды записываются в выходной файл. Используя разные словари, можно оттранслировать один и тот же файл для ПЭВМ с различными наборами кодов.

## Выводы

1. Предложено несколько вариантов кодовых таблиц символов для ПЭВМ, включающих одновременно символы латинского, русского и латышского алфавитов. Предложен вариант привязки дополнительных символов к клавиатуре. Аналогично могут быть построены кодовые таблицы для других национальных алфавитов (литовского, эстонского, украинского, белорусского и др.)

2. Целесообразна разработка стандарта кодовой таблицы символов для одновременного использования в перспективных ПЭВМ и программном обеспечении нескольких альтернативных вариантов переменной части кодовой таблицы.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Брябрин В.М., Лендау И.Я., Неменман М.Е. О системе кодирования для персональных ЭВМ // Микропроцессорные средства и системы. 1986. # 4. С. 61-63.
2. Брябрин В.М. Программное обеспечение персональных ЭВМ. М.: Наука. 1988. 272 с.
3. Елкина О.М., Курмис А.Г., Овиш И.Я., Павлов С.И., Паякено А.Р., Устинов Н.Н., Эглайс М.О. Комплекс программы вывода символов русского и латышского алфавитов для обработки текстовой информации на микроЭВМ "Асогол". Рига. 1988. 1 с. Информ. листок. Выставка "Компьютер-88". Рижский дом техники НТО / ЛатНИИ НТИ: # 703.

УДК 681.322

Л. Л. Бульгин, Л. П. Климанс, М. О. Эглайс  
Вычислительный центр при ЛГУ им. П. Стучки

### ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ДЛЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МИКРОЭВМ "ЭЛЕКТРОНИКА БК-0010"

Создание программного обеспечения (ПО) микроЭВМ "Электроника БК-0010" (далее "БК"), широко используемой в настоящее время в учебных заведениях (школах, вузах, техникумах), сопряжено с рядом трудностей:

- а) отсутствие удобного текстового редактора для создания текстов программы;
- б) отсутствие возможности напечатать текст программы на языке FOCAL и результаты работы программы;
- в) неудобство работы с клавиатурой;
- г) отсутствие возможности создания качественного ПО на языке АССЕМБЛЕР.

В настоящей работе описаны возможные пути решения выше-названных проблем с использованием инструментальных ЭВМ ("Асorn", "ДВК"). Инструментальная ЭВМ и "БК" соединяются между собой параллельным интерфейсом, обмен информацией осуществляется соответствующими программами.

#### 1. Создание текстов программ на инструментальной ЭВМ


При наличии микроЭВМ "Асorn" и "БК", тексты FOCAL-программ можно ввести на микроЭВМ "Асorn" при помощи текстового редактора и записать текст программы в файл на диске. Потом тексты программ можно пересылать на "БК" для запуска, если нужно, для отладки и редактирования.

Порт пользователя на "БК" подключается к выходу "printer" на "Асorn" при помощи кабеля (табл. 1).

Таблица 1. Схема подключения контактов порта пользователя "БК" и выхода "printer" "Acorn"

ВЫС PRINTER PORT		ПОРТ ПОЛЬЗОВАТЕЛЯ "БК"
# контакта	название сигнала	
1	STROBE	A30
2	GROUND	A18, A19, B18, B19
3	D0	B24
4	GROUND	A18, A19, B18, B19
5	D1	A24
6	GROUND	A18, A19, B18, B19
7	D2	B23
8	GROUND	A18, A19, B18, B19
9	D3	B17
10	GROUND	A18, A19, B18, B19
11	D4	B20
12	GROUND	A18, A19, B18, B19
13	D5	A20
14	GROUND	A18, A19, B18, B19
15	D6	B22
16	GROUND	A18, A19, B18, B19
17	D7	A23
18	GROUND	A18, A19, B18, B19
19	ACK	A25
20	GROUND	A18, A19, B18, B19
21	-	-
22	GROUND	A18, A19, B18, B19
23	-	-
24	GROUND	A18, A19, B18, B19
25	-	-
26	-	-

Программная часть интерфейса состоит из программ для "БК" и "Acorn".

Программа приема на "БК" (см. рис. 1) при запуске модифицирует адрес первоначальной обработки ЭМТ. т.е. по адресу 30 засылается адрес 1012 из программы приема. С этого момента обработка каждого ЭМТ начинается передачей управления на программу приема. Эта программа меняет режим работы операционной системы при нажатии на клавиатуре графического символа , устанавливая значение переключателя SW в программе равным 1. Это является признаком перевода операционной системы в режим, в котором клавиатура "отключена" (коды с клавиатуры игнорируются). интерпретатор ФОКАЛ обрабатывает информацию, поступающую с порта пользователя. При работающей на "Acorn" программе пересылки, это текст ФОКАЛ-программы. При получении кода "0" программа приема изменяет значение SW на 0, и "БК" переводится в первоначальный режим обработки ЭМТ до следующего нажатия символа. Пользователь "БК" может вводить команды с клавиатуры для запуска, редактирования программы или других целей.

Программа на "Acorn" выводит на экране короткую инструкцию, запрашивает название файла пересылаемой FOCAL-программы на диске. Далее программа на "Acorn" загружает указанную пользователем FOCAL-программу, потом осуществляет пересылку программы на "БК", т.е. вывод на выход "printer". В конце текста выводится код "0".

Текст FOCAL-программы можно ввести на "Acorn", используя редактор текстов "view" или какой-то другой, и записать в файл на диске. В этом случае строки FOCAL-программы, состоящие из номера и одного или нескольких операторов, вводятся как строки текста. Для удобства работы длина строки FOCAL-программы не должна превышать длину строки текста при распечатке. Графические и другие специальные символы, используемые в FOCAL-программе на "БК" не могут быть введены текстовым редактором. Операторы, содержащие такие символы, необходимо редактировать непосредственно на "БК". Если эти символы предназначены для вывода на экран, то можно пользоваться функцией FCHR. Например, оператор:

10.10 T "Г"

Рис. 1. Текст программы приема на "БК"

	Текст программы	Адрес	Машинные команды		
	MOV #1012, #30	:1000	012737	001012	000030
	JMP #120000	:1006	000137	120000	
EM:	MOV R5, -(SP)	:1012	010546		
	MOV 2(SP), R5	:1014	016605	000002	
	MOV -(R5), R5	:1020	014505		
	BIC #177400, R5	:1022	042705	177400	
	CMP #6, R5	:1026	022705	000006	
	BEQ DRUK	:1032	001420		
	CMP #16, R5	:1034	022705	000016	
	BNE NEXT	:1040	001006		
	CMPB #242, R0	:1042	122700	000242	
	BNE NEXT	:1046	001003		
	INC SW	:1050	005267	000104	
	BR NOEMT	:1054	000403		
NEXT:	CMP #14, R5	:1056	022705	000014	
	BNE MBACK	:1062	001002		
NOEMT:	JMP #100134	:1064	000137	100134	
MBACK:	JMP #100126	:1070	000137	100126	
DRUK:	TST SW	:1074	005767	000060	
	BEQ MBACK	:1100	001773		
	JSR PC, GET	:1102	004767	000002	
	BR NOEMT	:1106	000766		
GET:	MOV R2, -(SP)	:1110	010246		
	MOV #177714, R2	:1112	012702	177714	
	MOV #40000, (R2)	:1116	012712	040000	
	CLR (R2)	:1122	005012		
T1:	TST (R2)	:1124	005712		
	BPL T1	:1126	100376		
T2:	MOV (R2), R0	:1130	011200		
	MOV #40000, (R2)	:1132	012712	040000	
	BMI T2	:1136	100774		
	COM R0	:1140	005100		
	BIC #177400, R0	:1142	042700	177400	
	BNE GTEND	:1146	001002		
	CLR SW	:1150	005067	000004	
GTEND:	MOV (SP)+, R2	:1154	012602		
	RTS PC	:1156	000207		
SW:	.WORD 0	:1160	000000		

можно записать в форме:

10.10 X FCRH(170,184,163)

При отладке программы на "БК" исправления желательно запомнить и внести их в файл текста программы на "Acorn".

Чтобы запустить на "БК" подготовленную на "Acorn" FOCAL-программу, требуется следующее:

1. Порт пользователя на "БК" подключается к выходу "printer" на "Acorn".
2. На "БК" в режиме монитора с магнитной ленты запускается программа в машинных кодах для приема информации от "Acorn". Запускаются следующие команды:

P M	<ВВОД>
M	<ВВОД>
<имя>	<ВВОД>
S	<ВВОД>

Потом вводится графический символ ♡. Ввод этого символа переводит операционную систему в режим, в котором интерпретатор FOCAL обрабатывает информацию, поступающую с порта пользователя до получения кода "0".

3. На "Acorn" запускается программа, которая после показанной на экране инструкции выводит запрос наименования файла текста программы. Далее программа посылает текст программы на выход "printer". В конце текста программы посылается код "0", который на "БК" восстанавливает первоначальный режим операционной системы. Далее пользователь может ввести с клавиатуры "БК" команды для работы с полученной от "Acorn" FOCAL-программой.

## 2. Распечатка результатов работы и текстов программ на языке FOCAL

При наличии микроЭВМ "Acorn" и "БК" тексты FOCAL-программ могут быть введены с магнитной ленты на "БК" и переданы на "Acorn" для распечатки, а также редактирования текстовым редактором.

Интерфейс может быть использован для подготовки документации текстов FOCAL-программ.

Порт пользователя на "БК" подключается к порту пользова-

теля на "Асорт" при помощи кабеля (см. табл. 2).

Система сопряжения включает в себя также программную часть. Загруженная на "БК" программа в машинных кодах для пересылки (см. рис. 2) устанавливает режим операционной системы, где вся информация, поступающая на драйвер ТВ-приемника, передается также на порт пользователя. Команда "w" задает вывод текста FOCAL-программы на экран, тем самым текст программы передается на порт пользователя "БК".

Программа приема на "Асорт" выводит на экран короткую инструкцию. Далее запрашивается у пользователя режим пересылки кодов, т.е. запоминается в файле все коды или только те, для которых выполняется условие:

код>31 или код=10 или код=13

Кроме того, программа запрашивает наименование файла, в который записывается текст FOCAL-программы.

Далее программа вводит с порта пользователя информацию, пересылаемую из "БК", запоминает в памяти текст FOCAL-программы, т.е. часть информации, которая заключена между строками "\*" (команда вывода текста программы на экран "БК") и строкой "\*" (строка на экране после текста программы). В зависимости от режима пересылки в файл заносятся либо все коды, либо те, для которых выполняются условия

код>31 или код=10 или код=13

Длина текста FOCAL-программы не должна быть больше 20000 байт, в противном случае выдается сообщение об ошибке. Текст FOCAL-программы записывается в файл на диске под именем, заданным пользователем. Если в тексте FOCAL-программы использованы графические символы или режим редактирования, то пользователь должен отредактировать текст, например, заменить такие символы на пробелы, чтобы в отпечатанном листинге их нарисовать вручную.

Для пересылки FOCAL-программы с "БК" на "Асорт" требуется следующее:

1. Порт пользователя на "БК" подключается к порту пользователя на "Асорт".
2. На "Асорт" запускается программа приема.
3. Нажатием клавиши <пробел> пропускается страница текста инструкции. После запроса вводится режим пересылки

Рис. 2. Текст программы пересылки с "БК"

	Текст программы	Адрес	Машинные команды
	MOV #1012, @#30	:1000	012737 001012 000030
	JMP @#120000	:1006	000137 120000
EM:	MOV R5, -(SP)	:1012	010546
	MOV 2(SP), R5	:1014	016605 000002
	MOV -(R5), R5	:1020	014505
	MOV #20000, R2	:1022	012702 020700
	MOV #177714, R3	:1026	012703 177714
	BIC #177400, R5	:1032	042705 177400
	CMP #16, R5	:1036	022705 000016
	BEQ DRUK	:1042	001420
	CMP #14, R5	:1044	022705 000014
	BNE MBACK	:1050	001002
	JMP @#100134	:1052	000137 100134
MBACK:	JMP @#100126	:1056	000137 100126
DRUK:	JSR PC, SEN	:1062	004767 000004
	JMP @#100126	:1066	000137 100126
SEN:	CLR (R3)	:1072	005013
SEND:	BIT R2, (R3)	:1074	030213
	BEQ SEND	:1076	001776
	MOVB R0, SENWO	:1100	110067 000024
	BIS R2, SENWO	:1104	050267 000020
	MOV SENWO, (R3)	:1110	016713 000016
	BIC R2, SENWO	:1114	040267 000012
	MOV SENWO, (R3)	:1120	016713 000006
ZPIN:	BIT R2, (R3)	:1124	030213
	BEQ ZPIN	:1126	001776
	RTS PC	:1130	000207
SENWO:	.WORD 0	:1132	000000

символов нажатием "D" или "N", также вводится название, которое будет присвоено файлу текста, пересылаемой из "БК" FOCAL-программы на диске "Асорт". Далее программа на "Асорт" переходит в состояние ожидания и приема информации от "БК".

4. На "БК" в режиме монитора с магнитной ленты загружается программа пересылки, для чего на "БК" выполняются следующие команды:

```
Р М <ВВОД>
М <ВВОД>
<имя> <ВВОД>
S <ВВОД>
```

5. На "БК" с магнитной ленты загружается FOCAL-программа командой:

```
L G <имя> <ВВОД>.
```

где <имя> - имя FOCAL-программы на магнитной ленте.

6. На "БК" вводится команда

```
*W <ВВОД>
```

7. Для пересылки следующей FOCAL-программы из "БК" на "Асорт" необходимо выполнить действия по пунктам 3, 5, 6.

### 3. Создание и пересылка оттранслированных на "ДВК-2М" программ на языке МАКРОАСSEMBЛЕРА

Система позволяет созданную на базе МАКРОАСSEMBЛЕРА "ДВК-2М" программу переслать на "БК".

"БК" подключается через порт пользователя с параллельным интерфейсом печатающего устройства "ДВК-2М". Соединения микроЭВМ представлено в табл. 3.

Система сопряжения включает также программную часть.

На "ДВК-2М" к созданной с помощью текстовых редакторов программе на языке МАКРОАСSEMBЛЕРА добавляется фрагмент программы, который при запуске оттранслированной программы пересылает на "БК" созданную программу в машинных кодах. Программа пересылки на "ДВК-2М" представлена на рис. 3 и осуществляет следующие действия:

- пересылает два байта длины созданной для "БК" программы;
- пересылает адрес размещения программы в памяти "БК";
- пересылает программу в машинных кодах.

Таблица 2. Соединение "БК" и "ДВК-2М"

"БК" (ПП)	"ДВК-2М" (ХТ2)	Назначение сигнала
В 24	59	D0 - Данные 0
А 24	60	D1 - Данные 1
В 23	8	D2 - Данные 2
В 17	5	D3 - Данные 3
В 20	4	D4 - Данные 4
А 20	3	D5 - Данные 5
В 32	1	D6 - Данные 6
А 23	6	D7 - Данные 7
В 29	15	СТР-ПН-Строб приемнику информации
А 26	36	ЭП-ПН-Сигнал запроса приемника на выдачу информации
А 25	37	ГП-ПН-Готовность приемника
А 11, В 11	55, 52	Земля

Таблица 3. Схема подключения контактов портов  
пользователя микросВМ "Электроника БК-0010"  
и "Acorn"

VBC USER PORT		ПОРТ ПОЛЬЗОВАТЕЛЯ "БК"
# контакта	название сигнала	
1	2	3
1	+5V	-
2	CB1	B26
3	+5V	-
4	CB2	A29
5	GND	A18, B18, A19, B19
6	PB0	A16
7	GND	A18, B18, A19, B19
8	PB1	A13
9	GND	A18, B18, A19, B19
10	PB2	B12
11	GND	A18, B18, A19, B19
12	PB3	B10
13	GND	A18, B18, A19, B19
14	PB4	B5
15	GND	A18, B18, A19, B19
16	PB5	B7
17	GND	A18, A19, B19
18	PB6	B6
19	GND	A18, B18, A19, B19
20	PB7	A7

Рис. 3. Текст программы на "ДВК-2М"

```
-----  
BEGIN:          .WORD 1000  
                |  
                | Текст пересылаемой программы на  
                | языке Макроссамблер  
                |  
MMM1:          .BYTE  
                .EVEN  
MMM2:          .WORD MMM1-BEGIN-2  
MMM3:          .MOVB MMM2,RO  
                JSR PC,MMM5  
                .MOVB MMM2+1,RO  
                JSR PC,MMM5  
                .MOV #BEGIN,R1  
MMM4:          .MOVB (R1)+,RO  
                JSR PC,MMM5  
                .CMP R1,MMM1  
                .BLE MMM3  
                .TRAP  
MMM5:          .TSTB @#177514  
                .BPL MMM5  
                .MOVB RO,MMM47  
                .MOV MMM7,@#177516  
MMM6:          .TSTB @#177514  
                .BMI MMM46  
                .RTS PC  
MMM47:         .WORD 0  
                .END MMM3  
-----
```

Программе в машинных кодах должна быть перемещаемой.

Программа приема на "БК" представлена на рис. 4 и осуществляет следующие действия:

- считывает два байта длины программы;
- считывает адрес загрузки программы в память "БК";
- считывает код программы и записывает в память с заданного адреса.

Текст создаваемой для "БК" программы на МАКРОАССЕМБЛЕРе можно ввести на "ДВК-2М" с помощью текстового редактора EDIT или SCREEN.

Для пересылки созданной на "ДВК-2М" программы в "БК" требуется следующее:

1. Порт пользователя на "БК" подключается к параллельно-интерфейсу печатающего устройства "ДВК-2М".
2. На "БК" в режиме монитора с магнитной ленты загружается программа приема (см. рис. 4), для чего выполняются следующие команды

P M	<ВВОД>
M	<ВВОД>
<Имя>	<ВВОД>
S	<ВВОД>

Программа должна быть загружена в свободную область памяти "БК", не занимаемую пересылаемой программой.

3. К созданному тексту программы на "ДВК-2М" добавляется фрагмент текста программы, представленный на рис. 3. Программа в таком виде транслируется и запускается. Работа ведется стандартными системными средствами.

4. После окончания работы программы пересылки на "ДВК-2М" (о чем свидетельствует появление точки "." в новой строке на экране дисплея), созданная программа в машинных кодах переслана на "БК", и с ней можно проводить дальнейшие действия (запуск, запись на магнитную ленту и т.д.).

#### Заключение

Описанные системы могут получить более широкое применение, в частности:

1. В качестве инструментальной ЭВМ для создания текстов

Рис. 4. Текст программы приема на "БК"

```
-----  
JSP PC, PROG2  
MOV R3, R2  
JSR PC, PROG2  
MOV R0, R1  
ADD R1, R2  
LOOP1: JSR PC, PROG1  
        MOVB R0, (R1)+  
        CMP R1, R2  
        BLT LOOP1  
        RTS PC  
PROG1:  MOV #50000, @#177718  
        MOV R1, -(SP)  
LOOP2:  MOV @#177718, R0  
        BIT #10000, R0  
        BEQ LOOP2  
        MOVB @#177718, R0  
        MOV #40000, @#177718  
LOOP3:  MOV @#177718, R1  
        BIT #20000, R1  
        BNE LOOP3  
        MOV (SP)+, R1  
        MOV #50000, @#177718  
        RTS PC  
PROG2:  JSR PC, PROG1  
        MOVB R0, LSB  
        JSR PC, PROG1  
        MOVB R0, MSB  
        MOV LSB, R0  
        RTS PC  
LSB:   .BYTE 0  
MSB:   .BYTE 0  
-----
```

программы можно использовать любую ЭВМ с текстовым редактором и параллельным интерфейсом, соответственно модифицировать программу приема в "БК".

2. Фрагменты или полный текст программы на инструментальной ЭВМ могут быть созданы и другими средствами (например, графическим редактором [ 1 ]). Фрагменты текстов программы загружаются на "БК", производя слияние с существующей программой, следует соблюдать нумерацию строк.

3. Система пересылки текстов программ позволяет управлять работой "БК" с использованием инструментальной ЭВМ также во время выполнения программы путем пересылки из инструментальной ЭВМ необходимой для работы программы информации, загрузки и запуска следующей программы и т.д.

4. Использование параллельного интерфейса помогает пересылать с "ДБК-2М" оттранслированную на машинном языке программу длиной в 16 К за 3-4 секунды, что удобно при отладке больших программ.

#### Л И Т Е Р А Т У Р А

1. Раудио А.Ф. Формирование изображения в обучающих программах (на примере курса астрономии) // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 154-165.

2. Математическое обеспечение микроЭВМ "Электроника БК-0010". Язык Фокал. Руководство пользователя. Том 1. Книга 1. 1985. 15 с.

3. Математическое обеспечение микроЭВМ "Электроника БК-0010". Язык Фокал. Руководство системного программиста. Том 1. Книга 2., 1985. 78 с.

Межвузовский сборник научных трудов  
ЗВМ В ОБРАЗОВАНИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
1988. Рига: ЛГУ им. П. Стучки. С. 51-55

УДК 37.01:007+371.3+681.3.06

И. Я. Озиш

Вычислительный центр при ЛГУ им. П. Стучки

ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ  
МИКРОЭВМ "ДВК" В УЧЕБНОМ ПРОЦЕССЕ

Основная цель школьного курса "Основы информатики и вычислительной техники" состоит в формировании представлений об основных правилах и методах ринлиниции решений задач на ЭВМ и умений пользоваться микрокомпьютером для решения всевозможных задач. Выпускники современной школы должны также иметь представление об ЭВМ как средстве автоматической обработки информации и быть знакомы с основными принципами работы операционной системы. Они должны иметь представление о составе программного обеспечения ЭВМ и уметь пользоваться пакетами прикладных программ [ 1 ].

В настоящее время школа не имеет единой материально-технической базы для обеспечения "машинного варианта" обучения использованию средств вычислительной техники, поэтому в условиях ее дефицита для достижения основных целей курса необходимо базироваться на доступных школьнику средствах.

В школах Латвийской ССР установлено около 50 комплектов учебной вычислительной техники КУВТ-86. Комплект КУВТ-86 состоит из диалогового вычислительного комплекса ("ДВК") "Электроника MC 0501.06" и 12 бытовых микроЭВМ "Электроника MC 0510" ("Электроника ВК-0010", в дальнейшем "ВК"), объединенных локальной сетью.

В комплекте КУВТ-86 в качестве центральной ЭВМ используется микроЭВМ "ДВК", обеспечивающая работу локальной сети и хранение программ на гибких минидисках. Локальная сеть поддерживает следующие режимы работы [ 2,3 ]:

- чтение программы с диска и передачу на отдельную или

несколько "БК":

- прием программы от каждой отдельной "БК" и сохранение на диске;

- режим взаимодействия и контроля над программой, находящейся в памяти "БК".

Однако возможности использования микроЭВМ "ДВК", предназначенной для решения инженерно-вычислительных задач и задач по обработке и накоплению информации, намного шире. Работа микроЭВМ "ДВК" осуществляется под управлением операционной системы ОС ДВК. Программное обеспечение включает операционную систему, системы программирования (БЕЙСИК, МАКРО, ФОРТРАН, ПАСКАЛЬ, МОДУЛ-2), системы редактирования и отладки. Операционная система ОС ДВК обеспечивает обслуживание широкого набора внешних устройств (магнитных лент, съемных гибких минидисков, различных печатающих устройств). Большинство этих возможностей преподаватель может использовать в своей повседневной работе для создания различных файлов справочной информации, для подготовки, редактирования и распечатки документов, для решения вариантов контрольных работ, а также на уроках информатики. Использование возможностей микроЭВМ "ДВК" в повседневной работе чрезвычайно полезно: преподавателю важно самому убедиться и оценить преимущества, которые предоставляет микроЭВМ, хотя бы в такой ограниченной области применения, как создание, редактирование и печать документов, накопление и обработка файлов справочной информации. Для сознательного освоения курса информатики ученикам необходимо продемонстрировать примеры практического использования ЭВМ в повседневной работе при решении задач физики, химии, математики, обработки информации и др.

Ознакомление с возможностями использования "ДВК" для решения задач вычислительного характера, освоение языка программирования БЕЙСИК и знакомство с работой операционной системы необходимо проводить в комплексе. После предварительной теоретической подготовки (необходимый минимум по языку программирования и операционной системе) желательно выбрать конкретную задачу для практического решения на ЭВМ. Школьный курс математики подготавливает учащихся к применению численных методов решения задач, знакомит с графическими методами

решения алгебраических уравнений. Можно рекомендовать, приближенное решение уравнений методом половинного деления или методом хорд, решение систем линейных уравнений.

Решение задачи на ЭВМ естественно начать с повторения выбранного метода и разбора алгоритма. Алгоритм решения рекомендуется представить в виде блок-схемы, на которой можно наглядно проследить ход выполнения решения, затем записать программу на языке программирования. При решении задач вычислительного характера можно сделать оценку затрат времени на выполнение вычислений разными методами и с различной точностью, сравнить с затратами времени на решение вручную. Необходимо обратить внимание, какую роль играет совершенствование численных методов решения задач на рост эффективности использования вычислительной техники, как стремительное развитие вычислительной техники приводит к значительным результатам в разных направлениях вычислительной математики и теории алгоритмов [ 4 ].

Для удобства, простоты и наглядности работы с использованием БЕЙСИКА рекомендуется создать системный диск со следующими необходимыми компонентами операционной системы:

- MXMNSJ.SYS - монитор (управляющая программа операционной системы);
- SWAP.SYS - файл области свопинга (рабочий файл монитора операционной системы);
- TT.SYS - драйвер терминала (программа, управляющая выводом на экран дисплея);
- MX.SYS - драйвер дискового устройства для гибких минидисков;
- LP.SYS - драйвер печатающего устройства типа ТПУ и RОВ;
- BASIC.SAV - интерпретатор языка БЕЙСИК;
- DIR.SAV - программа распечатки каталога дискового устройства;
- PIP.SAV - программа работы с файлами (копирование, переименование, удаление);
- FORMAT.SAV - программа форматирования гибких минидисков;
- DUP.SAV - программа работы с дисками (инициализация, копирование, ожатие);

SCREEN.SAV - экранный редактор (для текстовых файлов и программ).

На диске остается достаточно свободного места для размещения программных файлов. Такой набор системных файлов предоставляет следующие удобства: возможность работы с одним дисковым устройством в случае неисправности второго; простота вызова интерпретатора языка ( BASIC ) и экранного редактора ( SCREEN ); при обращении к программным файлам можно не указывать имя физического устройства, на котором они размещены.

Для ввода программы в память ЭВМ и записи ее на диск можно использовать экранный редактор SCREEN, который дает некоторые преимущества по сравнению с вводом программы в режиме БЕЙСИКА. Экранный редактор позволяет свободно перемещать по экрану с помощью клавиш дополнительной клавиатуры курсор, который является указателем редактируемого символа. Для набора текста программы необходимо установить курсор в соответствующее место и вводить текст. Для ввода исправлений имеется возможность раздвигать текст строки и вставлять необходимые символы, вставлять пустые или удалять ненужные строки. Замена, перемещение, дублирование или удаление части текста выполняется с помощью задания контекста, т.е. уникального набора символов, который однозначно определяет место в тексте. После набора программы или ввода исправлений для проверки правильности набранной программы необходимо закончить работу экранного редактора и вызвать интерпретатор языка программирования БЕЙСИК. Краткое описание языка БЕЙСИК, возможностей операционной системы ОС ДВК и работы с экранным редактором SCREEN можно найти в [ 3 ].

Язык программирования БЕЙСИК дает возможность начинающему пользователю микроЭВМ достаточно легко и быстро овладеть основами программирования. Но нельзя ограничиваться только изучением языка программирования. Ученики должны понять, что неотъемлемую часть математического обеспечения современной вычислительной машины представляет операционная система. На примере решения конкретной задачи на ЭВМ необходимо проиллюстрировать состав операционной системы, назначение управляющей программы МОНИТОР, продемонстрировать работу некоторых

системных программ, например, форматирование, инициализацию и сжатие диска, копирование файлов и устройств. Практическая работа, сопровождаемая объяснением различных возможностей использования системных программ, способствует лучшему пониманию учащимися назначения операционной системы, овладению навыками программирования, укреплению математических знаний, расширяет представление о масштабах всестороннего применения вычислительной техники и необходимости ее использования.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Программа курса "Основы информатики и вычислительной техники". X-XI классы (102 ч) // Микропроцессорные средства и системы. 1986. # 2. С. 86-89.
2. Программное обеспечение ДВК. Операционная система с разделением функций ОС ДВК. Справочник по системе. Справочник оператора. У1.20001-01 90 01-1. 58 с.
3. Озиш И. Я. Работа с КУВТ-86. Операционная система ОС ДВК. Локальная сеть КУВТ-86. Рига: ЛГУ им. П. Стучки. 1987. 48 с. Информ. фонд РФАП Латвии: Инв. # Ю-0020.
4. Самарский А. А., Михайлов А. П. Компьютеры и жизнь (Математическое моделирование). М.: Педагогика. 1987. 128 с.

Межвузовский сборник научных трудов  
ЭВМ В ОБРАЗОВАНИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
1988. Рига: ЛГУ им. П. Стучки. С. 56-67

УДК 681.322.1

С. И. Павлов, А. А. Прикулис

Вычислительный центр при ЛГУ им. П. Стучки

### О ПЕРСПЕКТИВНОМ ШКОЛЬНОМ КОМПЬЮТЕРЕ

В школах страны в соответствии с программой компьютеризации образования уже накоплен некоторый опыт применения как отдельных персональных ЭВМ (ПЭВМ) - "Электроника БК-0010", "ДВК-2М", "Агат", - так и комплектов учебной вычислительной техники (КУВТ) - КУВТ-86, КУВТ "Yamaha". В школы начинают поступать разработанные и выпускаемые по заказу Государственного комитета СССР по народному образованию КУВТ "Корвет" и "Электроника УК-НЦ", а также закупленные за рубежом КУВТ на базе микроЭВМ "MSX-2".

Удовлетворяют ли имеющиеся средства вычислительной техники (СВТ) и их оснащенность системным и прикладным программным обеспечением (ПО) пользователей: учителей и учеников, разработчиков педагогических программных средств ?

Хорошо известны пользователям многочисленные недостатки КУВТ-86, разработанного на базе ПЭВМ "ДВК-2М" [ 1 ] и "Электроника БК-0010" [ 2 ] - неудобная клавиатура, бедные возможности интерпретаторов языков ФОКАЛ и БЕЙСИК, отсутствие устройств ввода-вывода на периферийных ЭВМ, несовместимость на программном уровне центральной и периферийных ЭВМ и др.

Характеристики ПЭВМ "ПК 8020" [ 3 ], "Электроника УК-НЦ" [ 4 ], "MSX-1" и "MSX-2" [ 5 ], входящих в состав производимых в СССР и закупленных за рубежом КУВТ, гораздо лучше, однако концепция организации класса ПЭВМ и его комплектация практически не изменились.

В настоящее время основные моменты концепции КУВТ, представляемых в школы, следующие:

1. Наличие рабочего места преподавателя (РМ1), оснащенного одной ПЭВМ (как правило, более мощной по своим характеристикам) с набором периферийных устройств (накопителем на гибких магнитных дисках (НГМД), печатающим устройством, монитором цветного изображения ("MSX-2"), манипулятором "мышь" ("MSX-2")).

2. Наличие нескольких рабочих мест учащихся (РМУ), оснащенных одной ПЭВМ (как правило, более слабой по своим характеристикам) с ограниченным набором периферийных устройств (монитором черно-белого изображения и кассетным магнитофоном ("Корвет")).

3. Объединение РМ1 и РМУ в локальную сеть, причем сеть не предусмотрена для работы в автоматическом режиме.

По-видимому, необходимо обсудить некоторые требования к ПЭВМ и КУВТ, которые будут разрабатываться для оснащения школ в перспективе. Некоторые требования к КУВТ изложены в [ 6 ]. Сопоставления, приведенные в настоящей работе, основаны на опыте работы с локальными сетями "Econet" для микроЭВМ "Acorn"; КУВТ "Yamaha"; КУВТ "MSX-2"; КУВТ "Корвет"; КУВТ-86; а также с разработанной системой локальной сети для микроЭВМ "Электроника БК-0010" [ 7 ] и по существу отражают интегральную точку зрения пользователя-преподавателя-разработчика.

#### 1. Комплектность классов школьных компьютеров

Вызывает сомнение количество РМУ в КУВТ, составляющее для КУВТ-86 - 12; для КУВТ "Корвет" и "Электроника УК-НЦ" - 12 или 15; для КУВТ "Yamaha" и "MSX-2" - 9, 12 или 15; технически возможно для пяти перечисленных КУВТ подключение соответственно 12, 15, 62, 15, 15 РМУ. С учетом реальной наполненности классов число РМУ в КУВТ при организации практических занятия в двух группах для одного класса с индивидуальной работой школьников с компьютером должно составлять не менее 15-20, а с учетом низкой надежности ПЭВМ - не менее 25-30.

Для эффективной организации работы в комплектацию КУВТ следует ввести следующие периферийные устройства (см. также рис. 1):

1. Внешнее запоминающее устройство (ВЗУ) на базе накопителя на жестком магнитном диске (НЖМД) типа "Бинчестер" емкостью не менее 20-30 Мбайт. НЖМД подключается к управляющей ПЭВМ (а не к ПЭВМ преподавателя!), позволяет создать эффективное обслуживание сети, поддерживать комплект сервисных программ, редакторов, баз данных, демонстрационных пакетов, которые в принципе (при наличии санкции учителя, реализуемой с помощью системы паролей) могут быть доступны любому пользователю в сети. Если локальная сеть будет включать автоматизированные рабочие места (АРМ), расположенные вне кабинета вычислительной техники и используемые для организации управления и делопроизводства в школе, емкость НЖМД следует увеличить на порядок.

2. Одно печатающее устройство (ПУ) на каждые 10 РМУ (не считая РМП). Одно ПУ, расположенного на РМП явно недостаточно для подготовки методических и раздаточных материалов, разнообразной (в т.ч. программной) документации, распечатки промежуточных версий при разработке ППС и т.п. (к перечисленным работам, естественно, следует привлекать учащихся) - всего, что необходимо в работе или невозможно (нецелесообразно) хранить на магнитных носителях.

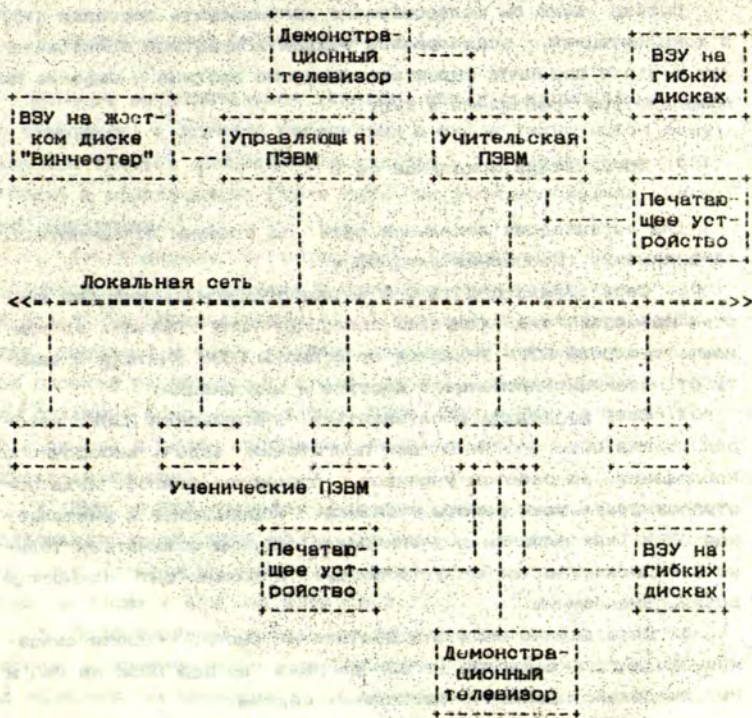
3. Два спаренных НГМД на каждые 5 РМУ (не считая РМП). НГМД необходимы для формирования небольших пакетов тематических прикладных программ, библиотек программ школьников, осуществления обмена информацией с другими учреждениями. При необходимости можно запускать сервисные или управляющие программы, охватывающие группу учащихся, работающих на части РМУ.

В пределах класса необходимо иметь по крайней мере одну ПЭВМ, к которой подключены НГМД с разными размерами диска: 3,5" и 5", объем каждого из которых не менее 512 Кбайт (достаточный для размещения пакета для одного урока).

4. Один кассетный магнитофон (КМГ) на каждые 2-3 РМУ. КМГ будут использоваться для создания архивов программ, личных библиотек школьников и других пользователей, заготовок программ и т.п.

5. Не менее двух на класс демонстрационных телевизоров с большим экраном. Подключение телевизоров производится к РМП

Рис. 1. Конфигурация ПЭВМ и локальной сети.



и одному ("демонстрационному") РМУ и обеспечивает коллективные занятия, конференции, собрания.

Вобщем было бы целесообразно организовать поставки КУВТ с комплектацией, заказываемой каждым конкретным пользователем, что в принципе выдвигает довольно жесткие и широкие по номенклатуре требования к КУВТ.

## 2. Организация локальной сети

Для организации локальной сети со стороны пользователя, по-видимому, требования следующие:

1. Сеть должна работать в автоматическом режиме без участия преподавателя, допуская приоритет ПЭВМ учителя, возможность контроля ПЭВМ учащихся со стороны ПЭВМ учителя и защиту от несанкционированного доступа к информации.

2. Сеть не должна обслуживаться учительской ПЭВМ, которая должна быть доступна для подготовки задач, выборочного наблюдения за работой учеников, просмотра файлов, куда поступают результаты работы учеников. Управляющая и учительская ПЭВМ (как впрочем и ученическая) должны отличаться только функционально (не по устройству) и должны быть полностью взаимозаменяемыми.

3. Сеть должна работать достаточно быстро. Время ожидания при загрузке файлов из ВЗУ на ЮКИД во все ПЭВМ на РМП и РМУ не должно превышать нескольких секунд.

Необходимо предусмотреть также некоторые дополнительные (нелишние) возможности:

1. Подключение к какой-либо (демонстрационной) ПЭВМ принтера в сетевом режиме, т.е. доступного любой ПЭВМ в классе. Это позволяет выводить на печать результаты работы нескольких пользователей в порядке их поступления (например, "оценка" контрольных работ, сочинений и т.п.).

2. Подключение программным путем демонстрационного телевизора к любой ПЭВМ, либо его подключение к демонстрационной ПЭВМ в сетевом режиме, т.е. с возможностью показа изображений, полученных на экране любой ПЭВМ. Отметим, при индивидуальном подходе к обучению с помощью компьютера не следует забывать также о коллективных формах работы, позволяющих

проводить обсуждение результатов, полученных при индивидуальной работе.

### 3. Оборудование рабочего места

Кабинет вычислительной техники ввиду принципиально иной по сравнению с другими кабинетами в школе технической оснащенности требует специального подхода к проектированию, планировке и оборудованию РМП и РМУ. При этом необходимо учитывать следующее:

1. Для создания рабочего места совершенно не подходит традиционное представление о школьном компьютере как о наборе отдельных блоков ("ящиков") - дисплее, клавиатуре, дисководе, принтере и т.д., которые необходимо разместить на любой плоской поверхности (стола, парты). В отличие от [ 8 ] для создания компоновки блоков ПЭВМ, РМП и РМУ, а также всего кабинета в целом необходима комплексная проработка специалиста-дизайнера.

2. РМП и РМУ являются многофункциональными и объединяют требования к обычному месту для занятий учащегося с необходимостью ведения записей в тетради, чтения книг, другой работы, а также к АРМ; на базе ПЭВМ.

3. Проектируемая мебель должна соответствовать гигиеническим и эргономическим требованиям, которые, по-видимому, не являются механическим объединением требований к АРМУ и обычному рабочему месту без компьютера.

### 4. Монитор школьного компьютера

Пока в качестве устройства отображения информации для школьной ПЭВМ применяется или бытовой телевизор - в комплектах КУВТ-86, "Корвет" и "Электроника УК-НЦ" - или монитор - в КУВТ "Yamaha", - не удовлетворяющие гигиеническим требованиям. Работа с ПЭВМ в таких условиях не может быть эффективной - менее получаса в неделю (по временным нормативам Министерства здравоохранения СССР). Только КУВТ "MSX-2" оснащен качественными мониторами, удовлетворяющими жестким гигиеническим требованиям. В дальнейшем должно планироваться

оснащение отечественных КУЗТ только качественными мониторами.

Требования к монитору (гигиенические подробно изложены в [ 9.10 ]), судя по ряду внешних проявлений, следующие:

1. Изображение не должно мерцать.

2. Изображение должно быть устойчивым - недопустима рябь на экране или волны, пересекающие экран.

3. Разрешающая способность монитора должна позволять без напряжения читать текст с 80 символами в строке (не менее 640 точек по горизонтали; по вертикали соответственно).

4. Монитор должен отображать не менее 16 цветов (оптимально не менее 64 цветов) с возможностью их одновременного использования. Цвет в естественных науках несет значительную часть информации. При этом выбор количества цветов в пределах одного кадра должен осуществляться по методическим и психологическим соображениям.

5. Значительную информацию несет также звук. Для монитора необходимо предусмотреть (с соответствующей поддержкой процессором) те же возможности, которыми обладают бытовые телевизоры. Для предотвращения помех другим пользователям необходимо снабдить монитор головными телефонами.

6. В мониторе (как и в ПЭВМ) недопустимо присутствие вентиляторов для охлаждения, создающих шум.

## 5. Клавиатура школьного компьютера

Функционально клавиатура школьного компьютера должна существенно отличаться от клавиатуры обычного компьютера, предназначенного в основном для вычислений. Фактически перечисленные ниже требования относятся к возможностям размещаемого в ПЗУ системного ПО, ответственного за обработку информации, вводимой с клавиатуры (отметим, что ПЗУ должны быть сменными, например, для обеспечения возможности оперативной замены специализированных шрифтов).

Для эффективной работы школьный компьютер должен поддерживать, кроме уже традиционных латинского и русского алфавитов, знаков препинания и специальных знаков, еще дополнительные алфавиты и знаки, используемые в преподавании тех или иных предметов. Причем нередко все знаки нужны одновременно.

в т.ч. (в дополнение к уже перечисленному) шрифты для национального и/или иностранного языка, специальные знаки, применяемые в предметах естественно-научного цикла (математике, физике, химии, астрономии и др.), символы табличной графики, греческий алфавит и т.п. Легко подсчитать, что количество необходимых одновременно знаков (учитывая, естественно, управляющие коды ПЭВМ) значительно превосходит 256, отведенных в стандартной кодовой таблице.

Клавиатура должна удовлетворять следующим требованиям:

1. Необходимо обеспечить возможность переопределения начертания символов. Такое решение заложено в ПЭВМ "Acorn" и "Yamaha". В ПЭВМ "Acorn" вновь определенные символы выводятся при нажатии клавиши с соответствующим (сравнен с определенным системным ПО) кодом. В ПЭВМ "Yamaha" принято альтернативное решение: вновь определенные символы выводятся при одновременном нажатии соответствующей клавиши и служебной клавиши (всего 31 вариант). Пользуясь последним подходом можно создать любое число вновь определяемых символов вплоть до оперативной замены переменной части кодовой таблицы.

2. Кроме возможности переопределения начертания символов следует обеспечить возможность назначать любой клавише (комбинации одновременно нажатых клавиш) произвольный код. При этом необходимо также решить относительно сложную техническую проблему "автоматической" маркировки клавиш. Тогда отпадает проблема маркировки клавиши (например, для текстов латинского алфавита QWERTY или JCPKEN), а также проблема, по каким принципам спаривать буквы различных алфавитов - по звучанию или по графическому подобию. Таким образом, переопределение кодов - возможность вложить в ПЭВМ любой стандарт или создавать свой "собственный стандарт".

3. Необходимо предусмотреть необходимое число переопределяемых функциональных клавиш (не менее десяти), а также функциональный регистр для всех клавиш с заданием ключевых слов основного языка программирования и команд операционной системы. Следует предусмотреть также возможность переопределения ключевых слов.

4. Клавиши переключения регистров (заглавных и строчных букв) должны свободно комбинироваться со всеми другими кла-

видами регистров (например, для ПЭВМ "Yamaha" с <ГРАФ>, <РУС> или <СЛУЖ>) для генерации графических знаков, русских букв и служебных кодов.

5. Необходимо, чтобы любая клавиша переключения регистров, по желанию, могла быть либо зафиксирована, либо сбрасывала только при нажатии.

#### 6. Печатающее устройство

Относительно принтера можно сказать то же, что и о наборе символов для клавиатуры: любой символ, который определяется для экрана, должен выдаваться и на печать.

Печать должна обеспечивать возможности задавать режим подчеркивания, курсив, NLQ-шрифт и т.п. Отметим, что принтер "Gemini", поставляемый с КУВТ "Yamaha", позволяет переопределять только часть кодов (от 32 до 127).

Необходима возможность переопределения всей кодовой таблицы, а также возможность одновременного вывода символов нескольких вариантов переменной части кодовой таблицы.

#### 7. Требования к содержимому ПЗУ школьного компьютера

ПЗУ школьного компьютера должны содержать следующее:

1. Набор программ операционной системы и системы локальной сети;

2. Интерпретатор и транслятор основного языка программирования (например, BASICa);

3. Дополнительно 2-3 языка программирования (в т.ч. PASCAL, C, а также ассемблер-дисассемблер).

4. Стандартный набор символов и набор символов для национального алфавита;

5. Наборы символов для школьных предметов.

6. Стандартные интегрированные пакеты обработки информации - текстовый редактор, электронная таблица, база данных, графический редактор и т.п.

7. Программная поддержка манипулятора "мышь" для ввода графической информации.

## 8. Основной язык программирования

Требования к основному языку программирования следующие:

1. Возможность использования широкого набора алгоритмических конструкций (включая, например, "ЕСЛИ-ТО-ИНАЧЕ", цикл, завершающийся по условию и т.п.).

2. Возможность использования функций и процедур, механизма формально-фактических параметров, локальных переменных, рекурсивности.

3. Наличие глобальных переменных, значения которых сохраняются при загрузке последующих программ (при организации оверлейной структуры).

4. Возможность загрузки дополнительной информации при работе программы на любой, в т.ч. учебной ПЭВМ:

5. Хорошую организацию и сочетание на экране графической и алфавитно-цифровой информации с возможностью многократного определения графических и текстовых окон.

6. Простые средства построения цветных изображений высокого качества, в т.ч. с применением спрайтов.

7. Непосредственное использование команд языка АССЕМБЛЕР в программе на языке программирования с возможностью применения всех объектов языка программирования в языке АССЕМБЛЕР.

8. Возможность программного управления звуком, периферийными устройствами.

Некоторые другие требования, присущие интерпретаторам:

1. Возможность выполнения одного или нескольких операторов без их оформления в виде программы с выводом результатов на экран или печать.

2. Проверка значения каких-либо параметров или правильности языковых конструкций.

3. Запуск отдельных фрагментов программы для отладки с минимальной затратой времени.

Ключевые слова языка программирования вполне могут быть построены на англоязычной основе (это подтверждается опытом обучения другим предметам). В качестве дополнительного может использоваться также язык программирования с ключевыми словами на русском языке, однако при этом возникает существен-

ные проблемы при обучении программированию в национальных школах.

9. Совместимость школьного компьютера с другими средствами вычислительной техникой

Одним из важнейших свойств школьного компьютера должна быть совместимость с другими СЭВ (по-видимому, требования по всему комплексу технических характеристик и свойств ПО более жесткие, чем для профессиональных ПЭВМ общего назначения), как техническая, так и программная, т.е. при разработке компьютера нового поколения необходимо:

1. Обеспечить преемственность технических решений и возможность применения ранних моделей периферийных устройств.
2. При разработке новых версий системного ПО и языков программирования не отказываться от опробованных возможностей ранних версий ПО.
3. Обеспечить возможность переноса уже разработанного ПО на компьютер нового поколения с новыми техническими возможностями.

Соблюдение этих принципов обеспечит:

1. Пользователю преемственность приобретенных при обучении навыков работы с компьютером и ПО при переходе к профессиональным компьютерам.
2. Возможность более простой переработки ППС с одного типа компьютера на другой как отечественного, так и зарубежного производства.
3. Подключение через региональные, всесоюзные или международные сети к банкам информации.
4. Использование домашних компьютеров для выполнения домашних заданий.
5. Программное управление другими техническими средствами обучения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Попов А.А., Хохлов М.М., Глукман В.Л. Диалоговые вычислительные комплексы "Электроника НИ-80-20" // Микропро-

цессорные средства и системы. 1984. # 4. С. 61-64.

2. Косенков С.М., Полосин А.Н., Счепицкий Э.А. и др. Бытовая персональная микроЭВМ "Электроника "БК-0010" // Микропроцессорные средства и системы. 1985. # 1. С. 22-25.

3. Велихов Е.П., Персианцев И.Г., Рахимов А.Т. и др. Персональный компьютер в системе автоматизации физического эксперимента // Микропроцессорные средства и системы. 1986. # 1. С. 34-36.

4. Полосин А.Н., Карпинский Н.Г., Лозовой И.О. и др. Учебный компьютер "Электроника УК НЦ" // Микропроцессорные средства и системы. 1986. # 5. С. 14-16.

5. Новый учебный компьютер MSX-2 (YAMAHA YIS 503 IIIR). Рекламная информация. Токио: Синдайдаяся Ко. Лтд. 1987. 2 с.

6. Архангельский А.Г., Рязанский М.В., Смирнов Е.П. Исходные требования к перспективному комплексу учебной вычислительной техники. М.: НИИ ШО ТСО АПН СССР. 1987. 78 с.

7. Булыгин Л.Л., Эглайс М.О. Локальная сеть в школе // Информатика и образование. 1987. # 3. С. 126-127.

8. Кабинет вычислительной техники всех типов средних учебных заведений (на базе персональных микроЭВМ). Методические рекомендации / Под рук. С.Г. Шаповаленко, Е.П. Смирнова. М.: НИИ ШОТСО АПН СССР. 1986. 39 с.

9. Временные санитарно-гигиенические нормы и правила к устройству видеотерминалов школьных ЭВМ. Часть I. Требования к дисплеям. М.: Министерство просвещения СССР. 1986. 7 с.

10. Гигиенические условия организации учебных занятий с применением компьютеров в средней общеобразовательной школе: Временные методические рекомендации. М.: Министерство здравоохранения СССР. 1987. 15 с.

Межвузовский сборник научных трудов  
ЗВМ в ОБРАЗОВАНИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
1988. Рига: ЛГУ им. П. Стучки. С. 68-74

УДК 37.01:007+74.26+681.3.06

Л. А. Андерсоне  
Республиканский институт  
усовершенствования учителей  
Министерства народного  
образования Латвийской ССР

А. Ф. Раудис, Р. В. Фрейвалд  
Вычислительный центр  
при ЛГУ им. П. Стучки

УЧЕБНЫЕ ТЕЛЕВИЗИОННЫЕ ПЕРЕДАЧИ  
ПО КУРСУ "ОСНОВЫ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ"

С введением во всех общеобразовательных школах республики курса "Основы информатики и вычислительной техники" ("ОИВТ") появилась необходимость ознакомить всех учителей с сущностью нового предмета, а также дать дополнительный материал как учителям, так и учащимся.

До введения предмета Республиканским институтом усовершенствования учителей (РИУ) совместно с Латвийским государственным университетом им. П. Стучки были организованы курсы учителей "ОИВТ". В качестве дополнения к этим курсам был организован цикл телевизионных передач: "Персональные компьютеры в школе", "Персональные компьютеры - в жизнь!". В подготовке передач участвовали сотрудники РИУ и преподаватели ЛГУ, Рижского политехнического института и ПТУ.

Передачи ознакомили учителей с преподаванием нового курса. Затем последовал цикл научно-популярных передач для широкого круга зрителей, в т.ч. для учителей и учеников. В передачах содержалась информация о незнакомом тогда понятии "информатика", о возможностях электронно-вычислительной техники, в т.ч. персональных ЭВМ (ПЭВМ).

Впоследствии появилась необходимость пополнить учебный курс дополнительным материалом. Сотрудниками РИУ и преподавателями ЛГУ были созданы учебные телевизионные передачи: "Информатика и ее технические средства", "Алгоритмы и их по-

строение", "Физические основы работы ЭВМ". В подготовке участвовали разные творческие группы, поэтому передачи получились разными по стилю и качеству.

Одновременно с введением курса "ОИИВТ" появились учебные передачи Центрального телевидения под руководством академика А.П.Ершова. Передачи успешно использовались в учебном процессе, и их материал изучался учителями. Однако следует отметить, что для учащихся национальных школ с латышским языком обучения были сложности с усвоением учебного материала.

#### 1. Принципы построения и отбор материала учебных передач

Для подготовки передач единого стиля и качества необходимо создать постоянный коллектив (творческую группу), изменить стиль телевизионных передач. Такая группа была создана в ВЦ при ЛГУ с привлечением специалистов РИУУ. Ведущим передач стал доктор физико-математических наук Р.В.Фрейвелд.

Передачи являются научно-популярными, содержат элементы актерской игры. По содержанию передачи непосредственно связаны с изучаемым материалом курса "ОИИВТ" и соответствуют определенным темам. В передачах использовалось программное обеспечение, разработанное в основном сотрудниками Лаборатории проблем школьной и вузовской информатики ВЦ при ЛГУ.

Для всех учебных телевизионных передач заранее готовились аннотации [ 1 ], в которых указывалась цель передачи, давались краткие сведения о ее материале и предлагаемых темах для обсуждения. Это помогало учителю своевременно определить связь передачи с предыдущим и последующим материалом, подготовить соответствующие задачи и организовать обсуждение изучаемой темы. Поскольку время передачи - 30 минут, то оставшиеся 15 минут урока учитель мог организовать по своему усмотрению. Указания по использованию передач на уроках рассматривались также на занятиях семинаров и курсов по повышению квалификации учителей.

После выхода передач необходимо было узнать мнения учителей и учащихся, поэтому РИУУ организовал письменный опрос и посещение школ. Были проведены беседы с учителями на республиканских семинарах.

Новый цикл открыла передача "Составление алгоритмов решения задач". Главной целью этой передачи был показ отличий этапов составления алгоритмов. В передаче участвовало несколько действующих лиц: Повар и Вухгалтер (специалисты организации, для которой проводятся вычислительные работы), Математик, Составитель алгоритма и Программист. Переход от одного этапа составления алгоритма к другому осуществлялся путем диалога участников. Далее в заранее отснятом сюжете было показано решение задачи в реальной обстановке. В студии Программист продемонстрировал решение задачи на ПЭЕМ.

Следующая передача "Исполнитель алгоритма - человек и ЭВМ" содержала информацию о конкретном применении компьютеров, в основном персональных. Было сопоставлено выполнение с их и тех же задач человеком и ЭВМ. Первым был рассмотрен текстовый редактор, поскольку на его примере можно просто объяснить, какую часть работы может выполнить компьютер, какую выполняет человек. Компьютер может выровнять правую сторону текста (что для человека крайне трудно), переставить фрагменты текста, подсчитать слова, произвести замену группы символов. В то же самое время проверка правильности текста для компьютера является крайне трудной задачей. Было отмечено практическое значение текстового редактора и указано, что он использовался для подготовки всех сценариев. На экране демонстрировался текст сценария передачи, определялось количество слов, которое должно было быть произнесено за время передачи. Кроме того, в передаче демонстрировался графический редактор, синтезатор звука, тренажер и обучающая система. Главной целью передачи было показать, что компьютеры не заменяют человека, а только выполняют трудоемкие и рутинные операции, расширяют возможности человека, оставляя за ним принятие решения и управление.

В передаче "Пакеты прикладных программ и их применение" было рассказано о пакетах, разработанных в ВЦ при ЛГУ. Была дана классификация пакетов прикладных программ по характеру решаемых задач или по сфере их применения. Разработка таких пакетов стала возможной и целесообразной только после создания универсальных вычислительных машин. В качестве примера были продемонстрированы "Редактор блоксхем" [ 2 ] и пакет

обучающих программ для микроЭВМ "Электроника БК-0010" [ 3 ].

Завершающая передача цикла - "Краткая история развития вычислительной техники". При рассмотрении темы важно было показать развитие не только техники, но и программирования, расширение круга решаемых задач, обратить внимание на то, что при переходе от поколения к поколению существенно менялась как элементная база вычислительной техники, так и технология программирования. Возникли трудности с подготовкой иллюстративного материала, так не удалось найти качественные фотоснимки отечественных ЭВМ. В передаче использовались киноматериалы, снятые в музее ВЦ при ЛГУ.

## 2. Подготовка визуального материала

В каждой передаче рассказ о конкретных проблемах иллюстрировался киноматериалом, снятым в рабочих помещениях ВЦ при ЛГУ. Отдельные сюжеты в разных передачах были представлены одними и теми же персонажами. Иногда для визуализации выполнения того или иного этапа решения задачи в сюжете снималась другая ЭВМ.

В передаче "Краткая история развития вычислительной техники" большая часть материала представлена в виде диапозитивов. Диапозитивы помогли выявить основные тенденции и поворотные точки в развитии вычислительной техники. Важно было также определить время, необходимое для восприятия различных по содержанию диапозитивов, учитывая при этом утомляемость учеников.

## 3. Программное обеспечение передачи

Во всех передачах использовались персональные компьютеры "Acorn", "Yamaha" и "Электроника БК-0010". С помощью этих компьютеров в каждой передаче выводились на экран специально созданные титры, иллюстрации и схемы, а также демонстрационные алгоритмы. Часть пакетов прикладных и обучающих программ была переработана специально для конкретной передачи, часть разработана заново. Ввиду того, что изображение снималось камерой, необходимо было переработать или разработать содер-

жение экрана: специальные увеличенные наборы символов, контрастные, крупные изображения. Кроме того, необходимо было учесть, что в большинстве школ имеются телевизоры с черно-белым изображением.

Необходимо было также сократить возможность ошибок при вводе данных. Для ввода символьных величин уже заранее в программе был заложен правильный набор. При нажатии любой клавиши на экране появлялись необходимые слова. Такие упрощения допустимы.

#### 4. Тиражирование программ во время передач

В передаче "Пакеты прикладных программ и их применение" представлена информация о пакете обучающих программ для ПЭВМ "Электроника БК-0010", разработанном в Лаборатории проблем школьной и вузовской информатики ВЦ при ЛГУ и переданном на тиражирование на фирму "Мелодия".

Поскольку промышленное тиражирование - трудоемкий и продолжительный процесс, была спробована идея передачи программ по звуковому каналу во время телевизионной передачи. Программа во время передачи может быть записана на магнитофон и впоследствии введена в память ПЭВМ.

Таким образом, телевизионная передача послужила качественно новым материалом. Выбор ПЭВМ "Электроника БК-0010" для передачи программ обоснован тем, что ими оснащено большинство кабинетов информатики и вычислительной техники в школах Латвийской ССР. О проводимом эксперименте заблаговременно оповещались учителя, в распоряжении которых находятся классы этих ПЭВМ. Предпочтение было отдано программе-исполнителю "Погрузка корабля" [ 4 ], позволяющей демонстрировать алгоритмизацию процессов, составлять последовательность команд.

Результаты эксперимента были положительны - запись сигнала удалась в большей части школ, которые находятся в зоне надежного приема Латвийского телевидения. Многие учителя высказали свои предложения по выбору программ для передач и форме воспроизведения сигнала.

Кодированный сигнал программы был передан и в следующей

передаче "Краткая история развития вычислительной техники". На этот раз была выбрана программа "Музыкальный синтезатор" [ 5 ]. Сигнал программы для большей надежности записи был передан 5 раз (в первой передаче сигнал программы передавался два раза). Во время передачи кодированного звукового сигнала демонстрировался фрагмент хроники 1964 года: установка в ВЦ ЛГУ ВЭСМ-2.

### 5. Перспективные учебные передачи

Пакет учебных передач переписан на видеокассету, а также дублирован на русский язык. Таким образом, значительно расширен круг потенциальных зрителей передач. Пакет учебных передач будет использован во время занятий с учителями на курсах и семинарах для подготовки к проведению уроков с использованием телевизионных передач.

Предусмотрено подготовить следующие передачи:

- "Понятие об алгоритме" (3 передачи)
- "Вычислительная техника в Латвийской ССР"
- "Предыстория развития вычислительной техники"
- "Алгоритмы для работы с графической информацией"
- "Возможности использования ЭВМ" (3 передачи)
- "Языки программирования" (3 передачи)

### В ы в о д ы

1. Целесообразно создавать циклы учебных телевизионных передач, опираясь на демонстрацию результатов работы конкретного научного коллектива.

2. Для поддержки школьного курса "ОИИВТ" необходимо издать красочные альбомы или серии диапозитивов, иллюстрирующие разные темы курса, в особенности историю развития вычислительной техники в нашей стране.

3. Используя в обучении видеомagnитофон, можно увеличить эффективность учебных передач, исключив пропуск просмотра передач из-за организационных трудностей.

4. Целесообразно средствами теле- и радиовещания передавать обучающие программы, решая таким образом проблему тира-

жирования программ.

5. Эффективность учебной передачи можно увеличить, заблаговременно обеспечив их аннотацией и методическими материалами.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Программа учебных телевизионных передач для 9 - 11 классов общеобразовательных школ и 1-3 курсов профтехнических школ на 1987/88 учебный год. Рига: Министерство просвещения Латвийской ССР, 1987. 86 с.

2. Ионин Г. Л., Супе В. В., Янкевица М. Я. Разработка и применение обучающих программ для персональных ЭВМ // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П. Стучки, 1988. С. 93-102.

3. Булыгин Л. Л., Витиньш М. В., Климанс Л. П., Краст К. А., Павлов С. И., Перфильева Е. В., Прикулис А. А., Рудис А. Ф., Такерис С. Я., Шкерстена И. А. Пакет обучающих программ для микроЭВМ "Электроника БК-0010". Рига: ЛГУ им. П. Стучки, 1988. 34 с.

4. Шкерстена И. А. Программа "Погрузка корабля" для микроЭВМ "Электроника БК-0010" (язык программирования ФОКАЛ). Рига: РФАП ЛатвССР, 1987. Инв. # ПП1035. 35 с.

5. Булыгин Л. Л. Демонстрационная программа для синтеза звука на ЭВМ // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П. Стучки, 1988. С. 174-179.

Межвузовский сборник научных трудов  
ЗВМ в ОБРАЗОВАНИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
1988. Рига: ЛГУ им. П. Стучки. С. 75-84

УДК 37.01:007+371.3+378.147

В.А. Каймин, А.Г. Шеголев

Московский институт электронного машиностроения

ОПЫТ ОБУЧЕНИЯ ИНФОРМАТИКЕ И ПРОГРАММИРОВАНИЮ  
ШКОЛЬНИКОВ И СТУДЕНТОВ

Перспективы развития научно-технического прогресса, интенсификация производства, совершенствование управления и образования на основе применения вычислительной техники и средств автоматизации вызвали значительный интерес к проблеме подготовки кадров в области информатики, вычислительной техники и программирования. Необходимо вооружить учащихся знаниями, навыками и умениями использования современной вычислительной техники и обеспечить широкое применение компьютеров в учебном процессе.

Одной из основных задач в обучении школьников информатике является приобретение опыта работы с ЭВМ, с деловыми и учебными программами и умения составлять алгоритмы и программы для решения простейших задач. В связи с этим возникает необходимость в постановке лабораторного практикума по информатике на ЭВМ и в выработке методики обучения алгоритмизации.

Для студентов, специализирующихся в области информатики и применения вычислительной техники, процесс обучения должен предполагать изучение научных основ информатики и овладение технологией разработки программных средств. Для этого необходима разработка научно-методических основ информатики и программирования и постановка учебно-производственных работ по созданию программных средств определенного (в частности, учебного) назначения.

В Московском институте электронного машиностроения

(МИЭМ) с 1980 года студенты осваивают программирование на основе современных принципов и методов теоретического, прикладного и системного программирования, включающих идеи систематического построения алгоритмов и программ с доказательством их правильности, а также элементы технологии программирования. С 1982 года информатика и начала программирования изучаются школьниками 8-10 классов, поступающими в физико-математическую школу МИЭМ. В 1985/86 учебном году школьники и студенты прошли лабораторный практикум в кабинетах учебной вычислительной техники и лаборатории информатики МИЭМ.

В настоящей статье излагается опыт постановки лабораторных практикумов с использованием учебных программ, идеи систематического построения алгоритмов и опыт использования их в обучении школьников и студентов.

Обучение студентов и школьников информатике базируется на системе понятий, которую можно проиллюстрировать следующей схемой:



Содержанием информатики как научной дисциплины является изучение информационных процессов:

- в общении людей;
- в общении человека и ЭВМ;
- в жизни человеческого общества в целом.

Деятельность людей (и общества в целом) происходит в предметном мире и состоит в решении определенных задач.

Всякая задача - это требование получения определенных

результатов в определенных условиях.

Решение задач требует определенных средств и определенных способов получения требуемого. Применение ЭВМ для решения задач требует формализации постановок задач, опирающихся на определенные модельные представления, а также обобщения и строгого обоснования методов (способов) их решения.

Процесс решения человеком задач с помощью диалоговых ЭВМ можно уподобить постановке некоторого спектакля, требующего сценария.

Функционирование ЭВМ осуществляется под управлением программ, реализующих определенные алгоритмы, которые должны строго соответствовать выбранным сценариям и методам решения задач.

В информатике как учебной дисциплине можно выделить несколько уровней изучения.

1-й уровень - уровень компьютерной грамотности - это приобретение навыков общения с ЭВМ и умения составления простейших программ.

2-й уровень - алгоритмическая культура - знание обобщенного алгоритмического языка для описания алгоритмов и умение составлять алгоритмы для решения задач.

3-й уровень - информационная (математическая) культура - умение ставить задачи, подбирать и обосновывать методы решения, а также строить модели, необходимые для математических постановок задач.

Педагогический эксперимент по постановке обучения основам информатики и программирования в МИЭМ был проведен в вечерней физико-математической школе, в которой учащиеся средних школ более глубоко изучают информатику и вопросы применения ЭВМ. Учебно-методическими целями этого обучения являются формирование математической культуры и расширенная компьютерная подготовка, соответствующая конкурсной программе курса "Основы информатики и вычислительной техники". Физико-математическая школа стремится сформировать у учащихся:

- умение ставить задачи и подбирать математические модели;
- умение тщательно планировать свою деятельность;

- умение логически обосновывать свои решения;
- знание методов составления алгоритмов и проверки их правильности;
- знание принципов функционирования ЭВМ.

Учебный план объединяет три взаимосвязанных курса: математика, физика и информатика.

Компьютерный практикум по этим учебным дисциплинам включает в себя:

- уроки информатики с использованием учебных, игровых и моделирующих программ;
- освоение возможностей персональных ЭВМ;
- лабораторные работы по физике;
- лабораторные работы по математике;
- практикум по составлению алгоритмов и программ.

Поставовка лабораторного практикума требует соответствующего программного обеспечения. В МИЗМ для нужд школьной информатики созданы и апробированы в физико-математической школе МИЗМ следующие учебные программы:

- клавиатурный тренажер ("Азбука");
- учебные графические редакторы ("Чертилка", "Паучок", "Дизайнер", "Зонтик");
- учебные базы данных ("Сводка");
- моделирующие программы по отдельным разделам физики и производственному обучению ("Станок", "Конвейер", "Трактор", "Полет");
- игровые программы, развивающие логическое мышление и пространственное воображение ("Зврика", "Пирамида", "Сфера", "Пятнашки").

Учащиеся приобретают навыки общения с ЭВМ. Подмечено, что после освоения клавиатуры и работы с графическим редактором школьники быстро осваивают систематические методы построения алгоритмов и программ.

Ключевой в обучении информатике является работа по составлению алгоритмов и программ. С методической точки зрения, важнейшим здесь является проверка правильности решений, представляемых учащимися. Поэтому учитель должен уметь отвечать на следующие вопросы:

- почему данный алгоритм правильный, а в другом - содержатся ошибки?
- почему алгоритм построен так, а не иначе?
- зачем нужен данный алгоритм?

Погробируем ответить на эти вопросы.

Существуют три основных способа составления программ:

- 1) традиционная практика;
- 2) структурный подход к программированию;
- 3) систематическое построение алгоритмов и программ.

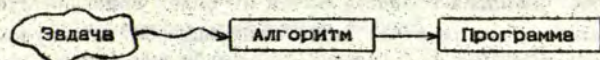
Рассмотрим каждый из этих способов.

Первый способ, который мы называем традиционным, можно изобразить в виде следующей диаграммы:



Написание программы производится, исходя из содержательной постановки задачи непосредственно на языке программирования. Эти языки имеют, как правило, английскую лексику; для людей, не изучавших английский, возникает труднопреодолимый "языковой барьер". Обычно в программах имеется много ошибок, поиск и исправление которых начинаются после ввода программы в ЭВМ. Продолжительность отладки заранее непредсказуема, и даже после ее окончания нет никаких гарантий отсутствия ошибок в программе. Применение такой методики в обучении в большинстве случаев приводит к порождению неуверенности в своих силах и, как правило, влечет за собой мышинобоязнь (компьютерфобию).

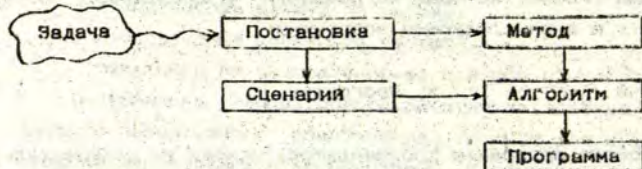
Структурный подход к программированию предполагает составление структурированных алгоритмов на обобщенном алгоритмическом языке:



Алгоритмы записываются с использованием русской лексики в специальной структурированной форме со ступенчатыми отступами. Основные достоинства подхода: возможность чтения алгоритма вслух и, как правило, возможность понимания результата

тов выполнения алгоритма за одно прочтение этого алгоритма сверху вниз. В таких алгоритмах обычно легко разбираться и выявлять ошибки. Колирование программ почти всегда представляет собой механическую работу и проводится с помощью справочника по языку программирования [ 1.2 ]. Как правило, при структурном подходе получаются легко понимаемые, правильные (не содержащие ошибок) программы.

Систематическое построение алгоритмов является развитием структурных методов программирования и состоит из следующей последовательности этапов:



Первый этап - строгая постановка задачи - заключается в определении набора исходных данных, набора требуемых данных и в определении зависимости требуемых результатов от исходных условий. Строгая постановка задачи позволит в дальнейшем однозначно определить, какие результаты будут считаться правильными, а какие нет в любых конкретных условиях.

Второй этап - определение метода решения - состоит в определении набора промежуточных результатов, позволяющих получить то, что требуется постановкой задачи. Метод может состоять в выделении более простых подзадач, результаты решения которых позволяют получить решение задачи в целом.

Третий этап - составление сценария - заключается в определении правил общения с ЭВМ для организации решения поставленной задачи. Сценарий представляется в виде совокупности картинок, фрагментов текстов, выдаваемых машиной, и системы возможных команд и запросов человека, а также реакции машины на любые его действия.

Четвертый этап - построение структурированных алгоритмов в соответствии со сценариями и методами решения задач. На этом этапе необходимы тщательное продумывание логики работы

компьютера и подбор вспомогательных алгоритмов, описывающих логику работы "исполнителей".

Последний этап - кодирование программы - проводится в строгом соответствии с предварительно составленными структурированными алгоритмами. Применение обобщенного алгоритмического языка позволяет достаточно легко "перевести" алгоритмы на практически любой из используемых языков программирования таких, как PASCAL, BASIC, PL/1, FORTRAN и т.п.

Вернемся к трем основным методическим вопросам, поставленным ранее.

Для ответа на вопрос о выявлении ошибок в алгоритмах и программах необходимо определить, что является ошибкой.

Об ошибках в программах мы судим по их проявлениям:

- 1) отказы и сбои в работе машины из-за нарушений логики обработки данных;
- 2) получение неправильных результатов;
- 3) завершение работы программы из-за недопустимых или неправильно записанных данных;
- 4) получение непонятных или оскорбительных сообщений от компьютера или отсутствие всяких сообщений.

Ошибки обычно делят на синтаксические и алгоритмические. Как известно, синтаксические ошибки - это нарушение правил записи выбранного языка программирования. Алгоритмические же ошибки следует подразделить на следующие типы:

- а) неправильная реализация в программе алгоритмических конструкций;
- б) неправильная реализация выбранного метода решения;
- в) дефекты в организации диалога человека с машиной в процессе решения задачи;
- г) принципиальные ошибки в методе (способе) решения;
- д) искажение требований задачи.

Правильным может быть только тот алгоритм, который выдает результат, строго соответствующий требованиям постановки задачи. Это должно выполняться для всех данных, удовлетворяющих исходным условиям. Для того, чтобы однозначно судить о правильности или неправильности алгоритма, необходимо иметь строгие в математическом смысле описания требования и исход-

ных условий задачи - постановку задачи. При отсутствии математических постановок задач оценка правильности алгоритмов становится чисто субъективной.

Строгие однозначные постановки решаемых задач - важнейший этап в построении алгоритмов и в любой деятельности людей и коллективов. Строгая постановка задачи должна давать ответ на следующие четыре вопроса: 1) что дано? 2) какие исходные условия допустимы? 3) что требуется? 4) как связано требуемое с исходным? Ответы на вопросы о данных и результатах состоят в перечислении состава исходных и требуемых данных. Ответы на вопросы о связях и допустимости исходных данных должны быть представлены в виде утверждения, допускающих однозначную проверку: какие данные считаются допустимыми, а какие - нет, какие результаты будут считаться правильными, а какие нет.

Используемая нами форма записи постановок задач:

<b>ЗАДАЧА:</b>	( содержательная формулировка задачи )
<b>ДАНО:</b>	( перечисление исходных данных )
<b>ПРИ:</b>	( условия допустимости исходных данных )
<b>ТРЕБ:</b>	( перечисление требуемых данных )
<b>СВЯЗ:</b>	( зависимость требуемого от исходного )

Для задач, решаемых с помощью ЭВМ (являющихся математическим инструментом обработки данных), постановки должны получать математическую форму. В этих целях могут и должны использоваться различные системы уравнений и неравенств, а также средства математической логики.

Для прикладных задач численного характера (расчетных задач физики, химии, геометрий, технического проектирования и конструирования) зависимости между требуемыми и исходными данными обычно имеют форму систем уравнений, построенных на основе анализа содержания задач и применения соответствующих (физических, химических, геометрических и т.д.) законов. Совокупность этих уравнений образует математическую модель процесса, явления или конструкции.

Последним этапом в составлении алгоритмов и программ

должна быть проверка их правильности. Эталоном для проверки правильности алгоритмов должны быть математические постановки решаемых ими задач, в эталонном "для проверки диалоговых программ - полное детальное описание сценария работы с ЭВМ.

Традиционно анализ правильности программы сводится к их вводу и испытаниям на ЭВМ, в ходе которых автор программы начинает уточнять математическую суть решаемых задач и функции программы. Такой подход в разработке алгоритмов приводит к длительной и мучительной отладке программ, продолжительность которой непредсказуема.

Наличие постановок задач и сценариев превращает процесс испытаний программ на ЭВМ в вычислительный эксперимент, для подготовки к которому нужно подобрать серию изощренных тестов с целью обнаружения тонких ошибок. Обнаружение ошибок должно повлечь небольшое расследование, заключающееся в поиске причин ошибок по их внешним проявлениям и обнаружении места их возникновения (в постановке, методе, алгоритме, программе или сценарии).

Поиск ошибок - традиционно мучительный процесс - при систематическом построении алгоритмов становится прозрачным и легко управляемым. Правильность программ сверяется по отношению к алгоритмам, алгоритмы проверяются на их соответствие сценариям и реализуемым ими методам вычисления. Правильность методов определяется на их соответствие постановкам решаемых задач.

Центральная проблема - анализ правильности алгоритмов - решается с помощью средств элементарной школьной математики выявлением методов вычисления, реализуемых алгоритмами, по этим алгоритмам.

Осознание или доказательство правильности алгоритмов позволяет обсуждать вопросы: нельзя ли улучшить, обобщить, оптимизировать выбранные методы и алгоритмы решения задач? Явное и строгое их описание дает возможность организовывать такого рода обсуждения с учащимися и искать пути, способы и приемы улучшения алгоритмов и обобщения методов решения задач. Последнее тесно связано с постановками задач и моделированием проблемных ситуаций и, как следствие, открывает возмож-

ность исследования вариантов постановок задач и выбора моделей в их взаимосвязи с вычислительным экспериментом на ЭВМ.

Развитие и распространение систематических методов построения алгоритмов и разработки программ для ЭВМ, как показывает наш опыт, позволяет существенно упростить и упорядочить процессы программирования, сделать разработку алгоритмов и программ понятной и контролируемой от начала и до конца, существенно упростить и интенсифицировать процесс обучения информатике и программированию за счет упорядочения интеллектуальной деятельности обучаемых. Самым значительным для нас является появление у наших учащихся активной позиции в желании изучать программирование, работать на ЭВМ, а также проявление творческого отношения к двойной деятельности.

#### Л И Т Е Р А Т У Р А

1. Каймин В.А., Питеркин В.М. Основы информатики и вычислительной техники. Пособие для учителей. М.: МИЭМ, 1985.
2. Каймин В.А. Методы разработки программ на языках высокого уровня. М.: МИЭМ, 1985.
3. Тыгу Э.Х. Концептуальное программирование. М.: Наука, 1984.
4. Фридман Г.Д. Как решать задачу? М.: Просвещение, 1985.

Межвузовский сборник научных трудов  
ЭВМ в ОБРАЗОВАНИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
1988. Рига: ЛГУ им. П.Стучки. С. 85-92

УДК 37.01:007+371.3+519.6+681.3.06

Е.С.Кельман, В.Д.Литвинский, С.И.Мельник  
Вычислительный центр при ЛГУ им.П.Стучки

#### РАЗРАБОТКА ПРОГРАММ НА КРУЖКОВЫХ ЗАНЯТИЯХ ШКОЛЬНИКОВ НА ЭВМ СМ-4

В настоящей работе отражен опыт занятий со школьниками 7-10 классов одного из кружков ВЦ при ЛГУ им. П.Стучки. Эти занятия были начаты в 1982 году. С появлением в 1985 году дисплейного класса стало возможным проводить занятия на 10-14 дисплеях одновременно. Основная цель занятий - пробудить и развить интерес, ознакомить школьников с возможностями и областями применения вычислительной техники, выработать простейшие навыки алгоритмизации, постановки задач и разработки программы. ЭВМ при этом рассматривается не как объект, а как инструмент исследования. Разработка программ школьниками на этих занятиях является закономерным завершением определенных этапов обучения.

#### 1. Организация занятий

Занятия проводятся 1-2 раза в неделю в течение всего учебного года. Каждое занятие продолжается 2-3 часа и состоит из теоретической и практической части. На теоретических занятиях читаются лекции по новому материалу, решаются задачи, даются индивидуальные задания, устраивается коллективный разбор программ, составленных школьниками. Практические занятия проходят в дисплейном классе миниЭВМ "СМ-4" и состоят в отладке и тестировании программ, индивидуальной работе с учащимися. Техническую базу занятий составляют миниЭВМ "СМ-4" (ОЗУ - 256 Кбайт, операционная система RSX11M, версия 4.1); два дисководы "ИЗОТ 5400" для дисков "ЕО 5269-

01" емкостью 2,4 Мбайт, 12-14 алфавитно-цифровых монохроматических дисплеев "СМ-1604".

В соответствии с основной целью разработан 4-летний план занятий, включающий некоторые разделы дискретной и непрерывной математики, основы технологии программирования, вопросы, связанные с применением ЭВМ. Эффективным введением в предмет является редактирование текстов. Это позволяет сразу включить в занятия практикум на ЭВМ, который проводится на протяжении всего 4-летнего цикла, способствует формированию более правильного представления о возможностях ЭВМ. Удобным при работе на ЭВМ "СМ-4" в многотерминальном режиме является текстовый редактор ТЕД. Школьники изучают его по методическому пособию [ 1 ]. Простота, удобство и содержательность команд ТЕДа, наглядность его работы позволяют уже на начальном этапе обучения легко и естественно ввести и глубоко усвоить такие важнейшие понятия, как алгоритм, команда, файл и т.п., показать богатство идей и применений программирования, рассеять распространенные заблуждения.

В качестве основного языка программирования выбран Бейсик. Как первый язык он имеет много недостатков. Однако использование более современных языков (таких как Пискаль, Си, Пролог) в режиме одновременной работы 10-14 дисплеев оказалось невозможным. Устранить некоторые недостатки Бейсика помогает, во-первых, изучение ТЕДа, во-вторых, школьники сразу начинают изучать программирование как структурное [ 2 ], усваивая некоторый набор правил для представления основных структур (следствие, развилка, цикл) с помощью операторов языка в удобном для восприятия виде. Структурное кодирование особенно полезно для начинающих: заставляет их выделять в тексте программ основные структуры, приучает к дисциплине программирования, облегчает разработку и отладку программ, упрощает контроль за обучающимися.

В 4-летний план проведения занятий, помимо изучения ТЕДа и Бейсика, входит изучение систем счисления (арифметика и применение систем счисления, отличных от десятичной), элементов математической логики (запись утверждений с помощью предикатов, таблицы истинности, логические операции), изучение особенностей машинной арифметики, методов решения урав-

нений и систем уравнений (методы Гаусса, Зейделя, простой итерации, секущих, касательных), простейших методов оптимизации (методы половинного деления, "золотого сечения", покоординатного спуска), технологии программирования (проектирование программ, выбор алгоритма, стиль написания программ, построение системы тестов) и др. Основное содержание 4-летнего курса разбито на темы, большинство из которых изучается на новом уровне в следующие годы. Такими темами, например, являются технология программирования, элементы математической логики, простейшие численные методы, устройство ЭВМ, комбинаторика. Для практических занятий на ЭВМ разработан задачник по программированию [ 3,4 ], пригодный в течении всего курса обучения. В нем содержится задачи по материалу всего курса. Эти задачи школьникам предлагаются (или выбираются ими) в качестве индивидуальных заданий.

## 2. Примеры программ школьников

В результате кружковых занятий за последние три года школьниками разработано около 250 программ. Ряд программ используется для проведения занятий и экскурсий. В качестве примеров были отобраны программы [ 5 ], представляющие, во-первых, различные темы, во-вторых, различный возраст школьников: от 7-х (программы PI, TEN) до 9 - 10-х классов. По темам программы разделены на "системные" (STE, DIRECTORY), численные методы (PI), учебно-обучающие-тренировочные (PERIOSSTM, TEN, MEMORY), игровые (QUASIBAS, HANGMAN, FIFTEEN). Следует отметить, что не всегда программу можно твердо отнести к той или иной теме (например, QUASIBAS и игровая, и "системная", а MEMORY - и тренировочная, и игровая). Все программы написаны на языке Бейсик версии RSX для ЭВМ "СМ-4".

Программа STE предназначена для редактирования небольших текстов и их увеличения. При ее вызове на экране появляются описание программы, инструкция и "рамочка", как на рис. 1-а. Перемещая курсор внутри рамочки, можно ввести и отредактировать любой текст из 3 строк по 9 символов в каждой (например, как на рис. 1-б). Нажав на клавишу <служ.код> и, не отпуская ее, на клавишу с буквой <O>, вы увеличите символы так,

Рис. 1. Примеры работы программы STE.

```
*****
*           *
*           *
*           *
*****
а)

*****
* SM-4 - *
* ОЧЕНЬ ХО-*
* РОШАЯ ЗВМ*
*****
б)
```

Рис. 2. Варианты игры на поле 6x6.

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
а)

* * * * *
* * 15 * * *
* * * * 12 *
* * * * *
* * * * *
* * * * *
б)
```

Рис. 3. Игра "15"

```
-----
| 7 | 11 | 8 | 2 |
-----
| 1 | 10 | 15 | 6 |
-----
| 4 | 5 | 13 | 3 |
-----
| 14 | 9 | 12 |  |
-----
```

что текст или рамочки займет весь экран. Затем его можно за-  
помнить или отпечатать. Название программы предложено школь-  
никами и образовано из слов TEP (текстовый редактор) и SUPLR  
(т.к. программа STE "умеет" получать большие буквы, а TEP  
"не умеет").

Программа DIRECTORY предназначена для чтения и удаления  
файлов заданной директории. Она оказывает большую помощь ру-  
ководителю занятий, позволяя читать программы, записанные в  
заданной им директории, и (если необходимо) удалять их. Если  
сразу после ввода программы нажать клавишу <S>, пользователь  
увидит название первого файла директории; нажав клавишу  
<R>, - первые 15 строк этого файла; последующим нажатием кла-  
виши <C> обеспечивается дальнейшее чтение файла; нажатием  
клавиши <D> файл подготавливается к удалению, если пользо-  
ватель подтвердит желание удалить файл (в ответ на запрос  
программы DIRECTORY), то рассматриваемый файл будет удален;  
нажав <G>, пользователь переходит к рассмотрению следующего  
файла; клавиша <F> показывает возможности программы, а кла-  
виша <E> обеспечивает окончание ее работы. Программа защище-  
на от несанкционированного доступа.

Программа PI приблизительно вычисляет число  $\pi = 3.1415926..$   
методом "случайной стрельбы в круг". Метод заключается в  
следующем. Пусть в единичный квадрат вписан сектор (четверть  
круга) единичного радиуса. С помощью датчика равномерно рас-  
пределенных случайных чисел производится "выстрел" по квад-  
рату. Подсчитывается отношение числа выстрелов, попавших в  
сектор, к общему числу выстрелов. Нетрудно видеть, что это  
отношение равно отношению площади сектора и квадрата, т.е.  
 $\pi/4$ . При достаточно большом ( $\approx 10^3$ ) числе выстрелов можно  
с относительно неплохой точностью ( $10^{-1}$  --  $10^{-2}$ ) получить  
число  $\pi$ . Например (число выстрелов задается пользователем):

число выстрелов	приближенное значение
I	0
II	3.27273
III	3.06306
IIII	3.11071

Программа PERIODSTM - справочник по периодической систе-  
ме элементов. Эта программа по одному или нескольким из эле-

данных признаков химического элемента (порядковый номер, символ, название на русском или латинском языках, атомный вес, группа и ряд в периодической системе, количество электронов на внешнем уровне) позволяет узнать один или несколько дополнительных признаков. Если введенным признакам удовлетворяют несколько элементов, печатается информация о каждом из этих элементов. Если элементов, удовлетворяющих введенным признакам, нет, программа сообщает об этом.

Программа TEN переводит целые числа от - 32768 до 32767, заданные пользователем в десятичной системе, в их машинное представление в двоичной и шестнадцатеричной форме записи (под машинным представлением понимается дополнительный код, т.е. двоичное или шестнадцатеричное представление исходного числа +  $2^{16}$ ). На экране печатаются 16 символов в двоичной форме записи и 4 символа в шестнадцатеричной форме. Например:

введено	двоичная форма	шестнадцатеричная форма
0	1000 0000 0000 0000	8000
125	1000 0000 0111 1101	807D
-32768	0000 0000 0000 0000	0000
32767	1111 1111 1111 1111	FFFF

На ввод числа, входящего за указанную границу, программа печатает сообщение, что число слишком мало или велико.

Программа MEMORY предназначена для тренировки памяти. Она моделирует игру PE - XE - SO. Суть этой игры заключается в следующем. Играют от 2 до 15 человек. Имеются 4, 16, 36, 64 (по желанию играющих в зависимости от их числа) карточки, на каждой из которых написано число. Существуют ровно две карточки, на которых написано одно и то же число. Карточки кладутся "рубашкой вверх" (рис. 2-а), образуя квадрат размером 2x2, 4x4, 6x6 или 8x8. В свой ход игрок переворачивает любые две карточки, т.е. указывает координаты строки и столбца в игровом квадрате. Если на карточках одиночные числа, то игрок получает очко и право сделать следующий ход, а карточки остаются "открытыми" до конца игры. Если же числа разные (рис. 2-б), то очередь хода передается следующему игроку, а карточки переворачиваются "вверх рубашкой". Числа на карточках - двузначные (от 10 подряд в порядке возрастания; каждое число повторяется дважды). Карточки восстанавливаются

программой случайным образом. Выигрывает тот, кто наберет больше всех очков, т.е. "поднимет" наибольшее число карточек.

Программа QUASIBAS - программа-шутка, имитирующая команды Бейсика. По словам ее авторов, "она приводит в недоумение, а затем в ярость малоопытного, но что-то знающего программиста на Бейсике". При обращении к этой программе на экране появляется строка READY. Пользователь впадает в заблуждение, думая, что это ответ Бейсика, а программа не подключилась. Если он вызовет теперь список программ (команда CAT), то на экране появится правдоподобный каталог. Попытки запустить какую-либо программу из этого "каталога" наталкиваются на нежелание машины делать это (сообщения NO CHOICE, READ, ?SYNTAX ERROR и т.д.). Попытки просмотреть "текущую программу" (команда LIST) вызовут появление синтаксически правильной, но бессмысленной программы. Ввод команд GUE, RUN, STOP, OLD, NEW, SUB и др. влечет сходные последствия. Выход из программы - "команда" GUE, ПОЖАЛИСТА!

Программа HANGMAN фиксирует позицию в известной игре "виселица". Один из играющих (играют двое) задумывает какое-либо слово и вводит его. Например, задумано слово ПРОГРАММА. На экране появится такая картина:

П - - - - А - - А

Отгадывающий вводит букву, например, "Р". Картинка изменится:

П Р - - Р А - - А

Если отгадывающий введет букву, которой нет в задуманном слове, то на экране появится соответствующее и будет нарисован элемент "виселица". "Виселица" состоит из 10 элементов, т.е. после десятой ошибки отгадывающий будет "повешен". Если отгадывающий введет букву, которую он уже называл, программа сообщит об этом и ошибка не будет засчитана.

Программа FIFTEEN позволяет использовать экран для игры в "15". Правила игры (по желанию пользователя) выводятся на экран. Затем возникает картинка (рис. 3) - разумеется, числа каждый раз расположены по-другому. Перемещения на пустое поле осуществляются три помощи клавиш <2>, <4>, <6>, <8> (вниз, влево, вправо, вверх). Программа FIFTEEN, в отличие от реальной игры, всегда может закончиться успехом. Исполнение

позиции, выбранные с помощью датчика случайных чисел таковы, что всегда можно расставить числа от 1 до 15 по порядку.

#### Заключительные замечания

Опыт показывает, что школьники в состоянии разрабатывать программные модули при условии тщательной организации этой разработки и руководства ими, особенно на этапах постановки задачи и проектирования программы. Как можно раньше следует принимать меры против детского "хакерства" и компьютерного хулиганства: у некоторых учеников возникает гипертрофированные представления и отношение к программированию и ЭВМ, желание работать в запрещенных (системных) директориях "только, чтобы что-нибудь так сделать". Существенно затрудняет работу со школьниками отсутствие ориентированных на работу с детьми: а) программного и методического обеспечения; б) вычислительной техники (в частности отечественные и большинство зарубежных дисплеев не удовлетворяют санитарным нормам [ 5 ]).

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кельман Е.С., Мельник С.И., Семушин В.И. Краткое описание текстового редактора ТЕР. Рига: ЛГУ им. П.Стучки. 1987. 15 с. (Информ. фонд РФАП Латвии: Инв. # Ю-0022)
2. Кельман Е.С., Мельник С.И. Структурное кодирование на Вейксике. Рига: ЛГУ им. П.Стучки. 1986. 17 с. (Информ. фонд РФАП Латвии: Инв. # Ю-0017)
3. Мельник С.И. Задачи по программированию. Рига: ЛГУ им. П.Стучки. 1987. Ч. I. 63 с.
4. Мельник С.И. Задачи по программированию. Рига: ЛГУ им. П.Стучки. 1987. Ч. II. 66 с.
5. Кельман Е.С., Литвинский В.Ю., Мельник С.И. Примеры программ школьников. Рига: ЛГУ им. П.Стучки. 1987. 12 с. (Информ. фонд РФАП Латвии: Инв. # Ю-0031)
6. Политевская А. Компьютер и дисплей // Неделя. 1987. # 12. С. 8.

Межвузовский сборник научных трудов  
ЭВМ в ОБРАЗОВАНИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
1988. Рига: ЛГУ им. П. Стучки. С. 93-102

УДК 37.01:007+371.3+681.3.06

Г. Л. Ионин, В. В. Супе, М. Я. Янкевица  
Вычислительный центр при ЛГУ им. П. Стучки

#### РАЗРАБОТКА И ПРИМЕНЕНИЕ ОБУЧАЮЩИХ ПРОГРАММ ДЛЯ ПЕРСОНАЛЬНЫХ ЭВМ

В жизни средней школы произошло важное событие - началось преподавание нового предмета "Основы информатики и вычислительной техники". Началось внедрение ЭВМ и их использование в учебном процессе.

Общение с ЭВМ вызывает у учащихся большой интерес. Сотрудники отдела исследования операций ВЦ при ЛГУ им. П. Стучки несколько лет вели кружковые занятия по информатике на мини-ЭВМ "СМ-4" для учащихся школы # 73 г. Риги. Ученики занимались составлением и отладкой программ на языке BASIC. Обучающие программы и пакеты прикладных программ в работе с учениками не применялись. Опыт показывает, что наибольший интерес к работе на ЭВМ ученики проявили в 1985/86 учебном году. В последующие годы интерес постепенно снижался: по-видимому, ученики не осознавали, где и как реально использовать ЭВМ в учебном процессе и в своей дальнейшей трудовой деятельности.

Эффективное применение ПЭВМ в учебном процессе в значительной мере зависит от наличия программного обеспечения, учитывающего имеющиеся технические возможности (графическое представление, цвет, звуковое сопровождение, возможность хранения больших массивов данных и т. д.). Использование ПЭВМ позволяет индивидуализировать обучение учащихся и является дополнительным мощным средством активизации творчества и повышения производительности труда учащихся при освоении нового материала и решении задач практически по всем предметам.

Одновременно со школьной началось развитие и вузовской информатики. Для студентов важно обеспечить доступ к полно-

ценной вычислительной технике и ознакомить их с проблемно-ориентированными пакетами прикладных программ.

В настоящее время в стране интенсивно ведется разработка обучающих программ и проблемно-ориентированных пакетов прикладных программ, охватывающих новые области применения и методы. В дальнейшем следует ожидать качественного скачка в создании пакетов программ, развивающих творческие способности учащихся.

Обучающие программы призваны автоматизировать умственную деятельность человека и являются частью реальной ситуации, которую они моделируют. Такая ситуация связана с контуром обратной связи, и при разработке программных средств необходимо соблюдать ряд принципов, которые будут рассмотрены на примере программ, разработанных сотрудниками нашего отдела для учебных целей.

#### 1. Характеристика разработанных программных средств

Для ЭВМ "Yamaha" сотрудниками отдела разработан ряд программных средств, предусмотренных для учебного процесса и нашедших практическое применение: из них рассмотрим редактор блоксхем RED (авторы - А. Гауенс, А. Дзерве, В. Супе, Л. Ковалевский) и систему имитационного моделирования SIM/BASIC (авторы - Я. Канепс, М. Янкевица).

Педагогическая практика показывает, что эффективными средством для изучения понятия "алгоритм" являются блоксхемы. Первые поколения программистов их успешно использовали в своей трудовой деятельности. Сейчас этот метод описания алгоритмов необоснованно забыт, но с появлением ЭВМ с графическими дисплеями может быть снова введен в практику обучения и описания алгоритмов. "Блоксхемы обладают значительно большей наглядностью, чем обычные языки программирования (как высокого, так и низкого уровня). Следовательно, они более эффективны (по крайней мере как средство обучения), а один из принципов современной дидактики требует представления учебного материала по возможности в наиболее наглядной форме. Поэтому вызывает удивление то, что в существующем учебном пособии по информатике блоксхемы почти не используются" [ 1 ].

Редактор блоксхем RED (рис. 1) не только помогает сформулировать алгоритм при помощи графических символов, но так же отрабатывает навыки обучаемого по взаимодействию с ПЭВМ, освоению клавиатуры, приобщает ученика к ПЭВМ еще до освоения программирования.

В природе, технике, различных отраслях народного хозяйства функционируют системы, в которых порядок наступления события и времена выполнения действий не являются детерминированными, а зависят от случая. В качестве примеров таких систем можно назвать обслуживание больных машинами скорой помощи, работу такси по обслуживанию пассажиров, работу продавца в магазине по обслуживанию покупателей, работу справочной службы в городе. Такого рода системы принято называть системами массового обслуживания.

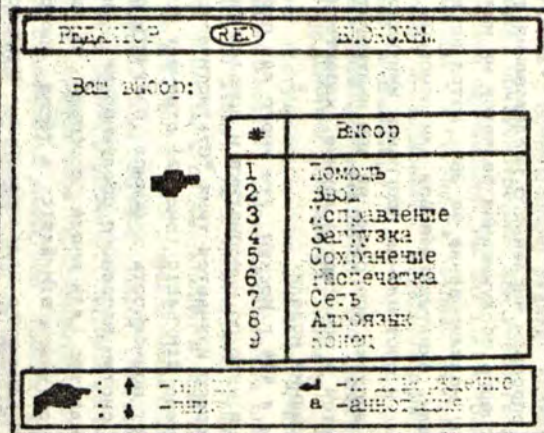
В общем случае системой массового обслуживания называют множество элементов (кассы магазина, такси, телефонные линии), на которые поступает поток процессов (покупателей, пассажиров вызовов), обслуживаемых по некоторому алгоритму.

Общим способом численного исследования функционирования систем массового обслуживания является их имитационное (статистическое) моделирование на ЭВМ. Метод заключается в построении модели, охватывающей все или основные свойства реальной системы, и составлении программы имитационного моделирования. При запуске программы на ЭВМ имитируется работа модели, собираются и печатаются данные, характеризующие качество функционирования модели.

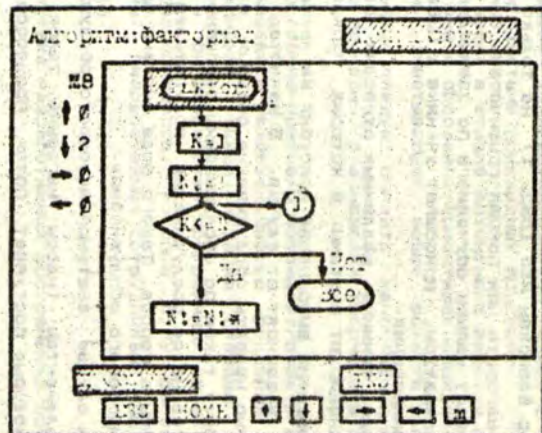
В школе № 404 г. Москвы уже много лет преподается экспериментальный курс по информатике и прикладной математике, в рамках которого изучается тема "Статистическое моделирование" [ 2 ]. Система SIM/BASIC (рис. 2) отвечает требованиям этого курса и дает возможность учащимся получить опыт моделирования разных систем массового обслуживания. Система SIM/BASIC содержит редактор для ввода программ на языке SIM/BASIC, предпроцессор языка SIM/BASIC, а также программу, обучающую основам имитационного моделирования. Учащимся легко освоить применение SIM/BASIC для анализа реальных ситуаций явлений, так как программирование основано на языке BASIC с добавлением специальных операторов, описывающих характерные для

# РЕДАКТОР БЛОКСХЕМ

Редактор блоксхем RED является средством обучения описанию алгоритмов в виде графических блоксхем. Редактор позволяет вводить, изменять, сохранять и распечатывать схемы алгоритмов, работать в сети класса. Редактор имеет две версии - для ЭМ учителя с принтером и дисководом и для ЭМ ученика. Широко используется принцип меню и подсказок, возможна демонстрация работы редактора на примере.



Экран выбора режимов



Рабочий экран режима МОДИФИКАЦИЯ

Рис.1. Редактор блоксхем.

# СИСТЕМА МОДЕЛИРОВАНИЯ SIM/BASIC

Система предназначена для имитационного моделирования систем массового обслуживания с целью расчета качества их функционирования. Программы составляются на языке SIM/BASIC, который является расширением языка BASIC. Предусмотрены режимы обучения, демонстрации и расчета. В режиме демонстрации на графическом экране ПЭМ показывается динамическое отображение поступающих процессов и их переходов из состояния в состояние.

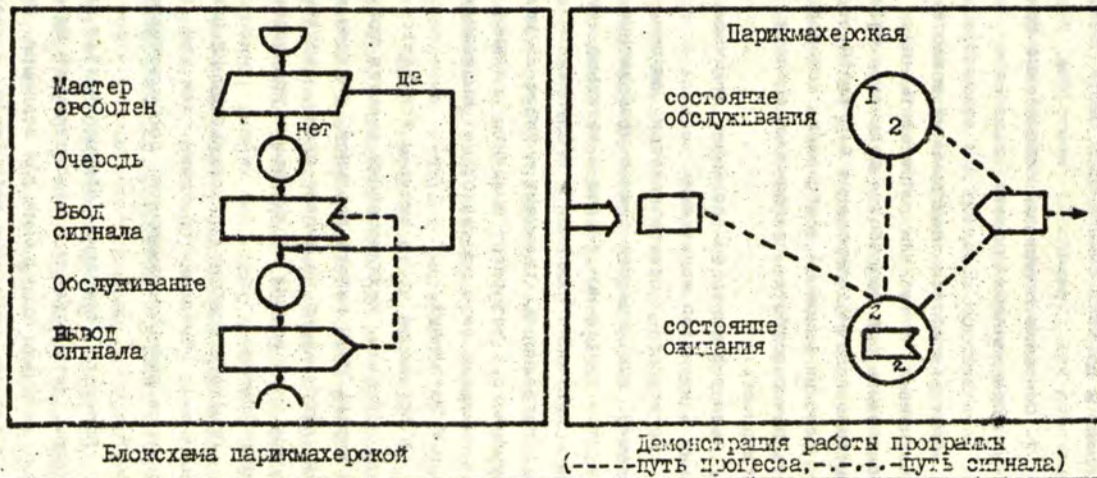


Рис.2. Демонстрация моделирования на SIM/BASIC.

имитационного моделирования действия (задержка, постановка в очередь и др.).

## 2. Основные принципы, соблюдаемые при разработке программных средств

Опыт разработки программных средств, а также работа со школьниками позволили определить ряд требований, которые соблюдались при разработке редактора блоксхем RED и системы имитационного моделирования SIM/BASIC.

Одно из основных требований, которое необходимо соблюдать при разработке программных средств для учебных целей - наглядность.

Редактор блоксхем RED имеет отличное эстетическое оформление и является наглядным.

Наглядность обеспечивается широким применением цветной графики, динамической сменой изображений. Одновременно используются зрительные образы и звуковые музыкальные средства для привлечения внимания и предупреждения об особых ситуациях. Наглядность повышает интерес и поддерживает внимание обучаемого, позволяет выделить главное и сосредоточить на нем внимание (это соответствует выделению "опорных сигналов" по В.Ф.Шаталову).

В обучающей части системы SIM/BASIC для иллюстрации работы программ: моделирования имеются три уровня: отображение на экране ПЭВМ работы реальной системы массового обслуживания, отображение на экране схемы действия модели и динамическое отображение обслуживания процессов во время выполнения программы.

Простота освоения и использования повышает эффективность применения обучающих программ, так как во время их работы основное внимание учащихся сосредоточено на предметной области.

Простота освоения обеспечивается наличием средств подсказок, автоматической демонстрацией возможностей на примерах, наличием упражнения для освоения. Но главную роль играет простота управления программой и задания данных. Здесь целесообразно использовать функциональные клавиши. Выбрать

клавиши для выполнения определенных функций программы желательно таким образом, чтобы обозначению клавиши ассоциировалось с выполняемым действием (например, для передвижения объекта подходят клавиши управления курсором).

Принцип "меню" освобождает пользователя от запоминания информации, не существенной для процесса приобретения знаний, но необходимой для контакта с компьютером. В настоящее время используются два основных типа "меню": а) с ответом номера функции; б) с передвижением указателя на выбранную функцию. Второй тип предпочтителен для применения в учебном процессе, предохраняет от ошибок и прост в управлении (лишь двумя клавишами курсора). Этот принцип используется в разработанных программных средствах.

Опыт компьютерных игр показывает, что демонстрация возможностей программы позволяет предугадать возможные действия программы, дает представления о работе программы и тем самым является мощным средством обучения. К сожалению, в программах для учебного процесса это средство используется недостаточно.

Для повышения интереса и отработки навыков целесообразно в обучающих программах по предметам использовать игровой метод обучения, например, в виде отдельного режима. Это также способствует закреплению учебного материала. В обучающей системе SIM/BASIC такой метод используется для изучения свойств случайных чисел. Например, программа генерирует случайные числа в заданном диапазоне, рисует числограмму, полученную в результате эксперимента, и показывает распределение случайных чисел согласно заданному закону распределения. Используя эту программу, ученик сам может проводить эксперимент любое число раз и наблюдать, как результаты эксперимента совпадают с теорией.

Возможность практического применения повышает активность, инициативу учащегося и способствует применению ЭЭМ в практической деятельности. Освоение программного средства в кружке или на уроках информатики само по себе лишь повышает уровень знаний обучаемого. Если школьник может освоенное программное средство применять для решения своих практических задач, например, по другому предмету в школе (для авто-

матризации лабораторных работ, решения домашних задания) или для своих увлечений (создания рисунков, банков данных спортивных результатов и т.д.), то интерес школьника значительно повышается, закрепляется учебный материал. Обучаемый на себе испытывает целесообразность использования компьютера.

Редактор блоксхем RED является также средством документирования алгоритмов. С его помощью ученик может создать архив алгоритмов.

Система SIM/BASIC содержит как обучающую часть, так и средства имитационного моделирования, позволяющие ученику решать и исследовать простые системы массового обслуживания.

Обеспечение активного диалога - необходимая требование к обучающей программе. Работая на ПЭВМ с обучающей программой, мы то классического диалога "учитель-ученик" совершается диалог "ученик-машина". Заменить диалог "учитель-ученик" ПЭВМ, конечно, не может, но тем не менее необходимо обеспечить по возможности более активный диалог. Самый простой вариант диалога "человек-машина" ограничивается ответами человека "да" или "нет" на задаваемые ему вопросы. Такие программы удобны для проведения контрольных работ, а также их можно использовать в качестве тренажера по пройденной теме. Такие программы желательны снабдить блоками, которые собирают статистику о встречающихся ошибках и оценивают результаты опроса.

Обучающие программы такого рода имеют общий недостаток - интерес к таким программным средствам быстро угасает. Можно сказать, что это программы одноразового пользования.

Наиболее активный диалог получается, если ученик может работать с динамическими объектами, управлять экспериментом, провести анализ полученных результатов, обращаться к базе данных для получения справочных данных, т.е. с помощью ЭВМ делать микрооткрытия и микроисследования.

Обеспечение детерминированного поведения программы в непредусмотренных ситуациях - это требование необходимо соблюдать при разработке любой программы, особенно обучающей. Существенным требованием к обучающей программе является обеспечение защиты программы от несанкционированных действий пользователя. Если пользователь совершил непредусмотренное

действие, желательно ему выдать подсказку, как следует продолжать работу.

По окончании работы обучающей программы необходимо восстановить системную среду - функциональные клавиши, цвета, размеры экрана и т.д.

## Выводы

Программное обеспечение должно обладать многими свойствами, которые трудно обеспечить одному лицу, поскольку при разработке требуются различные знания и навыки. Поэтому успеха можно добиться при сотрудничестве специалистов по программированию, педагогов, оформителя-дизайнера, психолога. Эти специалисты могут отработать все компоненты ПЭВМ: методику обучения и восприятия, звуковое сопровождение, цвет, зрительные образы, удобства при эксплуатации.

Для эффективного использования ПЭВМ необходимо иметь свободный доступ к ПЭВМ с необходимым периферийным оборудованием (принтером, цветным монитором с графическим экраном, дисковое устройство типа "винчестер", модемостроитель) и набором пакетов программного обеспечения для демонстрации, обучения и работы.

Программа должна быть апробирована в реальных условиях и имеет ценность только тогда, когда обратная связь (реакция учащихся, оценка процесса обучения методистом-предметником, эффективность процесса обучения) доказала ее жизнеспособность. Таким образом, обучающая программа проходит все стадии жизненного цикла - от момента осознания необходимости ее создания до момента появления программы, заменяющей рассматриваемую, или до исчезновения потребности в данной программе. Сопровождение и поддержка гарантирует ее качество и надежность. Ошибочным является мнение, что обучающая программа завершена после того, как выявлено ее соответствие начальной спецификации. Ценность обучающей программы выявляется только после ее применения на практике и как правило, влечет за собой необходимость ее доработки и сопровождения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бургин М. Проблемы преподавания основ информатики // Информатика и образование. 1987. # 4. С. 17-19.
2. Романов В. Информатика и прикладная математика. Программа для школ (классов) с углубленным изучением математики // Информатика и образование. 1987. # 6. С. 35-42.

Межузовский сборник научных трудов  
ЗВМ в образовании. Программное обеспечение  
1988, Рига: ЛГУ им. П. Стучки. С. 103-113

УДК 371.378.681.3.06

С. И. Павлов, Н. Н. Устинов

Вычислительный центр при ЛГУ им. П. Стучки

### РАЗРАБОТКА И ВНЕДРЕНИЕ ОБУЧАЮЩИХ ПРОГРАММ ДЛЯ ШКОЛ ШОТЛАНДИИ

В работе рассматриваются некоторые вопросы, связанные с организацией работ по созданию педагогических программных средств (ППС) и их распространению в школах Шотландии, выполняемых в рамках Национальной программы по микроэлектронике (НПМ) [ 1, 2 ]. Этот опыт, по-видимому, представляет значительный интерес для обеспечения успешного развития компьютеризации образования в СССР.

#### 1. Цели, задачи, организация и финансирование НПМ

Шотландия в рамках Великобритании отличается собственной системой образования. При населении 5,5 млн в Шотландии 2500 начальных (500 тыс. учащихся), 440 средних (400 тыс. уч.) школ и 190 школ (210 тыс. уч.) т.н. 3-го уровня, охватывающего дальнейшее среднее и начальное высшее образование (включая подготовку педагогов). Кроме того, в рамках единой системы образования Великобритании в Шотландии расположено 8 университетов (45 тыс. студентов). В 1979 г. во всех названных заведениях было 20 микрокомпьютерных систем, в 1986 г. - 12 тыс.

Основные задачи НПМ были сформулированы в 1983 г.:

- 1) развитие библиотеки ППС для школ;
- 2) развитие службы информации по внедрению и использованию микроЗВМ в школах;
- 3) разработка программы научно-исследовательских и опытно-конструкторских работ.

Основные цели реализации НГМ:

1) способствовать лучшему пониманию учащимися, студентами и педагогами возможностей применения микропроцессорной техники;

2) подготовка учебно-методических пособий и кадров для широкого обучения и переподготовки учителей по применению средств микропроцессорной техники в школах;

3) подготовка методических материалов для школьников по вопросам применения микроЭВМ;

4) использование средств микроэлектроники для помощи учащимся и студентам с физическими недостатками;

5) развитие системы обмена и распространения ППС и информации;

6) использование микроЭВМ в обучении и управлении учебным процессом школ.

Основным направлением выбрано применение микроЭВМ в школах в качестве средства обучения математике, программированию, микроэлектронике, физике, географии, экономике, бизнесу, управлению. Исследовались также возможности использования микроЭВМ в профориентации, обучении глухих, слепых и умственно отсталых детей.

Первоначально было утверждено 68 тем, за разработку каждой из них отвечали специально созданные региональные исследовательские центры, в помощь которым было подключено около 100 средних школ и вузов. Работа координировалась центральным советом при Министерстве образования.

Для осуществления НГМ первоначально выделено 320 тыс. фунтов стерлингов, а через год дополнительно еще 1 млн. в основном для разработки и производства ППС. Тем не менее ряд интересных направлений исследований остались без финансирования (объем предложения составил более 2 млн.).

При создании библиотеки ППС для школ была утверждена двухуровневая система: 1) включение в библиотеку и распространение (для удовлетворения прогрессирующей потребности) любых сколько-нибудь пригодных программ без достаточной документации и апробации; 2) включение в библиотеку разработанных и закупленных программ, прошедших экспертизу (на соответствие сформулированным требованиям), документированных.

технически совершенных.

Решение о бесплатном распространении ППС, создаваемого в рамках НГМ, было принято в 1980 г.

## 2. Развитие компьютерной технологии обучения (КТО)

Решающими факторами, определяющими прогресс в применении ЭВМ в обучении, являются: 1) улучшение технического уровня средств вычислительной техники (СВТ); 2) организация массового производства ППС и их сопровождения; 3) определение перспектив КТО.

Технический уровень СВТ - один из основных факторов, влияющих на КТО, причем сложность состоит в том, что возможности СВТ стремительно возрастают по мере развития микроэлектроники. Появление новых компонентов микропроцессорной техники (каких, нельзя сказать заранее; в настоящее время - это устройства ввода "мышь", синтезатор речи, интерактивное видео и др.) оказывает существенное влияние на КТО.

При определении перспектив развития КТО акцент сделан на методологию (дидактику, педагогику, психологию), изучение границ применимости КТО, которые определяются возможностями микроЭВМ и в особенности качеством и возможностями ППС.

При изучении стабильности КТО как подхода к образованию принимаются следующие соображения. На определенном этапе развития общества была принята определенная модель образования и разработана система обучения для реализации этой модели. Результатом развития системы обучения является создание более прогрессивных технологий и производств, определяющих прогресс развития общества, который сказывается и на системе образования. Развитие средств микроэлектроники и их внедрение в процесс обучения привело к возникновению эффективной обратной связи, приводящей к более быстрому влиянию современных технологий на развитие системы обучения (рис. 1). Однако, следует принимать меры, чтобы именно система образования оказывала существенное влияние на разработку современных технологий.

Существенную роль в успехе КТО и ее стабильности как ~~принципа~~ играет отношение учителей к КТО, и особенно ~~требует~~

Рис. 1. Образование и создание современных технологий.

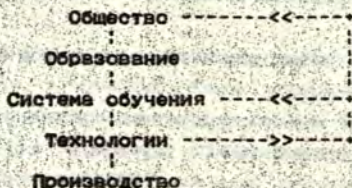
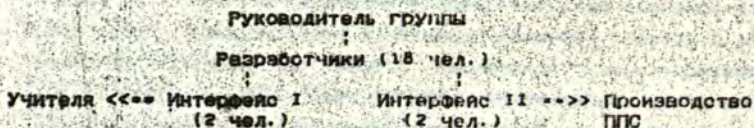


Рис. 2. Стадии разработки ППС.



Рис. 3. Структура группы по разработке ППС.



шая специальной подготовки способность учителей участвовать в КТО, умение оценивать образовательную ценность ППС.

### 3. Организация процесса разработки ПО

Первоначально ППС создавались учителями в инициативном порядке. Отсутствие проработанной образовательной концепции, а также опыта программирования у учителей привело к появлению технически и методически несовершенных обучающих программ. Предпринимались попытки улучшения качества этих ППС (усилиями профессионалов) для последующего массового распространения. Однако по мере накопления опыта очевидной стала необходимость специальной подготовки учителей.

Специалисты по разработке программного обеспечения (ПО) считают, что более простой является технология разработки ПО, когда разработчик создает ПО целиком - от начала до конца, т.е. сценарий и спецификация программы создаются параллельно с программой. По такой технологии создано значительное число программ.

Однако учителя считают, что при тщательной предварительной (до начала программирования) проработке идей уровень программ гораздо выше, т.е. учителями признается значение качественного сценария и спецификации программы. Нежелание предварительно разрабатывать сценарий опирается на представление о программе как способе реализации вычислительного алгоритма, тогда как обучающая программа - это средство, имеющее определенную образовательную ценность.

Некоторая часть учителей, имеющих необходимую подготовку, занимается разработкой ППС. Тем учителям, которые не хотят самостоятельно программировать, предоставлена возможность (как в плане информирования о потребности в разработках, так и в плане координации усилий учителей) участвовать в создании методических разработок и пособий по применению компьютеров и ППС.

Наиболее перспективным при разработке ППС признан путь, состоящий в объединении различных специалистов и разделении сфер их труда: разработка сценариев ППС компетентными в вопросах методологии обучения учителями; реализация сценариев

профессиональными программистами.

Следует отметить, задача учителя состоит не столько в представлении программисту описания возможностей будущей программы, сколько в обеспечении сбалансированности образовательных и технических компонентов программы, что возможно при разработке сценария совместно с программистами.

Процесс разработки ППС можно представить в виде следующей схемы (рис. 2).

**Пр е с п е ц и ф и к а ц и я** (эскизное проектирование). Ключевым моментом в процессе создания ППС является обсуждение образовательной ценности проектируемого ППС - целей, предметной области, методологии, критериев качества. Стадия т.н. эскизного проектирования позволяет систематизировать представления разработчика, создать надежный образовательный фундамент для создания ППС и документации, а также облегчить принятие решения, каким сценариям отдать предпочтение для производства. Эта стадия разработки обеспечивается в основном специалистами в области образования, поскольку аргументы в пользу разработки конкретного ППС имеют образовательный, а не технический характер.

В то же время разработчик сценария должен иметь достаточно подробное представление о компьютере как средстве реализации поставленных педагогических целей.

**С п е ц и ф и к а ц и я и п р о г р а м м и р о в а н и е** (техническое и рабочее проектирование). После принятия решения о создании конкретного пакета начинается этап, в котором активно участвуют программисты.

Как показывает опыт, для успешной работы по непосредственному программированию системные аналитики (организующие работу программистов), программисты должны глубоко проникнуть в образовательные аспекты проектируемого средства. Это достигается подключением программистов к разработке ППС уже на этапе подготовки сценария. В то же время программист независимо проводит работу по программированию "меню", ключевых в содержательном плане кадров, кадров с динамической ("оживленной") графикой. Результаты этой работы используются учителями при работе над сценарием, что позволяет внести необходимые коррективы. Предварительный просмотр модулей про-

граммы наряду с широкими дискуссиями между разработчиками сценария и программистами положительно сказывается на содержании и качестве создаваемого ППС, т.е. специфицирование и программирование являются взаимосвязанными, параллельно проводимыми этапами процесса разработки ППС.

На этой стадии разработки ППС учитываются следующие аспекты применения компьютеров:

1) образовательные: а) выбор целей и приоритеты в разработке ППС; б) преемственность и совместимость содержания создаваемого ППС с содержанием других тем, а также учет альтернативных возможностей изучения темы, в т.ч. без применения компьютера; в) возможность применения моделей обучения, позволяющих учесть результаты предварительного диалога с обучаемым с целью динамического изменения уровня сложности подачи материала, анализ затруднений обучаемого; г) трассировка программы, дающая возможность фиксировать "историю" контактов обучаемого с программой, оценивать приобретенные знания, для применения учителем в управлении обучением; д) методические материалы для учителя и ученика; е) документация ППС и другие вопросы.

2) технические: а) легкость применения; б) режим помощи; в) проверка вводимой информации; г) сообщение об ошибках и возможность их корректировки; д) блокировка клавиш, не используемых в диалоге; е) "дружественный" диалог с обучаемым; ж) планировка экрана с использованием графики; з) применение элементов унификации (стандартных ответов, кадров); и) применение звука и др.

**Д о к у м е н т а ц и я.** Для обеспечения эффективности применения ППС в комплексе с другим ПО и в "независимом" режиме ППС комплектуется методическими материалами. ППС, рассчитанные на интерактивный режим работы учителя с учениками, сопровождаются различными для учителя и учеников инструкциями и заметками.

В документацию, сопровождающую ППС, входят, как минимум, следующие пособия с изложением перечисляемых вопросов:

1) указания учителю: а) цель ППС; б) категория и контингент обучаемых; в) конспективное изложение изучаемого материала; г) уровень предварительных знаний учащихся; д) орга-

низация класса (технический аспект); е) дополнительные средства обучения.

2) Заметки для учащегося: а) инструкция по запуску программы б) информационные листки; в) рабочие материалы.

3) Технические данные: а) описание переменных, используемых в программе; б) описание структуры и организации файлов в памяти ЭВМ и на внешних носителях; в) возможности и границы применимости программы.

Основная работа по составлению указаний учителю может быть выполнена на стадии преспецификации. На последующих стадиях разработки ППС эти указания без существенной переработки можно модифицировать.

Конечно, есть риск, что на стадии специфицирования и программирования в результате совместной работы учителя и программиста первоначальный замысел будет настолько изменен, что придется практически заново разрабатывать документацию к разрабатываемому ППС. Однако указания учителю подвергаются при этом наименьшим изменениям.

Для составления сопровождающей документации разработано руководство, принятое в качестве стандарта.

**А п р о б а ц и я** (опытная эксплуатация). После завершения программирования и составления документации организуется апробация созданного ППС в условиях реального класса. Проверка проводится сначала учителями, участвовавшими в разработке ППС, затем учителями, знакомящимися с ППС впервые. Опытная эксплуатация до начала тиражирования и распространения ППС позволяет внести необходимые изменения в ППС и документацию.

Экспертиза при сдаче ППС проводится при участии членов экспертного совета, функционирующего в рамках ИПМ.

**С о п р о в о ж д е н и е.** Как правило, разработчик теряет интерес к своей программе после ее выпуска массовым тиражом. Чтобы разработчики сопровождали ППС, необходимо предпринимать специальные меры, по следующим причинам:

1. Кроме разработчиков никто не в состоянии качественно устранить ошибки в программах и документации, которые обнаруживаются при широком распространении программ.

2. При внедрении ППС в школы часто возникает потребность

в адаптации (перепрограммировании) конкретного пакета программ на микроЭВМ другого типа. Адаптация не является чисто технической работой, поскольку микроЭВМ существенно различаются по архитектуре и программно-реализуемым возможностям. К тому же перепрограммирование может привести к значительной переработке пакета и документации, а иногда к полной потере образовательной ценности.

3. Накопление опыта применения ППС может привести к появлению новых направлений использования, ранее не предусмотренных разработчиками. Выпуски документации последующих версий программ, как правило, содержат дополнительные рекомендации по применению пакета программ в приложениях.

4. Опыт применения компьютеров для поддержки процесса обучения показал, что ППС предоставляет учителю и обучаемому ряд методологических и концептуальных возможностей, не существующих при традиционных формах обучения (например, построение учащимися альтернативных гипотез с применением баз данных и последующим их использованием в интерактивном режиме для анализа и отбора). В таких случаях основной набор документации необходимо дополнить материалом, позволяющим учителю при подготовке к занятию заранее продумать методологические и концептуальные возможности ППС.

Для подготовки учителей к участию в разработке ППС созданы различные унифицированные пакеты программ, обеспечивающие повышение квалификации учителей с учетом уровня их подготовки. Пакеты программ основаны на демонстрации возможностей компьютера и технологии разработки ППС на различных ее стадиях и нацелены на развитие навыков создания ППС.

Структура группы по разработке ППС. В группу (см. рис. 3) входит 23 человека. Наибольшая по численности (18 чел.) подгруппа, состоящая из трех команд по одному системному аналитику и пяти программистам в каждой. Подгруппа из двух человек согласует работу учителей, разрабатывающих сценарии, с подгруппой программирования по вопросам реализации выбранной образовательной стратегии при создании ППС и документации. Подгруппа из двух человек осуществляет различные технические работы, связанные

о производством ППС.

Информация о ППС и их распространение. В рамках НГМ разработана и принята концепция свободного распространения и обмена ППС. Для обеспечения пользователей оригиналами ППС и документацией создана сеть региональных центров.

Для удовлетворения запросов потенциального пользователя в информации о ППС (педагогической, технической), разработанных как в рамках НГМ, так и других региональных центрах, колледжах, школах, университетах, фирмах, а также в частном порядке, наиболее перспективным считается внедрение в систему образования электронной почты.

Пока практикуются передвижные демонстрационные выставки-семинары, и распространяются два периодических информационных бюллетеня: один - на 3-4 страницах - издавается два раза в месяц и распространяется во все школы бесплатно; второй распространяется два раза в год по подписке и содержит в развернутом виде сведения из экспресс-бюллетеней, а также тематические обзоры, дискуссионные статьи и др.

В пределах Шотландии практикуются как централизованные поставки ППС во все школы, так и поставки отдельных ППС по индивидуальным заявкам школ и учителей. Постоянное увеличение объема распространяемых материалов и числа заинтересованных школ приводит к определенным затруднениям: в 1985/86 учебном году распространено 35 тыс. дискет и один миллион страниц документации (потребность составляет 200 тыс. дискет и 4 млн страниц документации).

По оценкам, возможности экстенсивного роста традиционных схем тиражирования ППС близки к пределу (при бесплатном распространении). В настоящее время проводится изучение цифровых технологий систем связи.

#### 4. Направления исследований

Применение ЭВМ в обучении является быстро прогрессирующим направлением. Стремительно возрастают возможности микропроцессорной техники: появляются новые периферийные устройства - специальные клавиатуры, синтезаторы речи, видео-дис-

ки развиваются программные средства, облегчающие диалог с ЭВМ для учителя и учащегося. "Концептуально свободные" пакеты программ (электронные таблицы, системы управления базами данных с развитыми языками запросов) обеспечивают работу пользователей с различными типами данных. "Авторские языки" позволяют привлечь учителей, не умеющих программировать, к разработке обучающих систем. Перспективно применение в обучении языков Logo, Prolog.

Однако опыт применения ЭВМ в обучении поставил ряд проблем, в т.ч.: 1) ППС не обеспечивают диалог с ЭВМ на естественном для учащегося языке; 2) возможности ППС для обучения конкретной предметной области не велики и быстро исчерпываются прогрессирующим учащимся; 3) в зависимости от информации, получаемой от учащегося, ППС не могут менять стратегию обучения и обеспечивают адекватную реакцию на сравнительно узкий набор возможных ответов учащихся; 4) ППС не всегда реализуют адекватную модель учащегося.

По мере увеличения числа компьютеров в школах и расширения их возможностей актуальным является направление исследований педагогических аспектов различных подходов в массовом внедрении компьютерной технологии обучения различных общеобразовательных предметов. Не решены еще вопросы создания наиболее адекватной для обучения программной среды, включающей операционные системы, языки программирования, ПО общего назначения. Перспективна разработка экспертных систем для применения в обучении.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Presentation and Documentation Standarts. Glasgow: Scottish Microelectronics Development Programme (SMDP). 1985. 52 p.
2. Microcomputers in Scottish Schools. A national plan. Glasgow: SMDP. 1985. 88 p.

Межвузовский сборник научных трудов  
ЭВМ В ОБРАЗОВАНИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
1988, Фига: ЛГУ им. П. Стучки. С. 114-127

УДК 37.01:007+371.3+378.147+519.6+681.3.06

С. И. Павлов, В. Л. Цилевич

Вычислительный центр при ЛГУ им. П. Стучки

ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ МЕТОДА МАТЕМАТИЧЕСКОГО  
МОДЕЛИРОВАНИЯ ДЛЯ РАЗРАБОТКИ ПЕДАГОГИЧЕСКИХ  
ПРОГРАММНЫХ СРЕДСТВ

В настоящее время на всех этапах системы непрерывного образования характерна возрастающая роль использования вычислительной техники. В применении ЭВМ в обучении выделяются два направления:

- изучение самой ЭВМ и основ программирования (ЭВМ как объект изучения);
- подготовка пользователя готовых программных средств (ЭВМ как инструмент профессиональной деятельности в самых различных предметных областях: сбор и обработка данных и информации, управление технологическими процессами, управление экспериментом, автоматизация проектирования, математическое моделирование и т. д.).

По мере компьютеризации общества все большее число специалистов и служащих будут использовать ЭВМ в своей повседневной профессиональной деятельности, в то время как разработка технических средств и программного обеспечения останется уделом сравнительно узкого круга профессионалов. Следовательно, второе из указанных выше направлений применения вычислительной техники должно стать преобладающим.

Важным частным случаем использования ЭВМ как инструмента является применение ее в учебном процессе в качестве технического средства обучения. Эффективность применения ЭВМ в учебных курсах по различным предметам и для различных этапов системы непрерывного образования определяется научным и методическим уровнем специализированного программного

обеспечения - педагогических программных средств (ППС).

Для сегодняшнего состояния дел в этой области характерны:

- недостаточная теоретическая база, отсутствие единой концепции разработки ППС;

- слабая координация практических разработок, ведущихся разными организациями;

- ориентация, как правило, на сравнительно небольшие пакеты или программные комплексы по одной или нескольким темам какого-либо учебного курса.

Одним из наиболее часто применяемых подходов, используемых при разработке ППС, является моделирование. В [ 1-3 ] рассмотрены некоторые дидактические требования к учебным моделирующим программам. Однако единая методология проектирования и реализации ППС на базе моделирования (ППС+М) пока отсутствует.

Основной целью настоящей работы авторы считали:

- обсуждение возможностей применения и специфики проектирования, реализации и эксплуатации ППС+М;

- анализ некоторых общих требований, обеспечивающих дидактическую эффективность ППС+М, в методическом, предметном, сценарном, технологическом и организационном плане с точки зрения разработчика ППС+М.

## 1. ППС на базе моделирования

Моделирование с гносеологической точки зрения является фундаментальным методом познания, а модель - весьма универсальным средством познания. Моделирование как метод применяется с незапамятных времен (в том числе и в обучении); но только появление и развитие вычислительной техники вызвало стремительный прогресс в этой области как в количественном (расширение сферы применения моделирования), так и в качественном отношении (возможность постановки и решения нелинейных задач). Вычислительные возможности ЭВМ обусловили появление теории математического моделирования и вычислительного эксперимента [ 4 ], бурное развитие как аналитических, так и численных методов решения возникающих задач.

Значительная часть ППС+М ориентирована на выполнение

чисто иллюстративных функций. Эти программные средства используются в качестве альтернативы натурному демонстрационному эксперименту [ 5,6 ]. Вычислительные и графические возможности ЭВМ позволяют визуализировать ход и результаты экспериментов, натурная реализация которых является слишком дорогостоящей или вообще невозможной в условиях учебного процесса.

Особый интерес, с нашей точки зрения, представляют ППС+М, построенные на базе математического моделирования (ППС+ММ), например [ 7,8 ], т.е. ППС, реализующие схему вычислительного эксперимента А.А.Самарского [ 4 ]. Последовательное применение ППС+ММ в обучении позволяет сознательно сблизить методологию изучения учебного курса с методологией научно-исследовательской работы.

Такой подход требует учета ряда факторов.

## 2. Уровни подготовки пользователей ЭВМ.

ППС на базе моделирования на различных этапах системы непрерывного образования

Уровень подготовки пользователя ППС+М определяется степенью владения методологией и техникой моделирования. С этой точки зрения можно выделить три категории обучаемых:

1) не владеющие ни методологией, ни техникой моделирования как метода (например, младший школьник; отметим, что в своей учебной деятельности и в быту пользователь данной категории тем не менее широко, хотя и неосознанно, использует умозаключение по аналогии, лежащее в основе моделирования);

2) владеющие методологией, но не владеющие техникой моделирования (например, студент или специалист-гуманитарий);

3) владеющие как методологией, так и техникой моделирования, но сознательно не применяющие последнюю (например, инженер, исследователь).

При изучении основных законов отдельных дисциплин и формировании характерных для них подходов и мышления ЭВМ не способна конкурировать с традиционными формами обучения [ 9 ]. Наиболее эффективные условия для использования ЭВМ в

учебном процессе возникают на этапе приложений основных принципов отдельных дисциплин, в частности, в форме моделирования [ 9 ].

В начальной школе обучение направлено в основном на выработку разнообразных умений и навыков (грамотно писать, владеть счетом и т.п.), а также на сообщение обучаемым некоторого набора отдельных фактов и описаний явлений (например, курс природоведения). Для этих целей весьма эффективно применение иллюстративных ППС+М, программ-тренажеров, ориентированных на первую категорию пользователей. В подобных ППС+М в максимальной степени должен быть реализован принцип наглядности за счет применения широкого набора средств визуализации (цвет, звук, динамика и пр.), ППС+М должны быть достаточно просты как с точки зрения содержания сценария, так и в общении с программой, должны содержать игровые элементы. Для ППС+ММ такого рода сама модель скрыта от пользователя, возможность варьирования параметров модели не является обязательной, а вышеупомянутый принцип наглядности относится к постановке задачи и представлению результатов моделирования.

Для процесса дальнейшего обучения (5-8 классы) характерно знакомство обучаемого с основными законами ряда предметов (физика, алгебра, химия, биология и др.), формирования представления о свойственной им "идеологии". Далее, в старших классах, в вузе, в системе повышения квалификации, обучение идет "вширь" и "вглубь", причем обучаемый уже владеет "технологией" освоения новых для себя дисциплин, в том числе и навыками самостоятельной работы, поэтому эффективность применения ППС+ММ, начиная со старших классов средней школы, постоянно возрастает. ППС+ММ, ориентированные на пользователя второй категории, должны позволять варьировать в широких пределах параметры модели, модифицировать саму модель. С целью облегчения интерпретации результатов моделирования ППС+ММ должны предоставлять пользователю возможность как (результатов) численной обработки, представления их в различной форме.

В системе повышения квалификации специалистов (например, преподавателей-естественников) методика обучения, основанная на использовании ППС+ММ, может играть ведущую роль. ППС+ММ,

ориентированные на пользователя третьей категории, должны включать развитые инструментальные средства, позволяющие специалисту удобно и быстро проводить моделирование задачи в требуемой постановке. Такие ППС+ММ по существу являются универсальным программным обеспечением, применимым как для учебного моделирования, так и в исследовательских целях.

Таким образом, методическая роль ППС+ММ по мере продвижения обучаемого по ступеням системы непрерывного образования возрастает.

### 3. Предметные области

Эффективность применения ППС+ММ в изучении той или иной дисциплины определяется в основном тем, насколько и каким образом в самой дисциплине используется метод математического моделирования. С этой точки зрения можно выделить три класса дисциплин.

1. В качестве многообещающих областей приложения ППС+ММ необходимо назвать предметы естественно-научного цикла - механику [ 10 ], электродинамику [ 3 ], химию [ 11 ], несписательные разделы биологии, технические дисциплины. В ряде случаев ППС+ММ не просто эффективны, но и существенно расширяют дидактические возможности преподавания той или иной дисциплины, раздела (молекулярно-кинетическая теория, теория механизмов и машин).

2. Возможности применения ППС+ММ в предметах общественного и гуманитарного циклов пока представляются довольно ограниченными ввиду весьма узкого использования этими дисциплинами метода математического моделирования. Однако, широкое внедрение ППС+ММ в учебный процесс при дальнейшем повышении качества и доступности средств вычислительной техники будет стимулировать "математизацию" этих дисциплин, в том числе и приложения моделирования.

3. Особое значение моделирование имеет в математических курсах [ 12 ] и в курсе информатики [ 13 ] (систематическое применение метода математического моделирования в [ 13 ] предложено рассматривать в качестве основы компьютеризации школьного образования). Если по отношению к другим предметам

ППС+ММ и метод математического моделирования в целом выступают в качестве средства для получения новых знаний в области самого предмета, то в названных курсах сам метод является предметом изучения. Освоение метода математического моделирования на необходимом (для данного этапа обучения) уровне является одной из основных целей курса информатики. При таком подходе роль курса информатики значительно повышается: курс дает общую методологическую базу, общий подход, используемый в виде конкретных приложений в ряде других предметов. Последовательное применение ППС+ММ влечет за собой необходимость тесной увязки учебных планов по курсу информатики с другими дисциплинами, а также, по-видимому, ведет к перераспределению учебного материала и пересмотру состава и содержания ряда учебных курсов. При этом обучение работе на ЭВМ с необходимым программным обеспечением и изучение элементов программирования целесообразно перенести на более ранние этапы обучения.

Отметим, что в существующем курсе информатики моделированию, как методу и как важнейшей области применения ЭВМ, уделяется явно недостаточное внимание. Нельзя говорить о компьютерной грамотности без овладения основами методологии и техники моделирования.

#### 4. Различные формы учебных занятий

Применение ППС+ММ на лекционных занятиях связано с частичной заменой натурального демонстрационного эксперимента моделированием, причем активный диалог обучаемого с ЭВМ не применяется. ППС+ММ для этих целей также могут использоваться, но их возможности остаются нераскрытыми.

Применение ППС+ММ для практических и самостоятельных работ значительно эффективнее. При такой форме процесс обучения полностью аналогичен процессу научного познания - с той разницей, что получаемые знания являются субъективно новыми. Используемые для таких целей ППС+ММ должны обеспечивать пользователю возможность самостоятельной реализации широкого на-

бора моделей - как "истинных", так и "ложных" (путем выбора с помощью "меню", описания в виде директив командного языка или построения с помощью инструментальных средств - в зависимости от уровня подготовки обучаемого), а также получения адекватных результатов численного эксперимента удобными для пользователя средствами и в удобных для пользователя формах представления, облегчающих интерпретацию.

Наибольшее число существующих сегодня ППС+ММ ориентированы на использование в рамках лабораторного практикума. Объединение компьютерного вычислительного эксперимента с традиционным натурным позволяет обеспечить единство экспериментальной и теоретической деятельности обучаемых [ 7 ]. ППС+ММ для лабораторного практикума позволяют реализовать и сравнивать результаты двух различных вычислительных экспериментов: по обработке данных натурального эксперимента [ 4 ] и по теоретическому (на базе математической модели) исследованию объекта или явления.

Перспективным представляется и использование ППС+ММ в качестве средства контроля за усвоением учебного материала. В этом случае особо важны средства документирования диалога, контроля времени, наличие "барьерных точек", а также средства, обеспечивающие возможность оперативной связи преподавателя с каждым контролируемым, что наиболее просто и удобно реализуется в случае локальной сети.

### 5. Роль преподавателя

Широкое внедрение ППС+ММ в учебный процесс ни в коей мере не влечет за собой устранения учителя из этого процесса. Тем не менее, функции преподавателя должны меняться: по мере продвижения обучаемого по ступеням образовательной системы роль учителя приближается к роли научного руководителя. Учитель освобождается от ряда рутинных операций, оставляя за собой содержательное общение с обучаемым на "стратегическом" уровне, а также элементы обучения, требующие индивидуального подхода. В перспективе, при включении в состав ППС+ММ элементов программной архитектуры экспертных систем и баз знаний [ 10 ], еще больше возрастет удельный вес межличностного

общения учителя с обучаемым на эмоциональном уровне, воспитательной (а не дидактической) деятельности.

#### 6. Реализация основных принципов дидактики

Педагогическая эффективность ППС (в том числе и ППС+ММ) обеспечивается за счет соблюдения при разработке ППС известных дидактических принципов. Однако требования к ППС+ММ не являются механическим объединением требований к ППС и пакетам по моделированию. Специфика учебного моделирования определяет изменение критериев важности тех или иных требований, а также появление новых. Рассмотрим практические требования к ППС+ММ, следующие из основных принципов дидактики.

**И н ф о р м а т и в н о с т ь** обучения с применением ППС+ММ достигается в результате оптимального сочетания двух противоречивых факторов:

- максимального соответствия используемой модели физической реальности (адекватности модели).
- простоты и доступности рассматриваемой модели для данного уровня подготовки обучаемых.

Следовательно, для реализации этого принципа необходимо разрабатывать не обучающие программы "вообще", а еще на стадии проектирования четко определить место данной ППС+ММ в соответствующем учебном курсе.

Особенно важно для оценки качества ППС+ММ такое тесно связанное с информативностью качество, как **н а г л я д н о с т ь**. В тех случаях, когда ППС+ММ используются лишь для передачи обучаемому субъективно новых для него знаний в предметной области, и не ставится задача осаледения самой методологией моделирования (либо не ставится задача сознательного применения этой методологии), т.е. ППС+ММ ориентированы на первую категорию пользователей, требование наглядности в применении к ППС+ММ означает наглядность постановки задачи (формулирования модели), а также представления результатов моделирования. Средства псвышения наглядности известны: визуализация с максимальным использованием графических возможностей имеющейся техники, применение цвета, звука, динамики, оптимальное с точки зрения восприятия обучаемым размещение

на экране дисплея текстовой и графической информации, выбор оптимального размера символов и т.п. Впрочем, по мере возрастания сложности рассматриваемых задач могут потребоваться дополнительные усилия для наглядного представления результатов моделирования [ 14 ].

Когда использование ППС+ММ подразумевает включение в число задач обучения осознанное применение математической модели и интерпретацию результатов, возникает вопрос о наглядности самой модели. С точки зрения гносеологии, наглядность идеальной (в том числе математической) модели определяется взаимодействием чувственного образа, порождаемого этой моделью в сознании обучаемого, с уже существующей в сознании системой чувственных образов. Таким образом, это подтверждает сделанный выше вывод о необходимости ориентации каждого ППС+ММ на строго определенное место в данном учебном курсе. Кроме того, отсюда следует, что форма представления и порядок формулирования математической модели, помимо общих требований наглядности, должны удовлетворять требованиям наглядности "содержательной": надо максимально помочь обучаемому уяснить, какие и каким образом свойства прототипа (объекта) отражены в рассматриваемой математической модели.

Отметим, что для ряда естественно-научных областей возникает проблема наглядности самого объекта моделирования, что требует включения в состав ППС+ММ определенной "предмоделировочной" стадии, обеспечивающей наглядность "доматематической" (например, физической) модели.

Активность обучения обеспечивается развитыми средствами диалога с ППС+ММ. Здесь имеется оптимальное соотношение между расширением спектра предоставляемых обучаемому возможностей и простотой общения с программой. Наиболее перспективным представляется вариант, при котором пользователь имеет возможность работать с программой как с использованием "меню" (с указанием выбранной позиции при помощи курсора), так и с помощью командного языка. Такой подход дает возможность разрабатывать ППС+ММ достаточно широкого диапазона применения - для всех категорий пользователя.

Принцип индивидуализации обучения, как уже отмечалось, требует для своей эффективной реализации

применения технологии искусственного интеллекта [ 15 ].

Построенные для целей обучения интеллектуальные расчетно-логические системы позволяют конструировать нужную математическую модель и выбирать нужный вычислительный алгоритм из имеющихся блоков, причем общение с системой ведется на удобном для пользователя языке. Дальнейшим развитием применения методов искусственного интеллекта в ППС+М являются гибридные системы, в состав которых, помимо расчетно-логической системы, включается экспертная система. Таким образом, наряду с математическим используется и логико-лингвистическое моделирование (для организации диалога, формирования и оценки математической модели, интерпретации результатов математического моделирования).

В наиболее распространенных на сегодня ППС+М индивидуализация достигается либо за счет участия преподавателя в общении обучаемого с программой (в режиме локальной сети), либо более грубо - за счет размещения в программе некоторого числа "барьерных точек" для контроля усвоения обучаемым уже "пройденного" материала.

Осознанное применение обучаемым принципов математического моделирования (выделение основных характеристик рассматриваемого объекта, абстрагирование от несущественных, второстепенных свойств, коррекция математической модели по результатам численного эксперимента и т.д.) является по существу практическим воплощением принципа **н а у ч н о с т и** обучения. С методической точки зрения это означает четкую "дозировку" скрытой в программе и открытой для пользователя информации, а также средства комментирования и поддержки ППС+М не на предметном, а на методическом уровне - в зависимости от места данной программы или комплекса как в предметном курсе; так и в системе обучения в целом. Таким образом, помимо знания в определенной предметной области, обучаемый расширяет представления о возможностях и закономерностях познания вообще (и с роли ЭВМ в этом процессе, в частности).

## 7. Некоторые аспекты проектирования и разработки ППС+ММ

При проектировании и реализации ППС+ММ необходимо учитывать ряд требований, обусловленных спецификой учебного моделирования. Рассмотрим некоторые из них.

1. Структурно в составе ППС+ММ можно выделить три компонента:

- блок диалога с обучаемым;
- блок моделирования;
- блок визуализации входных и выходных данных.

Для повышения эффективности ППС+ММ надо стремиться к максимальной унификации составных частей и блоков. Такая унификация дает возможность существенно расширить диапазон применения программных средств, облегчает их модификацию, разработку новых программных средств, упрощает проведение многовариантного моделирования.

2. Если унифицировать блок диалога с обучаемым не представляется возможным, необходимо выработать единые "конвенциональные" правила общения обучаемого с ЭВМ и строго руководствоваться ими при разработке ППС+ММ, причем желательно не только в рамках данного предметного курса (например, единая форма "меню", форма запроса на "подсказку", сообщение о паузе и т.п.). Другими словами, техника диалога с ЭВМ должна быть не столько простой, сколько единообразной, тогда внимание обучаемого естественным образом будет сосредоточено на содержательной стороне дела. Появления новых "слов" в диалоге по мере расширения возможностей применяемых ППС+ММ не потребует отказа от старых навыков и выработки новых.

3. Для ППС+ММ точность моделирования не имеет первостепенного значения, важна оперативность получения качественно верных результатов [ 7 ]. Это необходимо учитывать при выборе численных методов, используемых в ППС+ММ.

4. Режим реального времени подходит лишь для ограниченного круга рассматриваемых процессов. Кроме того, работа в таком режиме требует довольно высокого уровня подготовки пользователя. Следовательно, ППС+ММ должны обладать развитыми средствами масштабирования временных интервалов [ 10 ].

5. С точки зрения удобства многовариантного моделирования необходимо обеспечить возможность совмещения графического представления результатов для различных вариантов или моделей, например, за счет многооконности.

## В ы в о д ы

1. Последовательное применение метода математического моделирования в изучении ряда предметных курсов способно заметно повысить дидактическую эффективность ППС и может привести к необходимости пересмотра состава и содержания существующих курсов.

2. Широкое внедрение ППС+ММ в учебный процесс приводит к возрастанию роли курса информатики, при этом одной из основных задач этого курса становится освоение основ метода математического моделирования как методологической базы для изучения других дисциплин.

3. Разрабатываемые ППС+ММ должны быть четко ориентированы на определенные места как в учебном предметном курсе, так и в системе обучения в целом. Методическая ценность ППС+ММ для обучаемого повышается по мере продвижения по ступеням системы непрерывного образования.

4. Для эффективного формирования моделей, используемых в ППС+ММ, интерпретации результатов моделирования, а также организации диалога, обеспечивающей индивидуализацию процесса обучения, необходимо применение элементов техники искусственного интеллекта.

5. При разработке ППС+ММ возникает специфические проблемы реализации дидактического принципа наглядности, связанные с необходимостью наглядного представления математической модели, наглядностью процесса интерпретации результатов моделирования, в ряде случаев - наглядностью процесса построения модели.

6. Для ППС+ММ точность моделирования имеет второстепенное значение по сравнению с оперативностью получения качественно верных результатов.

7. Для эффективного и широкого применения ППС+ММ необходима унификация и стандартизация диалога обучаемого с ЭВМ как на сценарном ("конвенциональном") уровне, так и на уровне использования унифицированных программных блоков.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Фокин М. Л. Дидактические требования к учебным моделирующим программам на ЭВМ // Основные аспекты использования информационной технологии обучения в совершенствовании методической системы обучения. М.: НИИ СИМО АПН СССР. 1987. С. 37-54.
2. Фокин М. Л. Дидактические свойства учебных моделирующих программ на ЭВМ // Электронно-вычислительная техника в преподавании дисциплин физического цикла: Тезисы докладов Всесоюзной научно - практической конференции. Омск: ОГПИ. 1987. Ч. 2. С. 36-38.
3. Рубцов В., Марголис А., Пажитнов А. Компьютер как средство учебного моделирования // Информатика и образование. 1987. # 5. С. 8-13.
4. Самарский А. А. Нелинейные явления и вычислительный эксперимент // Вестник АН СССР. 1985. # 9. С. 64-77.
5. Досыбаев С. К., Нурмуханов С. Г. Применение ЭВМ при проведении лекционных и практических занятий при изучении темы "Движение в центрально-симметрическом поле" классической механики // Электронно-вычислительная техника в преподавании дисциплин физического цикла: Тезисы докладов Всесоюзной научно-практической конференции. Омск: ОГПИ. 1987. Ч. 2. С. 61-62.
6. Гуськова И. Л. Моделирование электростатических полей // Электронно-вычислительная техника в преподавании дисциплин физического цикла: Тезисы докладов Всесоюзной научно-практической конференции. Омск: ОГПИ. 1987. Ч. 2. С. 64-65.
7. Паскаль И. Ю., Пойзнер Б. Н. Компьютерный эксперимент в лабораторном практикуме вуза: методический аспект // ЭВМ в учебном процессе вуза. Новосибирск: НГУ. 1987. С. 81-84.
8. Sader U., Jodl H. J. Computer-Aided Physics - Particles In Electromagnetic Fields // European Journal of Phy-

sis. 1987. Vol. 8. # 2. P. 88-92.

9. Ильин В.Е., Кочевев А.А. Некоторые аспекты концепции использования ЭВМ в учебном процессе // ЭВМ в учебном процессе вуза. Новосибирск: НГУ. 1987. С. 3-8.

10. Осипов Н.А., Ким П.А. Поливариантноое учебное моделирование движения в центральных полях // ЭВМ в учебном процессе вуза. Новосибирск: НГУ. 1987. С. 20-28.

11. Сергеева Т.А., Зайцев О.С. Моделирование с помощью компьютеров при обучении химии // Химия в школе. 1987. # 2. С. 45-48.

12. Малкова Т.В., Монахов В.М. Математическое моделирование - необходимый компонент современной подготовки школьника // Математика в школе. 1984. # 3. С. 46-49.

13. Буланова Н.Л., Волков Д.В., Ховиев В.Б., Ширков П.Д. Математическое моделирование и перспективы развития школьного образования. М., 1937. 10 с. (Препринт / Институт прикладной математики им. М.В.Келдыша: РЖ "Математика", 1987. 11 Г298).

14. Литинский Г.И. О проблеме осмысленности представления результатов математического моделирования, полученных на ЭВМ // Человек - творчество - компьютер: Тезисы докладов 8 Международного конгресса по логике, философии и методологии науки (Москва. 1987, август). М., 1987. Ч. 1. С. 78-80.

15. Crews Ph. Tbt EXPERT: A Case Study In Integrating Expert System Technology With Computer Assisted Instruction // The 3rd International Conference on Data Engineering (Los Angeles, 1987). Washington, 1987. P. 556-562.

Межвузовский сборник научных трудов  
ЭВМ В ОБРАЗОВАНИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
1988. Рига: ЛГУ им. П. Стучки. С. 128-137

УДК 37.01:007+378.147+517.962.8+519.633+681.3.06

П. Г. Ветхере, У. А. Ветхерс, А. Т. Якович  
ЛГУ им. П. Стучки

ПРОГРАММА ДЛЯ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ  
В ЧАСТНЫХ ПРОИЗВОДНЫХ И СРАВНИТЕЛЬНОГО АНАЛИЗА  
СВОЙСТВ РАЗНОСТНЫХ СХЕМ

В связи с широким внедрением ПЭВМ возрастает роль математического моделирования как метода теоретического исследования физических процессов. Часто математическая модель физического процесса формулируется в виде задачи математической физики, состоящей из уравнения или системы уравнений в частных производных и необходимых для их однозначного решения дополнительных (граничных или начальных) условий.

Для успешного решения поставленных задач необходимо разработать эффективные методы для каждого из классов уравнений в частных производных (эллиптических, параболических, гиперболических и др.). Одним из наиболее распространенных является конечно-разностный подход [1]. В настоящей работе внимание сосредоточено на численном решении методом конечных разностей уравнений параболического типа и уравнений переноса (уравнения гиперболического типа первого порядка) с одной пространственной переменной и произвольными начальными и граничными условиями. Целью работы явилось создание общей программы для сравнительного анализа свойств широко распространенных конечно-разностных схем и численного решения прикладных задач, сводящихся к проблемам данных типов, с максимальным использованием преимуществ, предоставляемых применением ПЭВМ, таких, как визуализация результатов, диалоговый режим работы и т. д.

## 1. Характеристика используемой ПЭВМ

Работа выполнена на персональном компьютере "QL" фирмы "Sinclair" (Великобритания), с процессором "Motorola 68008" (32-битовый, тактовая частота 7.5 МГц, 8-битовая длина данных), 128 Кбайт ОЗУ, хранение информации реализовано с помощью микрокассет емкостью 100 Кбайт [ 2 ]. Компьютер имеет встроенный интерпретатор языка SUPERBASIC, значительно отличающийся от распространенных версий языка MSX BASIC и GW BASIC [ 3 ]. Версия языка имеет богатый набор графических операторов, позволяющих легко осуществить подготовку графических изображений и реализовать визуальное оформление программы. Для распечатки текстовых и графических данных используется матричный принтер "Epson CP-80".

Разработанная программа использует около 60 Кбайт ОЗУ. Этим обусловлено отсутствие возможности эффективной реализации созданной программы на распространенных в настоящее время отечественных ПЭВМ (таких как "Электроника БК-0010", "ДВК 2М" и др.) и выбор более мощной ПЭВМ "Sinclair-QL".

## 2. Постановка решаемых задач

Программа нацелена на решение задач математической физики в следующих постановках:

1. Смешанная задача для параболического уравнения второго порядка в частных производных [ 4 ]:

$$\frac{\partial U}{\partial t} = \frac{\partial}{\partial x} \left( k \frac{\partial U}{\partial x} \right) + gU + f \quad (1)$$

$$x \in [l_1, l_2], t \in [0, T]$$

где  $U = U(x, t)$  - неизвестная функция

$k = k(x) > 0, g = g(x) < 0, f = f(x, t)$  - заданные функции.  
Начальное условие:

$$U(x, 0) = U_0 \quad (2)$$

Граничные условия на концах рассматриваемого отрезка

$$x = l_1, x = l_2$$

$$k(l_1) \frac{\partial u(l_1, t)}{\partial x} + \alpha u(l_1, t) = \mu_1 \quad (3)$$

$$u(l_2, t) = \mu_2$$

где  $u_0 = u_0(x)$ ,  $\mu_1 = \mu_1(t)$ ,  $\mu_2 = \mu_2(t)$  - заданные функции.  
 $\alpha$  - заданный коэффициент.

Ограничения, накладываемые на функции  $k(x)$  и  $g(x)$  не имеют общности поставленной задачи (1-3) [1].

Сформулированная математическая модель описывает различные физические процессы: одномерный нестационарный процесс теплопроводности, диффузию примеси в изотропной среде, диффузию магнитного поля в проводящую среду и т.п. (см. табл. 1).

2. Смешанная задача для гиперболического уравнения первого порядка в частных производных (уравнения переноса) [4]:

$$\frac{\partial u}{\partial t} = \frac{\partial(ku)}{\partial x} + f; \quad x \in [l_1, l_2], \quad t \in [0, T], \quad (4)$$

где  $u = u(x, t)$  - неизвестная функция.

$k = k(x) > 0$ ,  $f = f(x, t)$  - заданные функции.

Начальное условие:

$$u(x, 0) = u_0 \quad (5)$$

граничное условие на одном конце рассматриваемого отрезка

$$u(l_1, t) = \mu \quad (6)$$

где  $u = u_0(x)$ ,  $\mu = \mu(t)$  - заданные функции.

Сформулированная задача (4-6) математически описывает конвективный перенос какой-либо физической величины, характеризующей состояние сплошной среды (внутренней энергии, вихря скорости, концентрации примеси и др.) в одном направлении. При этом  $u$  - характеристика среды,  $k$  - коэффициент конвективного переноса, пропорциональный скорости движения среды,  $f$  - плотность источников или стоков,  $u_0$  - распределение характеристики по отрезку  $[l_1, l_2]$  в начальный момент времени,  $\mu$  - меняющаяся во времени характеристика на входном ( $k > 0$ ) конце отрезка.

Таблица 1. Физическая интерпретация функций, входящих в (1-3)

Физический процесс	Теплопроводность в неподвижной среде	Диффузия нейтральной примеси	Распространение плоского магнитного поля в проводящей среде
Функция			
$u(x, t)$	температура $T$	концентрация примеси $C$	индукция магнитного поля $B$
$k(x)$	коэффициент температуропроводности $\alpha$	коэффициент диффузии $D$	коэффициент диффузии магнитного поля $(\mu/\mu_0\sigma)^{-1}$
$f(x, t)$	линейная плотность источников тепла	линейная плотность источников примеси	-
$g(x)$	коэффициент теплообмена по Ньютону или плотность стоков тепла, пропорциональная температуре	коэффициент пропорциональности распада диффундирующего вещества	-
$k \frac{\partial u}{\partial x} \Big _{l_1} = \mu_1$	заданная плотность теплового потока на концах отрезка	заданная плотность потока примеси через концы отрезка	заданная напряженность электрического поля
$u \Big _{l_i} = \mu_2$ $i=1,2$	заданная температура	заданная температура примеси	заданная индукция магнитного поля

### 3. Конечно-разностные схемы

При решении проблем (1-3) и (4-6) конечно-разностным методом существенно выбрать схему, наиболее адекватно отражающую свойства исходной задачи. При этом возникают проблемы аппроксимации, консервативности, сходимости, устойчивости и асимптотической устойчивости разностных схем [ 1,4 ]. Существование универсальной, наилучшей для всех задач схемы исключается многообразием физических проблем. Так как теоретическое исследование характеристик множества разностных схем трудоемко и зачастую возможно только для линейных задач и при простой форме заданных функций  $K, f, g, u_0, u_1, u_2$  в (1-6), то большое значение при выборе схемы имеют численные исследования. Созданная программа позволяет максимально облегчить решение этой проблемы: в программе реализованы часто используемые схемы, представляющие все основные типы схем для данных задач.

Для задачи (1-3) реализованы:

- явная двухслойная схема;
- явные трехслойные схемы Дюфорта-Франкеля и Саульева;
- неявные двухслойные схемы: полностью неявная схема, схема Кранка-Никольсона, схема повышенной точности, двухслойная схема;
- трехслойная неявная схема Вэйкера-Олифанта;
- некоторые неустойчивые схемы: схема Ричардсона и абсолютно неустойчивая схема.

Для задачи (4-6) выбраны схемы: Лелевье, Прайса, Адамса-Вешфорта, Лакса, Лакса-Вендроффа, Хойна, "чежарда", с центральными разностями, с разностями по потоку, неявная с разностями против потока. Схемы с центральными разностями и разностями по потоку, являются неустойчивыми, а схемы Адамса-Вешфорта и Хойна - слабо неустойчивыми [ 1,4 ].

Включение в программу неустойчивых схем обусловлено использованием их в процессе обучения и преследует методическую цель, слабо неустойчивые схемы успешно могут применяться для решения задач на небольшом временном интервале.

#### 4. Работа с программой

Работу с программой можно разделить на следующие этапы:

1. Формулировка конкретной проблемы типа (1-3) или (4-6).
2. Переход от исходной проблемы к равносному аналогу и выбор параметров схемы.
3. Решение разностной задачи в указанном интервале времени.
4. Обработка и вывод данных.

На первом этапе пользователь по запросу программы должен вводить все необходимые величины и функции, выбрать вид краевых условий.

На втором этапе программа приступает к формированию конечно-разностного аналога. Для этого пользователь должен выбрать одну из предложенных ему разностных схем (рис. 1) (для каждого типа уравнений предлагается 10 схем), после чего на дисплее представляется запись схемы и минимум информации о ее свойствах: порядок аппроксимации, устойчивость и др. (рис. 2). Ознакомившись с этой информацией, пользователь задает параметры сетки (шаг по координате, шаг по времени и полный интервал времени решения задачи (рис. 2)), после чего программа приступит к решению задачи. За ходом решения можно следить визуально по графику на экране дисплея, где отображается решение после заданного числа промежутков времени (рис. 3). Решение задачи можно прекратить, не дожидаясь достижения заданного времени завершения процесса, нажав <ESC>.

После окончания вычислений программа предлагает список возможных дальнейших действий, который содержит все ранее описанные этапы, а также три режима обработки данных. Предусмотрено получение:

- таблицы рассчитанных значений функции  $u(x,t)$  в узловых точках;
- таблицы значений плотности потока  $K \frac{\partial u}{\partial x}$  ;
- графиков этих функций.

Предусмотрена также возможность визуального сравнения четырех решений, полученных разными схемами с одним аналитическим решением, например, точным (рис. 4).

Рис. 1. Меню разностных схем для решения смешанной задачи для дифференциального уравнения параболического типа.

$$dU/dt = LU + f(x, t)$$

$$LU = d/dx (k(x) dU/dx) + g(x)U$$

$$0 < t < T \quad l_1 < x < l_2 \quad g(x) < 0$$
 With start condition:  

$$U(x, 0) = U_0(x)$$
 With boundary conditions:  
 1.  $U = \mu(t)$  - 1.type.  
 2.  $\lambda dU/dx = \mu(t)$  - 2.type.  
 3.  $\lambda dU/dx + \alpha U = \mu(t)$  - 3.type.

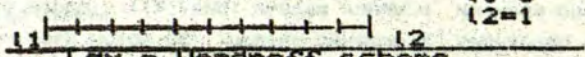
1. Explicit scheme
2. Absolutely implicit scheme
3. Crank - Nicolson scheme
4. Increased precision scheme
5. Absolutely unstable scheme
6. Two-step scheme
7. Richardson scheme
8. Du Fort - Frankel scheme
9. Baker - Oliphant scheme
10. Sauljev scheme

Рис. 2. Пример выбора параметров и описание схемы Лекса-Вендроффа.

Number of intervals ? 10  
 Co-ordinate step  $h = .1$

Time step ? .05  
 Time interval ? 1

$l_1 = 0$   
 $l_2 = 1$



$Lax - Wendroff$  scheme


$$Y_{j+.5}^{n+.5} = .5(Y_j^n + Y_{j+1}^n) + .5\phi_x = .5f_{j+.5}$$

$$Y_{j+.5}^{n+1} = \phi_j^{n+.5} + (\phi_{j+.5}^{n+.5} - \phi_j^{n+.5})/h = f_{j+.5}^{n+.5}$$

$$\phi = kY$$

$$\phi = 0 \text{ (} h^2 + \tau^2 \text{)}$$

Condition of stability  
 $\tau \leq h/k$



В ходе выполнения программы пользователю предоставляется возможность вывести все полученные на экране графики и таблицы на печать.

После обработки данных вновь предложится список возможных действий, который позволяет продолжить работу.

## 5. Характеристика структуры программы

Программа состоит из двух основных блоков подпрограмм для решения задач (1-3) и (4-6) и общего блока подпрограмм сервисного характера. Каждый из блоков можно разбить на функционально схожие группы подпрограмм.

Первый блок содержит:

- 1) группу подпрограмм, обеспечивающих гибкий диалог "пользователь-ЭВМ";
- 2) подпрограммы вычисления заданных функций;
- 3) подпрограммы, выводящие на экран дисплея информацию о конечно-разностных схемах (рис. 2);
- 4) подпрограммы, проводящей вычисление значений неизвестной функции в соответствии с выбранной разностной схемой. В случае неявных схем для задачи (1-3) при этом используется метод факторизации [ 1 ].

Второй из блоков программы содержит:

- 1) подпрограммы инициализации массивов, распределения экранной памяти, логических каналов и т.д.;
- 2) подпрограммы графического отображения решений для последовательности моментов времени (рис. 3);
- 3) подпрограммы обработки результатов, обеспечивающие "вспоминание" предыдущих результатов работы, вычисление значений плотности потока расчетных функций и т.д.;
- 4) подпрограммы, обеспечивающие вывод результатов на матричное печатающее устройство.

## 6. Применение программы

Разработанная программа предназначена для многоцелевого использования и в настоящее время применяется:

- для численного исследования зависимости свойств разно-

Рис. 3. Пример развития моделируемого процесса во времени (отационирования).

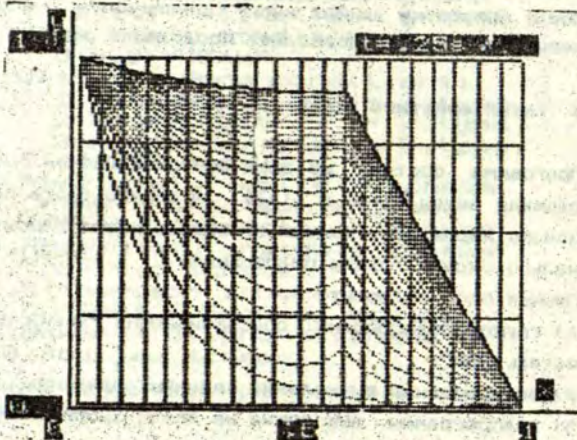
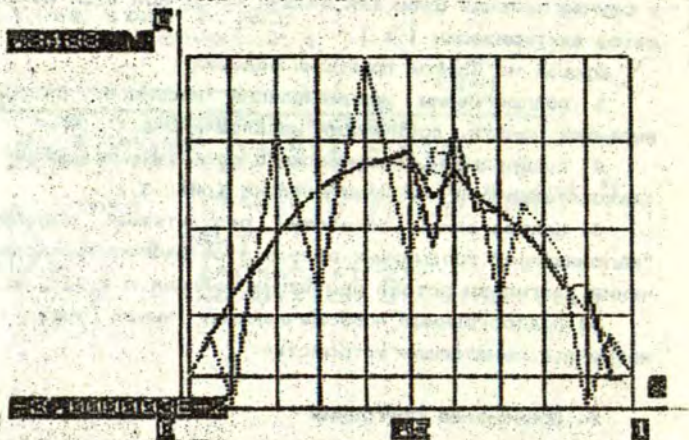


Рис. 4. Пример сопоставления неустойчивых решений разностных схем с точным.



отных схем (аппроксимации, устойчивости, сходимости и др.) от их параметров и вида исходных задач математической физики:

- в спецкурсе "Методы конечных разностей" для студентов специальности # 2016, специализирующихся в области электродинамики и механики сплошных сред, при проведении лабораторных работ по освоению практических навыков использования разностных схем;

- для самостоятельной работы студентов по математическому моделированию в области механики сплошной среды и электродинамики в курсе "Решение прикладных задач на ЭВМ" для специальности # 2016;

- в качестве инструмента в научно-исследовательской работе.

Программа может применяться также для студентов специальности # 0647 "Прикладная математика" при овладении методом математического моделирования и изучении конечно-разностных методов.

В настоящее время с целью более широкого внедрения программы ведется ее переработка на отечественные ПЭВМ ("Искра 1030", "ЕС 1840" и др.), совместимые с IBM PC XT.

#### ВИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Самарский А.А. Теория разностных схем. М.: Наука. 1977. 656 с.
2. Allan B. Sinclair-QL-Begleiter. Heidelberg: Huthig. 1985. 134 S.
3. Александров А.Л., Вашмаков Е.О., Гуткин М.Л., Либеров А.В. О стандарте языка БЕЙСИК // Микропроцессорные средства и системы. 1986. # 5. С. 25-30.
4. Kalis H. Diferencialvienādojumu tuvināšanas risināšanas metodes.- Rīga: Zvaigzne. 1986. 416 lpp.
5. QL User Guide. Sinclair Research Ltd. 1984. 40 pp.

Межвузовский сборник научных трудов  
ЭВМ В ОБРАЗОВАНИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
1988, Рига: ЛГУ им. П. Стучки. С. 138-147

УДК 37.01:007+519.6+681.3.06

М. В. Витиных

Вычислительный центр при ЛГУ им. П. Стучки

**ОБОБЩЕННЫЙ АЛГОРИТМ МЕТОДА ГАУССА  
ДЛЯ ПРИМЕНЕНИЯ В ОБУЧАЮЩЕЙ ПРОГРАММЕ**

В школьном курсе "Основы информатики и вычислительной техники" предполагается ознакомление учащихся с простейшими численными методами, в частности, методом Гаусса для решения систем линейных уравнений. Ценность метода заключается в том, что его реализация достаточно проста и не усложняется при увеличении числа уравнений и переменных.

В статье приводится обобщенный алгоритм и программная реализация решения систем линейных уравнений методом Гаусса. Обобщенный алгоритм обеспечивает эффективность получения результата как для систем, имеющих одно решение, так и для систем, имеющих бесконечное множество решений или совсем не имеющих решения.

Напомним некоторые определения [ 1 ]. Две системы уравнений называются равносильными, если они имеют одно и то же множество решений. Под системой уравнений треугольного вида понимается такая система уравнений, в которой коэффициенты на главной диагонали отличны от нуля, а слева от нее - равны нулю. Сущность метода Гаусса заключается в приведении последовательными целенаправленными преобразованиями исходной системы уравнений к равносильной системе треугольного вида.

**1. Алгоритм на алгоритмическом языке**

Аргументами алгоритма МЕТОД ГАУССА являются: количество уравнений  $R$ ; количество переменных  $K$ ; таблица коэффициентов  $A$  и таблица свободных членов  $B$ . Результатами алгоритма явля-

ются: признак PR, информирующий о наличии или отсутствии решения; решение, представленное таблицей RE. Таблица RE имеет K строк и K+1 столбец. Каждая строка содержит значение переменной, выраженной линейной комбинацией независимых переменных, которым можно придать произвольные значения, и числа. Так решения системы

$$2x + y - 3z = -5$$

$$4x + y - 4z = -9$$

$$2x - 2y + 3z = -2$$

$\{(0.5z - 2; 2z - 1; z) \mid z \in R\}$  таблицей RE будет представлено

0	0	0.5	-2
0	0	2	-1
0	0	1	0.

Алгоритм использует пять вспомогательных алгоритмов. Вспомогательный алгоритм MIN находит из двух чисел наименьшее. Вспомогательные алгоритмы ЗАМЕНА ЭЛЕМЕНТА 1 и ЗАМЕНА ЭЛЕМЕНТА 2 в процессе формирования треугольной системы выполняют замену элемента главной диагонали A(ID, ID) со значением ноль на ненулевой элемент. Вспомогательный алгоритм ВЫЧИТАНИЕ СТРОКИ служит для вычитания ID-го уравнения от всех последующих с целью получить нулевые элементы в ID-м столбце. Вспомогательный алгоритм ОПРЕДЕЛЕНИЕ ЗНАЧЕНИЙ осуществляет обратный код по треугольной системе для последовательного нахождения значений переменных.

В таблице M фиксируется обмен столбцов в таблице A, что соответствует перестановке переменных в системе. Переменная DD содержит наибольший номер ненулевого коэффициента на главной диагонали сформированной треугольной системы.

**АЛГ МЕТОД ГАУССА (цел R, K, вещ таб A[1:R, 1:K],**

**вещ таб B[1:R], лит PR, вещ таб RE[1:K, 1:K+1])**

**вдг R, K, A, B**

**вдв PR, RE**

**нач цел таб M[1:K], цел I, ID, DD**

**для I от 1 до K**

**нц**

**M[I]:=-I**

**кц**

```
ID:=1
ПОКА ID<MIN(R,K)
НЦ
    ЕСЛИ A[ID, ID]=0
        ТО ЗАМЕНА ЭЛЕМЕНТА 1 (R, K, A, B, M, ID)
    ВСЕ
    ЕСЛИ A[ID, ID]≠0
        ТО ЗАМЕНА ЭЛЕМЕНТА 2 (R, K, A, B, M, ID)
    ВСЕ
    ЕСЛИ A[ID, ID]≠0
        ТО ВЫЧИСЛЕНИЕ СТРОКИ (R, K, A, B, ID); ID:=ID+1
        ИНАЧЕ DD:=-ID-1; ID:=-MIN(R,K)+2
    ВСЕ
КЦ
ЕСЛИ ID=MIN(R,K)+1
    ТО DD:=-MIN(R,K)
ВСЕ
PR:="есть"
I:=DD+1
ПОКА I<R
НЦ
    ЕСЛИ B[I]≠0
        ТО PR:="нет"; I:=R+1
        ИНАЧЕ I:=I+1
    ВСЕ
КЦ
ЕСЛИ PR="есть"
    ТО ОПРЕДЕЛЕНИЕ ЗНАЧЕНИЯ (R, K, A, B, M, DD, RE)
ВСЕ
КОН
ВЛ: ЦЕЛ MIN (ЦЕЛ I, J)
НАЧ
    ЕСЛИ I<J
        ТО знач:=-I
        ИНАЧЕ знач:=-J
    ВСЕ
КОН
```

Вспомогательные алгоритмы ОБМЕН СТРОК и ОБМЕН СТОЛБЦОВ в приведенных далее двух алгоритмах служат для обмена уравнений и перестановки переменных в решаемой системе.

АЛГ ЗАМЕНА ЭЛЕМЕНТА 1 (цел R, K, вещ таб A[1:R,1:K],  
вещ таб B[1:R], цел таб M[1:K], цел ID)

ВРГ R, K, A, B, M, ID

ДВВ A, B, M

НАЧ ЦЕЛ IR, IK

IR:=ID+1

ПОКА IR<R

НИ

ЕСЛИ A[IR, ID]≠0

ТО ОБМЕН СТРОК (R, K, A, B, ID, IR); IR:=R+1

ИНАЧЕ IR:=IR+1

ВСЕ

КИ

ЕСЛИ A[ID, ID]=0

ТО

IK:=ID+1

ПОКА IK<K

НИ

ЕСЛИ A[ID, IK]≠0

ТО

ОБМЕН СТОЛБЦОВ (R, K, A, M, ID, IK)

IK:=K+1

ИНАЧЕ IK:=IK+1

ВСЕ

КИ

ВСЕ

КОН

АЛГ ЗАМЕНА ЭЛЕМЕНТА 2 (цел R, K, вещ таб A[1:R,1:K],  
вещ таб B[1:R], цел таб M[1:K], цел ID)

ВРГ R, K, A, B, M, ID

ДВВ A, B, M

НАЧ ЦЕЛ JD, IR, IK

JD:=ID+1

ПОКА JD<MIN(R,K)

НИ

ЕСЛИ A[JD,JD]≠0

ТО

ОБМЕН СТРОК (R, K, A, B, ID, JD)

ОБМЕН СТОЛБЦОВ (R, K, A, M, ID, JD)

ВСЕ

ЕСЛИ A[ID, ID]=0

ТО

IR:=JD+1

ПОКА IR<R

НИ

ЕСЛИ A[IR, JD]≠0

ТО

ОБМЕН СТРОК (R, K, A, B, ID, IR)

ОБМЕН СТОЛБЦОВ (R, K, A, M, ID, JD)

IR:=R+1

ИНАЧЕ IR:=IR+1

ВСЕ

КИ

ВСЕ

ЕСЛИ A[ID, ID]=0

ТО

IK:=JD+1

ПОКА IK<K

НИ

ЕСЛИ A[JD, IK]≠0

ТО

ОБМЕН СТРОК (R, K, A, B, ID, JD)

ОБМЕН СТОЛБЦОВ (R, K, A, M, ID, IK)

IK:=K+1

ИНАЧЕ IK:=IK+1

ВСЕ

КИ

ВСЕ

ЕСЛИ A[ID, ID]=0

ТО JD:=JD+1

ИНАЧЕ JD:=MIN(R,K)+1

ВСЕ

КМ

КОН

АЛГ ОБМЕН СТРОК (цел R, K, вещ таб A[1:R,1:K],

вещ таб B[1:R], цел ID, IR)

ВРС R, K, A, B, ID, IR

ВРВ A, B

НАЧ ЦЕЛ I, ВЕЩ P

ДЛЯ I ОТ 1 ДО K

НМ

P:=A[ID,I]; A[ID,I]:=A[IR,I]; A[IR,I]:=P

КМ

P:=B[ID]; B[ID]:=B[IR]; B[IR]:=P

КОН

АЛГ ОБМЕН СТОЛБЦОВ (цел R, K, вещ таб A[1:R,1:K],

цел таб M[1:K], цел ID, IK)

ВРС R, K, A, M, ID, IK

ВРВ A, M

НАЧ ЦЕЛ I, ВЕЩ P, ЦЕЛ PP

ДЛЯ I ОТ 1 ДО R

НМ

P:=A[I,ID]; A[I,ID]:=A[I,IK]; A[I,IK]:=P

КМ

PP:=M[ID]; M[ID]:=M[IK]; M[IK]:=PP

КОН

АЛГ ВЫЧИТАНИЕ СТРОКИ (цел R, K, вещ таб A[1:R,1:K],

вещ таб B[1:R], цел ID)

ВРС R, K, A, B, ID

ВРВ A, B

НАЧ ЦЕЛ IR, I

IR:=ID+1

ПОКА IR<R

НМ

B[IR]:=B[IR]\*A[ID,ID]-B[ID]\*A[IR,ID]

```
I:=K  
ПОКА I>ID  
НМ  
  A[I,R,I]:=A[I,R,I]*A[ID,ID]-A[ID,I]*A[I,R,ID]; I:=I-1  
КМ  
IR:=IR+1
```

КМ

КОН

ВЛГ ОПРЕДЕЛЕНИЕ ЗНАЧЕНИЙ (ЦЕЛ R, K, ВЕШ ТАБ A[1:R,1:K],  
ВЕШ ТАБ B[1:R], ЦЕЛ ТАБ M[1:K], ЦЕЛ DD,  
ВЕШ ТАБ RE[1:K,1:K+1])

ВРГ R, K, A, B, M, DD

ВРЗ RE

НАЧ ЦЕЛ I, J

I:=DD+1

ПОКА I<K

НМ

ДЛЯ J ОТ 1 ДО K+1

НМ

RE[M[I],J]:=0

КМ

RE[M[I],M[I]]:=1; I:=I+1

КМ

I:=DD

ПОКА I>1

НМ

ДЛЯ J ОТ 1 ДО K

НМ

RE[M[I],J]:=0

LI:=I+1

ПОКА L<K

НМ

RE[M[I],J]:=-RE[M[I],J]-A[I,L]\*RE[M[L],J]; LI:=L+1

КМ

RE[M[I],J]:=-RE[M[I],J]/A[I,I]

КМ

RE[M[I],J]:=-B[I]

L:=I-1

ПОКА L<K

НЦ

RE[M[I],J]:=RE[M[I],J]-A[I,L]\*RE[M[L],J]; L:=L+1

КЦ

RE[M[I],J]:=RE[M[I],J]/A[I,I]; I:=I-1

КЦ

КОН

## 2. Подпрограмма реализации алгоритма на языке MSX BASIC для КУВТ "Yamaha"

В подпрограмме на языке BASIC для переменных и массивов используются такие же имена, как и для соответствующих переменных и таблиц, используемых в записи решения на алгоритмическом языке. Программа, в которой предусмотрен вызов подпрограммы, должна содержать следующие описания:  
DIM A(KX,KX), DIM B(KX), DIM RE(KX,KX+1).

9000 DIM MX(KX)

9002 FOR IX=1 TO KX:MX(IX)=IX:NEXT IX

9004 IF RX>KX THEN MIX=KX ELSE MIX=RX

9006 IDX=1

9008 IF IDX>MIX THEN 9016

9010 IF A(IDX,IDX)=0 THEN GOSUB 9100

9012 IF A(IDX,IDX)<>0

THEN GOSUB 9200:IDX=IDX+1

ELSE DDJ=IDX-1:IDX=MIX+2

9014 GOTO 9008

9016 IF IDX=MIX+1 THEN LDJ=MIX

9018 PR\$="ЕСТЬ"

9020 IX=DDJ+1

9022 IF IX>RX THEN 9028

9024 IF B(IX)<>0 THEN PR\$="НЕТ":IX=RX+1

ELSE IX=IX+1

9026 GOTO 9022

9028 IF PR\$="ЕСТЬ" THEN GOSUB 9300

9030 RETURN

```
9100 IRX=IDX+1
9102 IF IRX>RX THEN 9108
9104 IF A(IRX,IDX)<>0 THEN GOSUB 9150:
      IRX=RX+1 ELSE IRX=IRX+1
9106 GOTO 9102
9108 IF A(IDX,IDX)<>0 THEN RETURN
9110 IKX=IDX+1
9112 IF IKX>KX THEN 9118
9114 IF A(IDX,IKX)<>0 THEN GOSUB 9160:
      IKX=KX+1 ELSE IKX=IKX+1
9116 GOTO 9112
9118 IF A(IDX,IDX)<>0 THEN RETURN
9120 JDX=IDX+1
9122 IF JDX>MX THEN RETURN
9124 IF A(JDX,JDX)<>0 THEN IRX=JDX:
      GOSUB 9150:IKX=JDX:GOSUB 9160:
      RETURN
9126 IRX=JDX+1
9128 IF IRX>RX THEN 9134
9130 IF A(IRX,JDX)<>0 THEN GOSUB 9150:
      IKX=JDX:GOSUB 9160:IRX=RX+1
      ELSE IRX=IRX+1
9132 GOTO 9128
9134 IF A(IDX,IDX)<>0 THEN RETURN
9136 IKX=JDX+1
9138 IF IKX>KX THEN 9144
9140 IF A(JDX,IKX)<>0 THEN IRX=JDX:
      GOSUB 9150:GOSUB 9160:IKX=KX+1
      ELSE IKX=IKX+1
9142 GOTO 9138
9144 IF A(IDX,IDX)<>0 THEN RETURN
9146 JDX=JDX+1:GOTO 9122
9150 FOR IX=1 TO KX:P=A(IDX,IX):
      A(IDX,IX)=A(IRX,IX):A(IRX,IX)=P:
      NEXT IX
9152 P=B(IDX):B(IDX)=B(IRX):B(IRX)=P:
      RETURN
9160 FOR IX=1 TO RX:P=A(IX,IDX):
```

```
A(IX,IDX)=A(IX,KX):A(IX,IKX)=P:
NEXT IX
9162 PX=MX(IDX):MX(IDX)=MX(IKX):
    MX(IKX)=PX:RETURN
9200 IRX=IDX+1
9202 IF IRX>RX THEN RETURN
9204 B(IRX)=B(IRX)*A(IDX,IDX)-
    B(IDX)*A(IRX,IDX)
9206 IX=KX
9208 IF IX<IDX THEN 9212
9210 A(IRX,IX)=A(IRX,IX)*A(IDX,IX)-
    A(IDX,IX)*A(IRX,IDX):IX=IX-1:
    GOTO 9208
9212 IRX=IRX+1:GOTO 9202
9300 IX=DDX+1
9302 IF IX>KX THEN 9306
9304 FOR JX=1 TO KX+1:RE(MX(IX),JX)=0:
    NEXT JX:RE(MX(IX),MX(IX))=1:IX=IX+1:
    GOTO 9302
9306 IX=DDX
9308 IF IX<1 THEN RETURN
9310 FOR JX=1 TO KX:RE(MX(IX),JX)=0:
    GOSUB 9320:RE(MX(IX),JX)=
    RE(MX(IX),JX)/A(IX,IX):NEXT JX
9312 RE(MX(IX),JX)=B(IX):GOSUB 9320:
    RE(MX(IX),JX)=RE(MX(IX),JX)/A(IX,IX)
9314 IX=IX-1:GOTO 9308
9320 LX=IX+1
9322 IF LX>KX THEN RETURN
9324 RE(MX(IX),JX)=RE(MX(IX),JX)-
    A(IX,LX)*RE(MX(LX),JX):LX=LX+1:
    GOTO 9322
```

#### ВИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Дьяконов В.П. Справочник по алгоритмам и программам на языке БЕЙСИК для персональных ЭВМ. М.: Наука. 1987. 240 с.

УДК 37.01:007+371.3+378.147+543/547+681.3.06

А. А. Прикулис, С. Я. Такерис

Вычислительный центр при ЛГУ им. П. Стучки

ИСПОЛЬЗОВАНИЕ ЭВМ

ПРИ ИЗУЧЕНИИ НЕКОТОРЫХ ТЕМ КУРСА ХИМИИ

Вычислительная техника в школах становится все более доступной, поэтому актуальным становится вопрос о ее месте в процессе изучения различных предметов, в т.ч. химии.

Как можно использовать ЭВМ при изучении химии в настоящее время? Для проведения различных расчетов, например: для вычисления молекулярных масс простых и сложных веществ, мольной доли веществ в растворах; для определения процентного состава соединения по его формуле или для определения формулы по процентному составу соединения, для вычисления с использованием химических уравнений. При помощи ЭВМ, в т.ч. программируемых и непрограммируемых калькуляторов эти вычисления проводятся во много раз быстрее, чем вручную. Таким образом, можно увеличить число упражнений и сосредоточиться на их химической сути, а не на арифметических операциях. Программы для таких расчетов весьма просты, и их составление под силу ученикам, имеющим навыки работы с карманными калькуляторами. Для персональных компьютеров такие программы можно составлять на месте (вычисление по формуле и вывод результата можно запрограммировать одной строчкой).

Несколько сложнее представляется использование ЭВМ как технического средства обучения. Например, чтобы демонстрировать на экране телевизора или дисплея какой-либо химический процесс. Реальный эксперимент в химии, конечно, может давать такую информацию о химическом процессе, которую невозможно воспроизвести на компьютере (постепенное изменение окраски, запах, тепловые эффекты). Однако есть эксперименты, которые

нельзя провести в школьной лаборатории из-за отсутствия реактивов, их токсичности, взрывоопасности, большой длительности процесса или отсутствия внешних эффектов. В таком случае полезнее была бы наглядная модель с наблюдаемыми изменениями. В некоторых случаях для учащихся могут оказаться полезными математические модели реальных технологических процессов, например, промышленного производства серной кислоты. Такие модели можно создавать на ЭМ, однако для составления программ необходим определенный опыт, требуется либо переподготовка учителей, либо необходимо заказывать программы у профессионалов.

Возможности компьютеров лучше всего используются при наличии всевозможных обучающих программ. Здесь подразумеваются обучающие тесты, тренажеры, имитаторы. Возможно в будущем для изучения конкретных предметов будут необходимы целые компьютерные курсы. Обучающие программы, а тем более компьютерные курсы должны отвечать определенным эргономическим и методическим требованиям. Поэтому составители программ должны владеть химией и иметь навыки программирования.

Школьный курс химии условно можно разделить на три части. Во-первых, введение и общие вопросы, которые изучаются в седьмом классе. Во-вторых, неорганическая химия с некоторыми элементами аналитической химии, которые изучаются в седьмом и восьмом классах. В-третьих, органическая химия, которая изучается в девятом и десятом классах. Очевидно, обучающие программы должны создаваться по всем этим разделам, а также по ряду вопросов практического характера, неотраженных в школьной программе, но очень важных для практической деятельности человека, учитывая очень широкое распространение и влияние на окружающую среду средств бытовой химии, продуктов химизации сельского хозяйства, медицинских препаратов, фотохимреактивов. Возможно, что использование компьютеров в учебном процессе позволит в будущем эти и другие вопросы включить в школьную программу. Таким образом, при составлении обучающих программ не следует строго ограничиваться официальной программой для средних школ.

При разработке комплектов (пакетов) обучающих программ желательно, чтобы выдерживался единый стиль диалога и, по

возможности, одинаковые способы управления программой. Это, однако, не означает, что все обучающие программы должны создаваться только централизованно, одним или несколькими ведущими коллективами. У учителя должен быть выбор между несколькими программами или пакетами даже по одной теме, различающимися по методике, объему, степени трудности, чтобы он мог предложить разные программы в соответствии с различными психологическими особенностями своих учеников.

Определенное число обучающих программ по химии разработано в Лаборатории проблем школьной и вузовской информатики ВЦ при ЛГУ им. П. Стучки.

Программу по обучению узнаванию химических элементов [ 1 ] можно отнести к обучающим тестам. Она обучает распознавать химические элементы по химическим и физическим свойствам простых веществ. Ученик может задавать вопросы об агрегатном состоянии, электропроводности, плотности вещества, его реакциях с водой, кислотами и т.п. По ответам на вопросы нужно определить химический элемент. Если это не удается, в программе имеется подсказка, в которой указывается на характерное свойство вещества или его применения. Ответ можно вводить в виде химического символа, названия или порядкового номера. Ученик может выбирать уровень трудности материала согласно программе 8-го класса средней школы или первого курса вуза.

Аналогично действуют программы по обучению органическому и неорганическому анализу [ 2,3 ]. Они проверяют как ученик умеет распознавать органические и неорганические вещества по их химическим свойствам. Программа органического анализа после определения класса соединения сообщает процентный состав и молекулярную массу вещества и требует вычислить его суммарную формулу. В анализе неорганических веществ предусмотрена дополнительная возможность "анализа" промежуточных продуктов реакций. Работа с этими программами поможет выбрать наиболее рациональный путь анализа и таким образом лучше подготовиться к проведению реальных лабораторных экспериментов.

Программа по расстановке коэффициентов химических реакций [ 4 ] - типичный пример программы-тренажера. На экране

появляются уравнения химических реакций без коэффициентов, и ученику нужно по возможности быстро расставить правильные коэффициенты. Программа подсчитывает общее время обдумывания, число уравнений, вычисляет среднее время, затраченное на решение одного уравнения. В программе собрано 100 уравнений, сгруппированных по числу веществ в левой и правой частях. Ученик или учитель может выбрать одну или несколько таких групп или вводить свои уравнения (до 25). Если коэффициенты подобраны неправильно и число атомов какого-либо элемента слева и справа не совпадают или же коэффициенты имеют общий делитель, программа сообщает об этом и требует исправить коэффициенты. Следующее уравнение выдается только после того, как правильно решено предыдущее.

Программа по волнометрическому титрованию [ 5 ] представляет пример программы-имитатора. Она воспроизводит процесс титрования, рисует кривую титрования, отображает изменение окраски при достижении точки нейтрализации, создает соответствующие звуковые эффекты (звук падающей капли или струи). Есть возможность выбора различных вариантов титрования: титрование щелочи неизвестной концентрации, титрование кислотой с неизвестной концентрацией или валентностью или же титрование со своими растворами. Процесс титрования в любой момент можно прервать и повторить заново с теми же растворами, провести необходимые расчеты (не выходя из программы). После ввода ответа сообщается величина относительной погрешности и дается возможность выбора нового варианта.

Программа по конструированию молекул и вычислению их параметров [ 6 ] объединяет несколько функций. Программа позволяет конструировать ковалентные молекулы, содержащие до 20 атомов, вычислять межатомные расстояния, валентные и торсионные углы, представлять молекулы на экране как набор сфер, отображающих атомы, или как набор отрезков, отображающих валентные связи, сохранять комплекты данных об атомах в дисковых или кассетных файлах для последующего их использования в этой или других программах, выводить эти данные или другой текст с экрана на печать. Программу может использовать учитель как демонстрационный материал при ознакомлении учеников со строением молекул и ученики или студенты - при самостоя-

тельным изучении материала. Программа может быть полезной также в научной работе при подготовке данных о молекулах.

Такие программы можно применять на практических занятиях, при повторении учебного материала, на кружковых занятиях, при подготовке к олимпиадам или ко вступительным экзаменам. Работа с обучающими программами дает возможность нахождения материала в наиболее подходящем темпе, соответствующем предыдущей подготовке обучаемого или его способностями.

Таких программ пока недостаточно (например, в РФАП Латвии включено 15 программ по химии, из них для каждой конкретной школы можно выбрать не более 6). Причин такого положения несколько. Во-первых, в настоящее время все еще не ясно, какая вычислительная техника будет поступать в школы, какие языки программирования будут применяться. Компьютеры, полностью отвечающие требованиям школы, в нашей стране пока не выпускаются. Их место заполняется различными другими компьютерами (более 10 резко отличающихся моделей), находящимися в школах и на шефских предприятиях. Наиболее массовый бытовой компьютер "Электроника БК-0010", поступающий в школы отдельно или в виде КУВТ-86, снабжался языком ФОКАЛ, теперь выпускается с языком БЕЙСИК, отличающимся от используемых до сих пор версий БЕЙСИКА. Вследствие этого для каждого компьютера приходится создавать свою версию программы. В ближайшем будущем положение, видимо, не улучшится, поскольку новые модели компьютеров, создаваемые у нас или закупаемые за рубежом, сильно отличаются друг от друга. Во-вторых, преподаватели химии в школах или в вузах (хорошо знакомые с методикой преподавания химии) загружены и не могут уделить достаточно внимания разработке обучающих программ. Большинство из них не имеют опыта программирования, или им не доступна необходимая техника. В настоящее время учителя химии в республике вообще не обучены пользованию ЭВМ. В-третьих, в стране пока что не решаются вопросы защиты авторских прав разработчиков программ, вопросы тиражирования, финансирования, материального снабжения. Вследствие этого разработкой программ занимаются энтузиасты, а распространение программ проводится путем личных контактов.

Можно надеяться, что после решения этих проблем компью-

тер сможет стать умным помощником при освоения химии в школе и решении повседневных проблем прикладной химии.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Такерис С.Я., Прикулис А.А., Зекунде А.А., Эглайс М.О. Обучающая программа "ELEMENTS" для освоения свойств химических элементов для КУВТ "Yamaha" (язык программирования - MSX BASIC; язык диалога - русский). Рига: РФАП ЛатвССР, 1986. Рег. # 2075895.00275, инв. # ПП1001. 22 с.
2. Такерис С.Я., Прикулис А.А. Обучающая программа "ORGANAL" для освоения основ органического анализа для КУВТ "Yamaha" (язык программирования - MSX BASIC; язык диалога - русский). Рига: РФАП ЛатвССР, 1986. Рег. # 2075895.00276, инв. # ПП1002. 19 с.
3. Такерис С.Я., Прикулис А.А., Зекунде А.А. Обучающая программа "ANALIZE" для освоения основ неорганического качественного анализа для КУВТ "Yamaha" (язык программирования - MSX BASIC; язык диалога - русский). Рига: РФАП ЛатвССР, 1986. Рег. # 2075895.00277, инв. # ПП1003. 23 с.
4. Такерис С.Я., Прикулис А.А., Алкснис У.Я. Обучающая программа "РЕАКЦИЈА" для освоения химических уравнений для КУВТ "Yamaha" (язык программирования - MSX BASIC; язык диалога - русский). Рига: РФАП ЛатвССР, 1986. Рег. # 2075895.00301, инв. # ПП1004. 24 с.
5. Такерис С.Я., Прикулис А.А. Обучающая программа "TITR" для освоения волюметрического титрования для КУВТ "Yamaha" (язык программирования - MSX BASIC; язык диалога - русский). Рига: РФАП ЛатвССР, 1987. Рег. # 2075895.00334, инв. # ПП1027. 28 с.
6. Такерис С.Я., Прикулис А.А., Вирсис И.Е. Обучающая программа "MOLRED" для конструирования молекул и вычисления их параметров для КУВТ "Yamaha" (язык программирования - MSX BASIC; язык диалога - русский). Рига: РФАП ЛатвССР, 1987. Рег. # 2075895.00425, инв. # ПП1034. 33 с.

Межеузовский сборник научных трудов  
ЭВМ В ОБРАЗОВАНИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
1985, Рига: ЛГУ им. П. Стучки. С. 154-165

УДК 37.01:007+523:52-17+681.3

А.Ф. Раудио

Вычислительный центр при ЛГУ им. П. Стучки

ФОРМИРОВАНИЕ ИЗОБРАЖЕНИЙ В ОБУЧАЮЩИХ ПРОГРАММАХ  
(НА ПРИМЕРЕ КУРСА АСТРОНОМИИ)

МикроЭВМ располагают мощными средствами обработки текстовой и графической информации, интерактивного управления. Значительны возможности методов обучения, например, моделирования [ 1 ]. Одновременно эти средства и методы используются преимущественно в игровых программах, не содержащих воспитательных и познавательных элементов.

Большая часть программ, поступающих в школы в качестве обучающих, разрабатывается не специалистами по педагогике и психологии. В таких программах нет строгого отбора учебного материала и продуманной методики. В свою очередь программы учителей, не знакомых с техникой программирования и возможностями компьютерной графики, являются серыми и скучными. В результате создается множество не пригодных для обучения программ [ 2 ].

Для успешной разработки качественных педагогических программных средств (ППС) необходимо объединить педагогов и специалистов по созданию программного обеспечения в творческие коллективы.

1. Основные требования к изображению в обучающих программах

Вся информация от микроЭВМ ученик получает на экране дисплея. Это означает, что удачная организация вывода информации на экран является необходимым условием создания качественной обучающей программы.

Требования к изображению в обучающих программах следующие:

- следует выделять главное, не загружая экран второстепенной информацией;
- уровень символики и упрощений изображения должен соответствовать уровню подготовленности и восприятия ученика;
- текстовая информация должна иллюстрироваться рисунками.

При организации экрана следует также учитывать требования, выдвигаемые для профессиональных программ [ 3 ].

Опыт работы с прикладными пакетами показывает, что совершенствование изображения (тональная раскраска, изображение теней), детализация (отделка) рисунка целесообразна только до определенной степени, дальнейшее совершенствование не повышает эффективности восприятия информации [ 4 ]. Более того, условность и образность рисунка активизирует процесс мышления ученика.

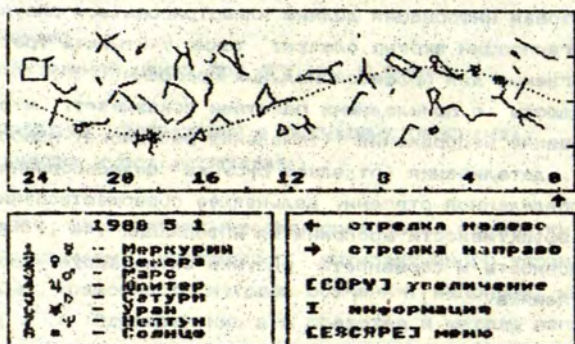
## 2. Организация экрана

Экран должен делиться на функциональные части: текстовое окно, графическое окно, окно вывода дополнительной информации и предлагаемого выбора команд управления (рис. 1.). В программах следует предусмотреть возможность увеличения части сложной схемы или рисунка (например, звездной карты), смещения видимой части изображения в графическом окне [ 5 ]. По выбору можно выделять группу объектов. В пределах одного раздела программы (тематически и функционально независимой части) общая структура экрана должна сохраняться, а изменяться только содержание информации в определенных окнах.

## 3. Использование цвета

Современные микроЭВМ предоставляют широкие возможности использования цвета для окраски фигур и частей экрана. Определение логического цвета позволяет мгновенно изменять окраску фигур и экрана. МикроЭВМ допускают одновременное приме-

Рис. 1. Формирование окон в обучающей программе "Планеты" по курсу школьной астрономии.



нение от 4 до 256 и более цветов и тональностей, что создает для разработчиков обучающих программ проблему отбора цветов.

Рекомендуется работать с простыми цветами, одновременно использовать не более 4 цветов, включая цвет фона. Превышать это число следует только при изображении сложной схемы или рисунка, где меньшее число цветов недостаточно. Необходимо учесть, что в школы в основном поступают черно-белые мониторы. Поэтому цвета должны различаться и по интенсивности. Хорошим набором является - черный, красный, зеленый и белый цвета. Желательно предоставлять пользователю возможность выбора экранных цветов [ 6 ].

В обучающих программах применяются также мигающие цвета (пара цветов, чередующихся 3-4 раза в секунду), как средство выделения детали экрана. Одновременно на экране мигающим цветом рекомендуется "закрашивать" не более одного объекта или одной группы связанных объектов. Для определения мигающей пары цветов желательно выбирать из набора экранных цветов, например, черный и зеленый. Для особого выделения можно использовать цвет, не присутствующий на экране, например, желтый.

#### 4. Вывод текста и средства псевдографики

Текст в обучающих программах может быть следующего характера: инструкция пользователю, познавательный материал, тексты вспомогательной информации и поясняющие подписи.

Инструкция и познавательный текст может не поместиться в поле экрана. В таком случае следует разделить текст на страницы (страница соответствует полю экрана). Текст выдается по страницам, после нажатия клавиши <ПРОБЕЛ>, как правило [7]. Нужно отметить, что читать тексты на экране неудобно. Длина текста не должна превышать 3-4 страницы.

Пояснительные подписи к рисункам выводятся в текстовом окне, подписи к отдельным фрагментам рисунка располагаются либо рядом с рисунком либо на самом рисунке.

Для улучшения восприятия информации часть текста рекомендуется выделять другим цветом. Например, основной текст - белым цветом, выводы - красным, заключение - зеленым. Для

заглавий рекомендуется использовать двойную высоту и ширину символа, для обозначения клавиш - инверсный фон (рис. 2-а).

Тексты обучающих программ, как правило, содержат рисунки или схемы. Их можно выполнить средствами графики высокого разрешения. Поскольку для некоторых микроЭВМ в текстовом режиме это невозможно, рекомендуется применение средств псевдографики: встроенного набора символов (рис. 2-б). К сожалению, встроенные наборы не имеют стандарта, исключая наборы знаков телетекста [ 8 ], применяемых для передачи рисунков по информационным сетям общественного пользования.

## 5. Набор символов

Каждый учебный предмет имеет определенную совокупность знаков, владение которыми является частью требований курса. Это, например, обозначения фигур и их элементов в курсе геометрии, условные знаки в курсе географии. В курсе астрономии условными обозначениями планет и звездных созвездий поясняются явления на небесной сфере. Освоение таких совокупностей знаков достигается их активным использованием и применением для формирования ответов.

Большинство микроЭВМ располагают возможностью переопределения набора символов пользователя, а также назначения символов функциональным клавишам. Закодированное изображение знака, представляемого на экране в матрице 8x8 в виде точек (см. рис. 3), сохраняется в памяти в специальной таблице. Эта таблица может быть дополнена или изменена. Для микроЭВМ создаются генераторы для построения изображений символов и их кодировки. Существуют системы, в которых можно построить символы больших, чем в матрице 8x8, размеров. В обучающих программах рекомендуется использовать одноцветные и не содержащие элементов объемного изображения символы.

Набор символов используется в пояснительных надписях к рисункам, для организации выбора, а также для формирования ответов. Использование совокупности символов в обучающих программах способствует освоению "языка" предмета.

Рис. 2. Вывод текстовой информации и использование псевдографики.

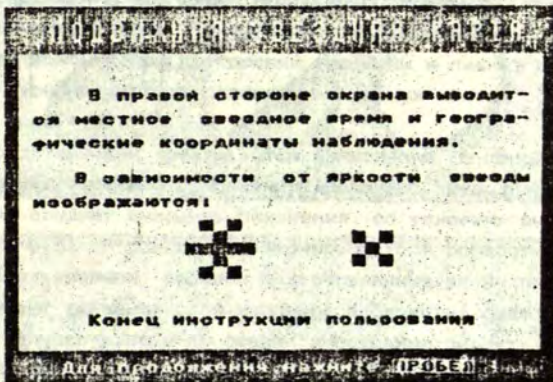
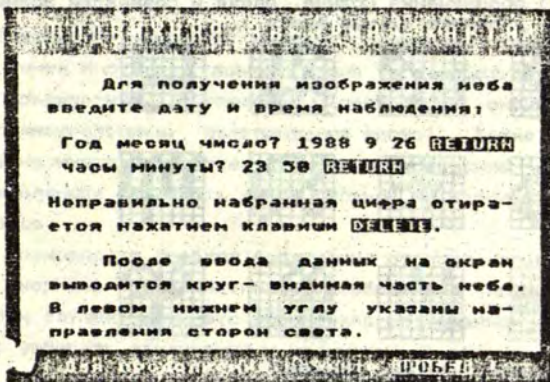


Рис. 3. Символы для обучающих программ по курсу школьной астрономии:

- а) обозначение планет;
- б) обозначение Солнца, звезд и созвездий Зодиака;
- в) вспомогательные знаки.

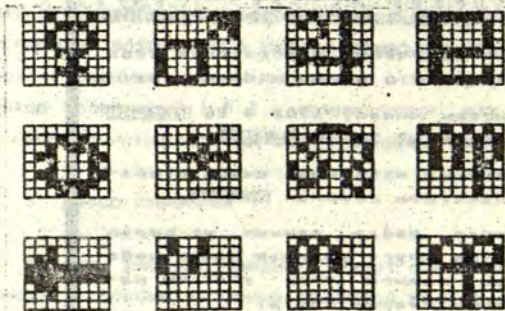
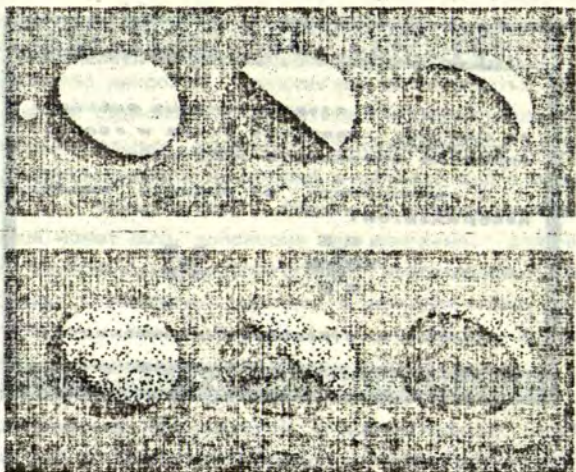


Рис. 4. Изображение фаз планеты:

- а) без учета;
- б) с учетом неравномерности освещения.



## 6. Построение изображений высокого разрешения

Для большинства микроЭВМ имеются операторы отображения на экране точки и отрезка прямой. Кроме того, доступен набор сложных графических операторов (построение окружности, эллипса, прямоугольника, закрашивание фигур). Такие наборы операторов не имеют стандарта. Существуют программы - редакторы, позволяющие создавать любой рисунок с небольшими затратами труда.

Такие возможности требуют соблюдения строгих правил построения рисунков. Перспективные изображения применяются только в том случае, если это обосновано содержанием рисунка. Необходимо избегать стилизаций и излишеств на рисунках.

Модель по которой строится изображение, должна соответствовать уровню восприятия возрастной группы ученика. Например, для программ по курсу природоведения для младших классов при изображении фаз Луны достаточно показать освещенную и затемненную часть (рис. 4-а). Для программ по курсу школьной астрономии при построении фаз Луны и планет следует учитывать неравномерность освещения в разных местах освещенной части (рис. 4-б).

При представлении результатов вычислений с помощью рисунков графики высокого разрешения (например, в моделирующих программах) следует выводить пояснения об условиях вычисления (рис. 5-а,б). В случае достаточно быстрой обработки данных или использования заранее подготовленных рисунков можно создать эффект движения. Для хорошего восприятия быстро протекающего динамического процесса необходимо создать четкую неподвижную часть экрана.

Хорошие результаты достигаются при создании рисунков с комбинированным использованием графики высокого разрешения, псевдографики, наборов символов и соответствующего текста.

## 7. Средства создания изображений

МикроЭВМ "Электроника БК-0010" позволяет строить рисунки с помощью графических средств с указанием направления

Рис. 5. Моделирование шарообразного замкнутого скопления звезд с разным числом звезд.

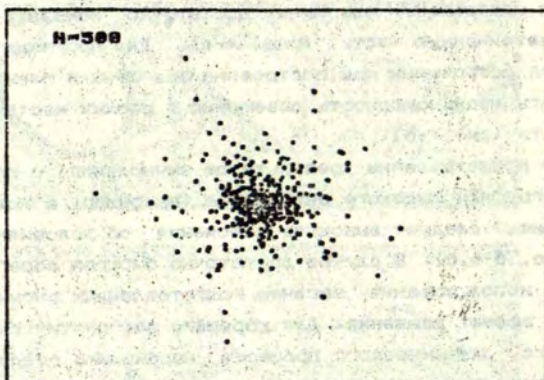
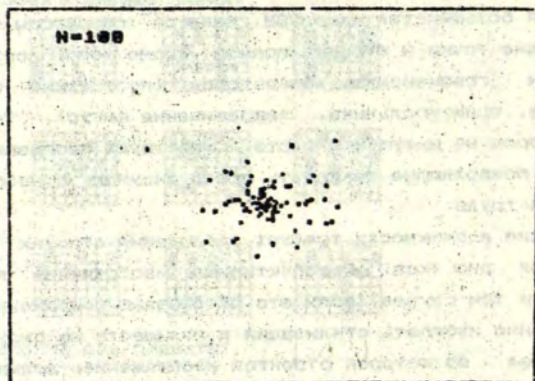
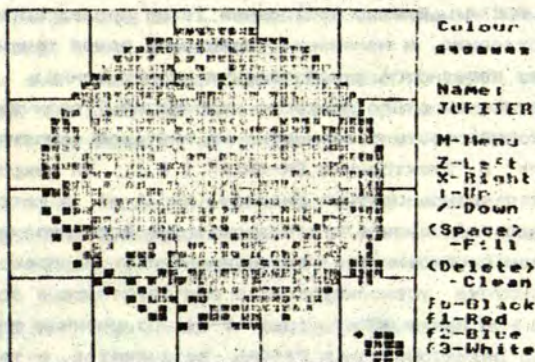


Рис. 6. Построение рисунков с помощью "Графического редактора".  
 а) экранный редактор с изображением Юпитера;  
 б) часть соответствующего фрагмента на языке ФОКАЛ, созданный "Графическим редактором".



89.05 C -J-U-P-I-T-E-R-

89.10 T "␣2↓3→␣13→␣2→␣6→␣14→↓12←␣6←␣5←␣2←␣79←↓8→

89.12 T "␣2←␣3←␣2←␣4←␣6←↓6→␣5→␣7→␣4→␣5→f17→!7←␣2

89.14 T "␣2→␣2→␣3→␣3→␣3→␣2→␣3→␣3→␣7→␣5→↓5←␣4←␣5←␣

89.16 T "␣5→␣6→␣3→␣4→␣3→↓4←␣4←␣2←␣3←␣5←␣2←␣4←

89.18 T "5→␣6→␣4→␣2→␣3→␣4→␣2→↓3←␣3←␣3←␣2←␣3←␣5←␣2←

89.20 T "→␣2→↓2←␣4←␣7←␣2←␣4←␣2←␣4←␣3←␣3←␣3←↓␣2→␣2

89.22 T "␣→␣↓2←␣4←␣2←␣4←␣10←␣5←␣6←␣↓␣2→␣2→␣6→␣2

89.24 T "␣2←␣3←␣5←␣11←␣3←␣↓␣→␣3→␣2→␣4←␣7→␣4→

движения, цвета и режима рисования (заполнить, стереть). Имеется возможность "сжать" информацию, указывая количество посторяющихся команд. Построение графических изображений может быть представлено в виде последовательности операторов TUFFE на языке ФОКАЛ. Однако построение такой последовательности очень трудоемко, а изменение положения одной точки на рисунке требует пересмотра и исправления многих команд.

Для автоматизации этого процесса разработан "Графический редактор", который работает совместно с системой сопряжения микроЭВМ "Acorn" и "Электроника ВК-0010" [ 9 ]. На микроЭВМ "Acorn" строится четырехцветный рисунок (рис. 6-а) в матрице 40x40 точек (или 5x4 символа), с возможностью его сохранения на диске и (при необходимости) редактирования. Изображение законченного рисунка транслируется в соответствующий фрагмент программы на языке ФОКАЛ (рис. 6-б). Полученный фрагмент с помощью редактора языка PASCAL вставляется в текст основной программы. Для многократного использования рисунка к соответствующей группе операторов следует обращаться как к подпрограмме.

## В ы в о д ы

1. Вывод информации и организация изображения на экране, являющихся решающими факторами при создании обучающих программ, подчиняются рассмотренным в статье принципам.

2. При формировании изображения в обучающих программах необходимо придерживаться методики данного предмета, поскольку микроЭВМ лишь дополняет методику преподавания курса.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Разумовский В.Г. Основы ЭВМ в школе // Вечерняя средняя школа. 1985. # 3. С. 51-54.
2. Монахов В.М. Психолого-педагогические проблемы обеспечения компьютерной грамотности учащихся // Вопросы психологии. 1985. # 3. С. 14-22.
3. Brown G.P., Carling R.T., Herot C.F., Kramlich D.A., Souza P. Program Visualization: Graphical Support for Soft-

ware Development // IEEE Computer Graphics & Applications, 1985. # 8. P. 27-35.

4. Фолли Дж.Д. Человеко-машинные интерфейсы // В мире науки. 1987. # 12. С. 59-65.

5. Фолли Дж.Д., ван Дем А. Основы интерактивной машинной графики. М.: Мир, 1985. Т 1. 263 с.

6. Раудис А.Ф. Пакет обучающих программ по курсу школьной астрономии для микроЭВМ "Acorn" (язык программирования - BASIC). Рига: РФАП ЛатвССР, 1987. Рег. # 2075895.00373, инв. # ПП1023. 71 с.

7. Раудис А.Ф. Пакет программ по курсу школьной астрономии для микроЭВМ "Электроника БК-0010" (язык программирования ФОКАЛ). Рига: РФАП ЛатвССР, 1987. Рег. # 2075895.00374; инв. # ПП1024. 82 с.

8. Мартин Дж. Видеотекс и информационное обслуживание общества. М.: Радио и связь, 1987. 182 с.

9. Вульгин Л.Л., Климанс Л.П., Эглайс М.О. Система разработки программного обеспечения для микроЭВМ "Электроника БК-0010" // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 38-50.

Межвузовский сборник научных трудов  
ЗВМ В ОБРАЗОВАНИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
1988, Рига: ЛГУ им. П. Стучки. С. 166-173

УДК 37.01:007+371.3+537.2+681.3.06

К. А. Краст, Е. В. Перфильева

Вычислительный центр при ЛГУ им. П. Стучки

ДЕМОНСТРАЦИОННО-ОБУЧАЮЩИЕ ПРОГРАММЫ ПО ФИЗИКЕ  
ДЛЯ МИКРОЗВМ "ЭЛЕКТРОНИКА ВК-0010"

Использование ЭВМ при изучении курса физики средней школы должно способствовать привитию ученикам навыков работы с компьютером как современным и перспективным средством исследования, дающим возможность нагляднее представить физические процессы. В этой связи широкое применение могут найти графические и диалоговые возможности микроЭВМ.

1. Выбор задач и языка программирования

При отборе задач для разработки демонстрационно-обучающих программ на микроЭВМ "Электроника ВК-0010" (далее "ВК" - наиболее распространенной в школах Латвийской ССР) основной целью было использование ее графических возможностей с организацией динамичного и управляемого изображения при моделировании приборов и экспериментов по физике.

При этом учитывались следующие обстоятельства:

- практическое отсутствие какого-либо прикладного программного обеспечения для микроЭВМ по естественно-научным предметам (в т.ч. по физике):

- ограниченность объема ОЗУ (16 Кбайт) микроЭВМ, необходимого для размещения программы;

- наличие готовых простых, прошедших методическую апробацию сценариев, реализованных в виде демонстрационно-обучающих программ для микроЭВМ "Acorn" с объемом доступным программе ОЗУ, составляющим 12 Кбайт.

С минимальной доработкой были использованы сценарии про-

грамм для микроЭВМ "Асорт" из разделов "Электростатика" и "Геометричная оптика". Программы построены на базе простых математических моделей, с применением динамической графики.

Программы разрабатывались на стандартном для микроЭВМ "БК" языке программирования ФОКАЛ. Несомненно, многие проблемы не стояли бы столь остро при использовании машинного языка вместо языка ФОКАЛ. Однако нельзя не учитывать то, что использование программ предполагалось не только на уроках физики, но и в курсе "Основы информатики и вычислительной техники", на внеклассных занятиях. Программы на языке ФОКАЛ более доступны для разбора учителям и ученикам, при этом в качестве иллюстрации разработанных алгоритмов могут изучаться отдельные фрагменты программ. Ввиду наглядности и доступности текстов программ возможна их переработка для других, составляемых учителями и учениками, обучающих программ. Отдельные группы строк могут включаться как стандартные процедуры, например: организация паузы, гашение служебной строки и курсора, перевод режима экрана с 64 символов в строке на 32 и наоборот и т.д.

## 2. Краткое содержание программ

Содержание разработанных программ следующее.

Программа "Преломление света" [ 1 ] дает возможность проследить за процессом преломления света в различных материалах при многократном изменении величины угла падения. Кроме того, можно получить таблицу, содержащую значения величин углов падения и преломления (рис. 1).

Программа "Линзы" [ 2 ] позволяет рассмотреть ход лучей в собирающих (рис. 2) и рассеивающих (рис. 3) линзах с различным фокусным расстоянием. Здесь также можно получить таблицу со значениями расстояний от оптического центра линзы до объекта и изображения.

Программа "Электроскоп" [ 3 ] дает возможность наблюдать за перемещением зарядов и постепенным отклонением стрелки электроскопа при приближении или отдалении заряженной палочки, в т.ч. при наличии заземления (рис. 4).

Программа "Электризация" демонстрирует перераспределение

Рис. 1-5. Информация, отображаемая на экране в программах:

Рис. 1. "Преломление света";

Рис. 2. "Линзы" - собирающая линза;

Рис. 3. "Линзы" - рассеивающая линза;

Рис. 4. "Электроскоп";

Рис. 5. "Электризация".

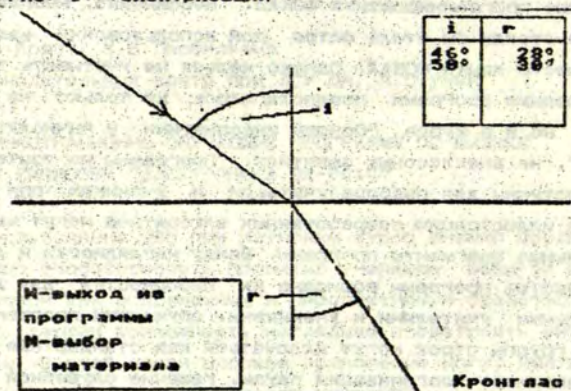


Рис. 1.

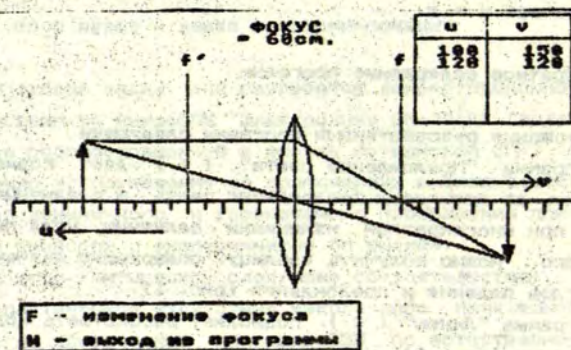


Рис. 2.

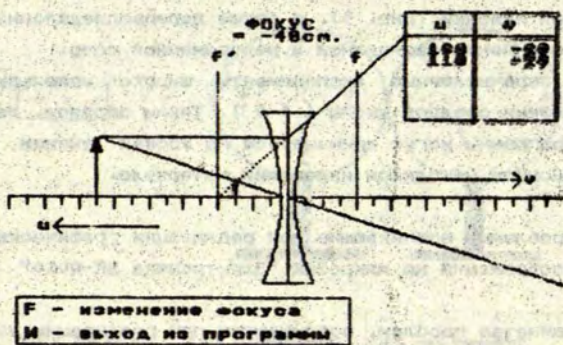


Рис. 3.

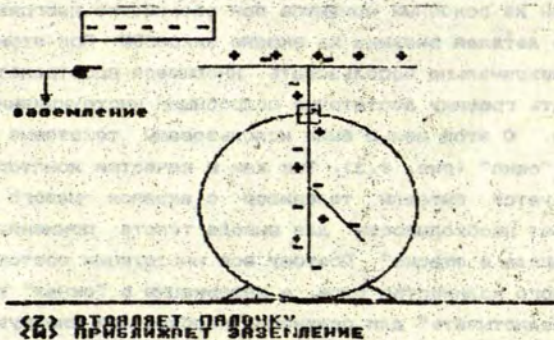


Рис. 4.

зарядов между двумя сферами при электризации под влиянием заряженной палочки (рис. 5), а также перераспределение заряда при соединении заряженной и незаряженной сфер.

Все перечисленные эксперименты широко используются в курсе физики средней школы [ 4-6 ]. Таким образом, разработанные программы могут применяться на уроках физики наряду с традиционной методикой изложения материала.

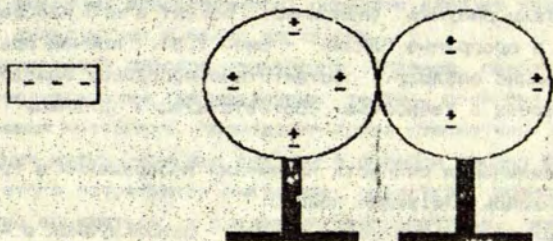
### 3. Проблемы, возникающие при реализации графических изображений на микроЭВМ "Электроника БК-0010"

Большинство проблем, возникающих при реализации управляемых динамических изображений, связаны с ограниченными графическими возможностями микроЭВМ.

Один из основных моментов при разработке программ - размещение деталей рисунка на экране микроЭВМ. При этом необходимо максимально использовать имеющееся пространство и сопровождать графику достаточно подробными инструкциями и пояснениями. С этой целью были использованы текстовые и графические "окна" (рис. 2,3). Так как в качестве монитора к "БК" используется бытовой телевизор с экраном малого размера, возникает необходимость для вывода текста применять режим "32 символа в строке". Поэтому все инструкции состоят из минимального количества слов, а информация в "окнах" необходимо "перелистывать" для получения необходимой инструкции.

При создании графических изображений пришлось столкнуться с проблемой, не описанной в системном руководстве по языку ФОКАЛ: различными размерами графической ячейки вдоль осей  $x$  и  $y$ . Такое несоответствие создает неудобства при рисовании окружностей, используемых для отображения приборов (рис. 1, 2) и нахождения координат отдельных частей рисунка. Это привело к необходимости предварительного создания на миллиметровой бумаге проекта отображения экрана с учетом "цены деления" ячейки вдоль осей  $x$  и  $y$ . Например, в программе "Электроскоп" предварительное проектирование нескольких последовательных изображений экрана с электроскопом, его стрелкой, заряженной палочкой и зарядами (рис. 4).

На "БК" отсутствует оператор, позволяющий достаточно



**<2> НАЛЕВО, <X> НАПРАВО, <8> ОТДЕЛИТЬ**

**Рис. 5.**

быстро перезакрашивать линии в процессе создания динамичных изображений, поэтому приходится последовательно рисовать и гасить детали рисунка (например, объект и его изображение, ход лучей в программе "Линзы" - рис. 2.3), причем при гашении необходимо соблюдать соответствие координат выведенных и гасимых точек с точностью, соответствующей размеру ячейки экрана.

Для увеличения скорости изменения изображений в программах применялись следующие приемы:

- сокращение и упрощение формул, используемых в математической модели, и для вычисления координат точек;

- достаточно большой шаг при перемещении движущихся частей рисунка (например, заряженная палочка, заряды в программе "Электризация" - рис. 5);

- схематичность рисунка (например, схема линзы вместо ее изображения в программе "Линзы" - рис. 2.3);

- при рисовании окружностей используется ее приближение правильным многоугольником (при выводе электроскопа и сфер).

В рассматриваемых программах создавалось управляемое изображение. Управление осуществляется несколькими клавишами, причем не требуется ввода каких-либо символов: для того, чтобы было выполнено перемещение какого-либо элемента рисунка, достаточно нажать указанную в инструкции клавишу. Так можно получить таблицу изменяемых величин в программах "Линзы" и "Преломление света" (рис. 1-3), повторить программу или завершить ее работу. Для предотвращения реакции на несанкционированные нажатия клавиш применяется оператор FCHR(-1) вместо оператора ASK. При этом каждый введенный символ проверяется перед выводом изображения на экран, что увеличивает время от нажатия клавиши до появления отображения символа на экране.

#### 4. Перспективы совершенствования обучающих программ

Используя языки программирования с более развитыми средствами организации вывода графической и текстовой информации, реализованные на более совершенных школьных микроЭВМ, можно увеличить скорость и сделать более плавными переме-

ния изображений, текстовые инструкции - более подробными. Использование цветовых эффектов и определение дополнительных (сконструированных разработчиком) символов сделает изображение менее схематичным и более наглядным.

Возможно также широкое применение обратной связи учащегося с преподавателем для различных тестов, проверки знаний и закрепления материала, проведения сбора статистики, а также генерации индивидуальных заданий с выбором уровня сложности. Для этого потребуются системные средства, позволяющие многократно обращаться к различным подпрограммам, из одной программы вызывать другую (например, в зависимости от полученных знаний).

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Перфильева Е.В. Программа "Преломление света" для микроЭВМ "Электроника БК-0010" (язык программирования - ФОКАЛ; язык диалога - русский). Рига: РФАП ЛатвССР. 1987. Рег. # 2075895.00400. Инв. # ПП1040. 27 с.
2. Перфильева Е.В. Программа "Линзы" для микроЭВМ "Электроника БК-0010" (язык программирования - ФОКАЛ; язык диалога - русский). Рига: РФАП ЛатвССР. 1987. Рег. # 2075895.00423. Инв. # ПП1042. 28 с.
3. Краст К.А. Обучающие программы "Электроскоп" и "Электризация" для микроЭВМ "Электроника БК-0010" (язык программирования - ФОКАЛ; язык диалога - русский). Рига: РФАП ЛатвССР. 1987. Рег. # 2075895.00386. Инв. # ПП1031. 36 с.
4. Демонстрационный эксперимент по физике в средней школе / Под ред. А.А.Покровского. М.: Просвещение. 1978. Ч. I. 351 с.; Ч. II. 287 с.
5. Методика преподавания физики в 8 - 10 классах средней школы / Под ред. В.П.Орехова, А.В.Усовой. М.: Просвещение. 1980. Ч. II. 355 с.
6. Вуховцев В.В., Климонтович Ю.Л., Мякишев Г.Я. Физика. Учебник для 9-го класса средней школы. М.: Просвещение. 1984. 272 с.

Межвузовский сборник научных трудов  
ЭВМ В ОБРАЗОВАНИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
1988. Рига: ЛГУ им. П.Стучки. С. 174-179

УДК 37.01:007+681.3.06

Л. Л. Булыгин

Вычислительный центр при ЛГУ им. П.Стучки

#### ДЕМОНСТРАЦИОННАЯ ПРОГРАММА ДЛЯ СИНТЕЗА ЗВУКА НА ЭВМ

В настоящей работе описаны принципы построения демонстрационной программы для синтеза звука. На экране формируется изображение клавиатуры клавишного инструмента в пределах полутора октав. Имеется возможность изменения октавы от контроктавы до второй октавы, тембра и длины звука.

#### Принципы построения программы

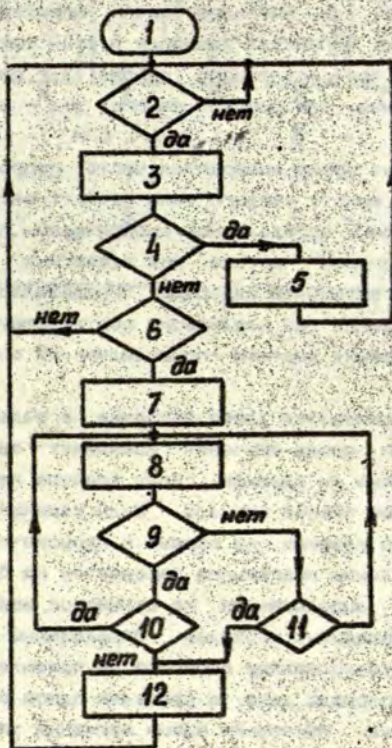
Блоксхема программы представлена на рис. 1. В начале работы программы (блок 1) производятся следующие действия:

1. Запрещаются прерывания от клавиатуры регистра состояния клавиатуры (РСК), 177660<sub>г</sub> - в 6-й бит засылается 1.
2. На экране выдается титульный лист программы, потом рисунок клавиатуры и меню (см. рис. 2).

Далее программа проверяет, поступил ли новый код с клавиатуры (блок 2 на рис. 1). Это проверяется по значению бита 7 РСК. При значении этого бита, равном 1, считывается код с регистра данных клавиатуры (РДК - адрес 177662<sub>г</sub>) - блок 3 на рис. 1. Если считанный код принадлежит клавишам меню, то проводится изменение параметров программы (блок 5). Если нажатая клавиша не принадлежит клавишам меню, проверяется, соответствует ли код клавишам музыкального инструмента (блок 6). При выполнении этого условия соответствующая клавиша на экране закрашивается в цвет, отличный от первоначального (блок 7) и синтезируется один период звука (блок 8). Для вывода звука используется интерфейс кассетного магнитофона. Аппаратная реализация интерфейса позволяет на выходе получить

Рис. 1. Блоксхема программы:

- 1 - инициализация программы;
- 2 - проверка поступления нового кода с клавиатуры;
- 3 - считывание кода с клавиатуры;
- 4 - проверка нажатия клавиши меню;
- 5 - изменение параметров;
- 6 - проверка нажатия клавиши клавиатуры;
- 7 - закраска клавиши;
- 8 - синтез одного периода звука;
- 9 - сравнение длины звучания с заданной;
- 10 - проверка нажатия клавиши;
- 11 - проверка поступления нового кода с клавиатуры;
- 12 - закраска клавиши в первоначальный цвет.



четыре уровня сигнала в зависимости от информации, записанной в 5-м, 6-м битах регистре управления системными внешними устройствами (адрес 177716<sub>g</sub>). Синтезируемый период делится на шесть одинаковых интервалов, что позволяет получить несколько тембров звучания, варьируя соотношение гармоник в сигнале (см. рис. 2). Звук синтезируется согласно параметрам, выбранным из меню. Высота звука определяется длиной одного периода, тембр - наличием и соотношением гармоник, частота которых кратна основной частоте. Период звучания выбирается настройкой на равномерно темперированный лад [ 1 ], частоты нот брались из [ 2 ]. Тон сигнала, форма которого представлена на рис. 3-а, наиболее близко аппроксимирует синусоидальный сигнал, амплитуды гармоник в этом случае наименьшие. Для сигнала, представленного на рис. 3-г сильно выделяется 3-я гармоника, для сигнала на рис. 3-д - вторая гармоника.

Поскольку синтез звука проводится чисто программным путем, то определить высоту каждого тона *a priori* затруднительно. Временные задержки определялись эмпирическим путем - по измеренной частотомером частоте при известных параметрах программы в цикле временной задержки определялась таблица параметров, необходимых для реализации соответствующего звукового ряда. Измерения частоты проводились на сигнале прямоугольной формы.

В блоке 9 сравнивается длина звучания с заданной параметрами меню. Если время звучания превышает заданное, то проверяется, отпущена ли клавиша. Если клавиша не отпущена, синтезируется новый период сигнала. Если клавиша отпущена, проводится закраска клавиши на экране в первоначальный цвет (блок 12), и управление программой передается на блок 2.

Длина звучания определяется по суммарной временной задержке для синтезированных периодов. Определение длины звучания по числу синтезированных периодов не приемлемо, поскольку длина одного периода даже в пределах одной октавы различается в два раза. Поскольку время звучания определяется через каждый период, то максимальная погрешность составляет один период, что даже при наиболее низких синтезируемых нотах составляет десятые доли секунды.

Рис. 2. Отображение клавиатуры и меню на экране.

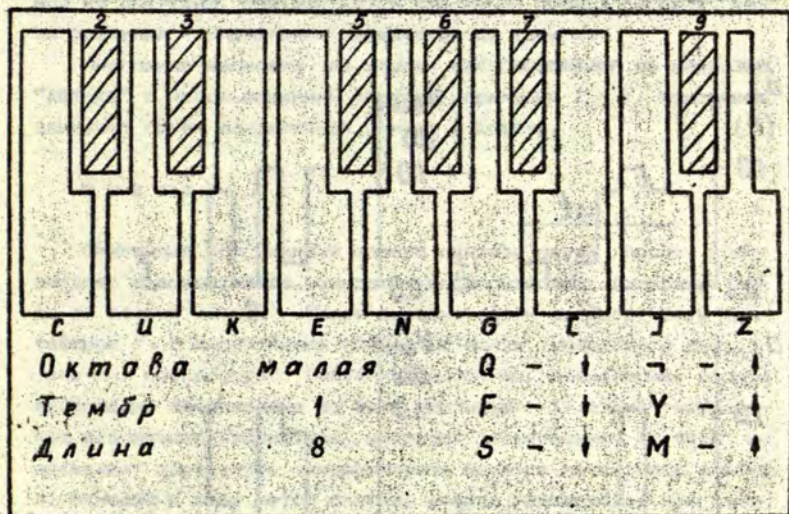
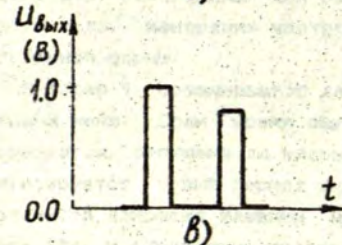
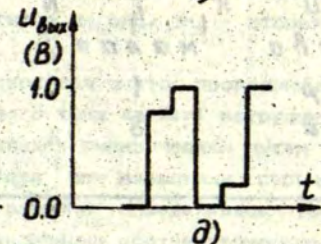
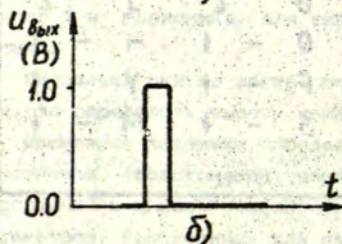
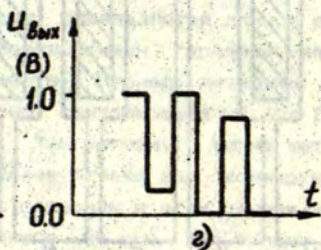
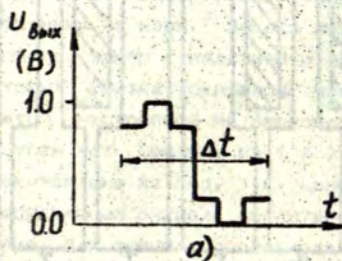


Рис. 3. Синтезирование одного периода звука:  
а) тембр 0; б) тембр 1; в) тембр 2; г) тембр 3; д) тембр 4.



В случае, если длина звучания не превышает заданную параметрами меню, проверяется, не нажата ли другая клавиша, т.е. не поступит ли новый код в РДК (блок 11). Если новый код не поступил, синтезируется следующий период сигнала, если поступил - управление передается на блок 12.

Программа написана на языке МАКРОАССЕМБЛЕР на микроЭВМ "ДВК-2М" с использованием системы пересылки [ 3 ]. Программа занимает объем оперативной памяти 3 Кбайта.

## В ы в о д ы

Описанный программный способ синтеза звука прост и позволяет с достаточной точностью воспроизводить одиночные ноты до второй октавы. Более высокие звуки не могут быть синтезированы с достаточной точностью из-за небольшого быстродействия процессора. Создание программы, позволяющей играть аккордами, не возможно по двум причинам: 1) схемотехническая реализация клавиатуры микроЭВМ "Электроника ВК-0010" не позволяет определять одновременное нажатие нескольких клавиш, 2) описанный выше метод синтеза сильно усложняется при одновременном звучании нескольких нот. Эти ограничения следует иметь ввиду также при создании игровых, обучающих и других программ.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Алексеев В., Мясоедов А. Элементарная теория музыки. М.: Музыка. 1986. 240 с.
2. Пул Л. Работа на персональном компьютере. М.: Мир. 1986. 384 с.
3. Вульгин Л.Л., Климанс Л.П., Эглияс М.О. Инструментальные средства для разработки программного обеспечения для микроЭВМ "Электроника ВК-0010" // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки. 1988. С. 38-50.

## ЗАКЛЮЧЕНИЕ

В сборнике представлены результаты исследований в области применения ЭВМ в образовании. Рассматриваются некоторые научно-методические, практические и организационные аспекты. Ряд статей посвящен вопросам разработки педагогических программных средств для персональных ЭВМ по общеобразовательным предметам "Основы информатики и вычислительной техники", "Физика", "Математика", "Химия", "Астрономия".

Результаты исследований могут быть использованы специалистами по разработке программного и методического обеспечения: преподавателями и студентами при изучении основных и специальных курсов, при разработке курсовых и дипломных работ; работниками институтов усовершенствования учителей и самими учителями при повышении квалификации по информатике и вычислительной технике; а также непосредственно в учебном процессе.

Некоторые разработанные педагогические программные средства для ПЭВМ "Электроника БК-0010", "Агат", "Yamaha" уже внедрены в средних и высших учебных заведениях, в т.ч. педагогических.

СО Д Е Р Ж А Н И Е

ВВЕДЕНИЕ ..... 3

Базовое и инструментальное программное обеспечение

1. Климанс Л.П., Озолина Л.В., Прикулис А.А., Шкярстена И.А. Особенности вывода информации для микроЭВМ "Acorn", "Yamaha", "Корвет", "Агат", "Электроника БК-0010" .....	5
2. Елкина О.М., Курмис А.Г., Озиш И.Я., Павлов С.И., Паякене А.Р., Эглайс М.О. Представление национальных алфавитов на персональных ЭВМ и кодовые таблицы символов .....	25
3. Булыгин Л.Л., Климанс Л.П., Эглайс М.О. Инструментальные средства для разработки программного обеспечения для микроЭВМ "Электроника БК-0010" .....	38
4. Озиш И.Я. Возможности применения математического обеспечения микроЭВМ "ДВК" в учебном процессе .....	51
5. Павлов С.И., Прикулис А.А. О перспективах школьного компьютера .....	56

Программно-методическое обеспечение обучения информатике

6. Андерсоне Л.А., Раудис А.Ф., Фрейвалд Г.В. Учебные телевизионные передачи по курсу "Основы информатики и вычислительной техники" .....	68
7. Каймин В.А., Шаголев А.Г. Опыт обучения информатике и программированию школьников и студентов .....	75
8. Кольман Е.С., Литвинский В.Д., Мельник С.И. Разработка программ на кружковых занятиях школьников на ЭВМ "СМ-4" .....	85
9. Ионин Г.Л., Супе В.В., Янковица М.Я. Разработка и применение обучающих программ для персональных ЭВМ .....	93
10. Павлов С.И., Устинов Н.Н. Разработка и внедрение обучающих программ для школ Шотландии .....	103

Программное обеспечение предметов естественно-научного цикла

11. Павлов С.И., Цилевич В.Л. Возможности применения метода математического моделирования для разработки педагогических программных средств .....	114
12. Бетхере П.Г., Бетхере У.А., Якович А.Т. Программа для решения дифференциальных уравнений в частных производных и сравнительного анализа свойств различных схем .....	128

13. Витиньш М.В. Обобщенный алгоритм метода Гаусса для применения в обучающей программе .....	138
14. Прикулис А.А., Такерис С.Я. Использование ЭВМ при изучении некоторых тем курса химии .....	148
15. Раудис А.Ф. Формирование изображений в обучающих программах (на примере курса астрономии) .....	154
16. Крест К.А., Перфильева Е.В. Демонстрационно-обучающие программы по физике для микроЭВМ "Электроника БК-0010" .....	166
17. Булыгин Л.Л. Демонстрационная программа для синтеза звука на ЭВМ .....	174
<b>ЗАКЛЮЧЕНИЕ</b> .....	<b>180</b>

**Э В М В О Б Р А З О В А Н И И**  
**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

**Сборник научных трудов**  
**(межвузовский)**

**Рецензенты:**

**А.Р. Спектор, зав. отделом**  
**ВЦ при ЛГУ им. П. Стучки,**  
**канд. физ.-мат. наук**  
**Л.П. Новицкий, доцент**  
**РПИ им. А.Я. Пельше,**  
**канд. техн. наук**

**Редакторы: Н. Устиков, Р. Павлова**  
**Корректоры: К. Крест, Е. Перфильева**  
**Технический редактор А. Корхонена**

---

Подписано к печати 22.06.88      ят 09221      Ф/б 60x84/16  
Бумага # 1. 12,0 физ. печ. л. 10,8 усл. печ. л. 8,9 уч.-изд. л.  
Тираж 500 экз.      Зак. # 868      Цена 1 р. 80 коп.

---

Латвийский государственный университет им. П. Стучки  
226098, Рига, б. Райниса, 19  
Отпечатано в типографии, 226050, Рига, ул. Вейденбаума, 5  
Латвийский государственный университет им. П. Стучки

УДК 681.322.068

Климанс Л. П., Озолиня Л. В., Прикулис А. А., Шкерстена И. А.  
ОСОБЕННОСТИ ВЫВОДА ИНФОРМАЦИИ ДЛЯ МИКРОЭВМ "ACORN", "YAMAHA",  
"КОРВЕТ", "АГАТ", "ЭЛЕКТРОНИКА БК-0010" // ЭВМ в образовании.  
Программное обеспечение. Рига: ЛГУ им. П. Стучки, 1988. С.  
5-24.

В статье приводится сравнительное описание возможностей языка программирования БЕЙСИК для микроЭВМ "Acorn", "Yamaha", "Корвет" и "Агат", а также языка ФОКАЛ для микроЭВМ "Электроника БК-0010" для вывода информации на экран дисплея микроЭВМ. Рассматриваются: а) геометрия экрана; б) система отсчета графических и текстовых координат; в) установка режимов экрана; г) определение цвета, очистка экрана; д) формирование элементов графической (точки, линии, прямоугольника, окружности и т. п.) и текстовой информации, определение места вывода. Дается краткая характеристика дополнительных возможностей вывода графической информации (режим ГРАФ для микроЭВМ "Электроника БК-0010", макроязык графики и спрайты для ПЭВМ "Yamaha"). Ил. 3, табл. 9, библиогр. 9 назв.

УДК 681.3.06

Елкина О. М., Курнис А. А., Озиль И. Я., Павлов С. И., Пайкено А. Р., Эглайс М. О. ПРЕДСТАВЛЕНИЕ НАЦИОНАЛЬНЫХ АЛФАВИТОВ НА ПЕРСОНАЛЬНЫХ ЭВМ И КОДОВЫЕ ТАБЛИЦЫ СИМВОЛОВ // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П. Стучки, 1988. С. 25-37.

Рассмотрены некоторые проблемы представления национального (латышского) алфавита для вывода на экран дисплея и принтера на примере ПЭВМ "Acorn". Предложены варианты кодовых таблиц символов, включающие одновременно символы латинского, русского и латышского алфавитов. Предложен вариант привязки символов к клавиатуре. Указано на аналогичные проблемы, возникающие для других национальных алфавитов (литовского, эстонского, украинского, белорусского и др.). Признано целесообразным разработать стандарт на кодовую таблицу символов для одновременного использования в перспективных ПЭВМ и программном обеспечении нескольких альтернативных вариантов переменной части кодовой таблицы. Ил. 4, табл. 9, библиогр. 3 назв.

УДК 681.322

Булыгин Л.Л., Климанс Л.П., Эглыас М.О. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ДЛЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МИКРОЭВМ "ЭЛЕКТРОНИКА БК-0010" // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 38-50.

Рассмотрена возможность использования микроЭВМ "Acorn" и "ДВК-2М" в качестве инструментальных при создании программного обеспечения для микроЭВМ "Электроника БК-0010". Приводятся схемы подключения "БК" и "Acorn", "БК" и "ДВК"; тексты программ интерфейса, позволяющие: а) подготавливать тексты ФОКАЛ-программ с использованием текстового редактора на "Acorn" с дальнейшей пересылкой текста на "БК"; б) пересылать тексты ФОКАЛ-программ из "БК" на "Acorn" для дальнейшего редактирования и распечатки; в) создавать программы на языке МАКРОАССЕМБЛЕРА с использованием редактора текста и транслятора на "ДВК" с последующей пересылкой кода созданной программы в память "БК". Ил. 4, табл. 3, библиогр. 3 назв.

УДК 37.01:007+371.3+681.3.06

Озиш И.Я. ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ МИКРОЭВМ "ДВК" В УЧЕБНОМ ПРОЦЕССЕ // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 51-55.

Рассматриваются некоторые методические вопросы применения в обучении информатике системного и прикладного программного обеспечения микроЭВМ "ДВК" (наиболее распространенной в средних учебных заведениях Латвийской ССР), в т.ч. операционной системы ОС ДВК, интерпретатора языка программирования BASIC, редактора текстов SCREEN. Библиогр. 4 назв.

УДК 681.322.1

Павлов С.И., Прикулис А.А. О ПЕРСПЕКТИВНОМ ШКОЛЬНОМ КОМПЬЮТЕРЕ // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 56-67.

Рассматриваются требования к персональным ЭВМ (ПЭВМ) и комплектам учебной вычислительной техники (КУВТ), предназначенным для решения задач компьютеризации образования, обсуждается необходимый состав и комплектность компьютерного класса, организация локальной сети. Сформулированы некоторые требования к оборудованию рабочих мест учащегося и преподавателя, к возможностям периферийных устройств (дисплея, клавиатуры, принтера), которыми комплектуются ПЭВМ, входящие в КУВТ. Затронут вопрос о необходимом системном и языковом программном обеспечении ПЭВМ. Ил. 1, библиогр. 10 назв.

УДК 37.01:007-74.26+681.3.06

Андерсоне Л.А., Раудис А.Ф., Фрейвалд Р.В. УЧЕБНЫЕ ТЕЛЕВИЗИОННЫЕ ПЕРЕДАЧИ ПО КУРСУ "ОСНОВЫ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ" // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 68-74.

Рассмотрен опыт создания учебных телевизионных (ТВ-) передач по курсу "Основы информатики и вычислительной техники". Целесообразно создавать циклы учебных ТВ-передач, опираясь на результаты работы конкретного научного коллектива. Показаны возможности дополнения информации, поступающей в школы, видеозаписями передач, красочными альбомами и сериями слайдов, а также необходимость систематической передачи звуковых сигналов программ для ПЭВМ, решая таким образом проблему тиражирования. Эффективность учебной передачи увеличивается при своевременной рекламе, распространении аннотаций и методических материалов. Библиогр. 5 назв.

УДК 37.01:007+371.3+378.147

Каймин В.А., Шеголев А.Г. ОПЫТ ОБУЧЕНИЯ ИНФОРМАТИКЕ И ПРОГРАММИРОВАНИЮ ШКОЛЬНИКОВ И СТУДЕНТОВ // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 75-84.

В статье обобщается опыт работы по обучению основам информатики школьников и студентов в Московском институте электронного машиностроения. Определены уровни изучения информатики, сформулированы основные задачи изучения курса, описаны структура и содержание компьютерного практикума. Изложены основные идеи метода построения алгоритмов и программ. Подробно обсуждаются основные методические вопросы, связанные с эффективностью усвоения материала по курсу информатики, даны подходы к их решению в рамках последовательного применения метода систематического построения алгоритмов. Ил. 4, библиогр. 4 назв.

УДК 37.01:007+371.3+519.6+618.3.06

Кельман Е.С., Литвинский В.В., Мельник С.И. РАЗРАБОТКА ПРОГРАММ НА КРУЖКОВЫХ ЗАНЯТИЯХ ШКОЛЬНИКОВ НА ЭВМ СМ-2 // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 85-92.

В работе отображен опыт занятия со школьниками одного из кружков ВЦ при ЛГУ им. П.Стучки. Описывается организация этих занятий. Приведены примеры программ, составленных школьниками 7-10 классов. Ил. 3, библиогр. 6 назв.

УДК 37.01:007+371.3+681.3.06

Ионин Г.Л., Суле В.В., Янкевица М.Я. РАЗРАБОТКА И ПРИМЕНЕНИЕ ОБУЧАЮЩИХ ПРОГРАММ ДЛЯ ПЭВМ // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 93-102.

Рассмотрены характеристики редактора блоксхем RED, системы имитационного моделирования SIM/BASIC и основные принципы, соблюдаемые при разработке этих программных средств. Изложен опыт применения ЭВМ в учебном процессе. Ил. 2, библиогр. 2 назв.

УДК 371+378+681.3.06

Павлов С.И., Устинов Н.Н. РАЗРАБОТКА И ВНЕДРЕНИЕ ОБУЧАЮЩИХ ПРОГРАММ ДЛЯ ШКОЛ ШОТЛАНДИИ // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 103-113.

В статье обобщен опыт организации работ по разработке педагогических программных средств (ППС) и их распространению в школах Шотландии. Обсуждаются факторы, определяющие эффективность применения ЭВМ в обучении. Подробно разбирается процесс разработки ППС, особое внимание уделено организационной стороне дела; рассмотрены вопросы хранения, передачи, финансирования, поддержки и сопровождения разработанных ППС. В заключение рассматриваются проблемы, возникающие при массовом внедрении ППС в процесс обучения. Ил. 3, библиогр. 2 назв.

УДК 37.01:007+371.3+378.147+519.6+681.3.06

Павлов С.И., Цилевич Я.Л. ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ МЕТОДА МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ДЛЯ РАЗРАБОТКИ ПЕДАГОГИЧЕСКИХ ПРОГРАММНЫХ СРЕДСТВ // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 114-127.

Рассмотрены возможности применения, различные аспекты проектирования и реализации педагогических программных средств (ППС) на базе последовательного применения метода математического моделирования. Обсуждены вопросы, связанные с уровнем подготовки пользователя, спецификой предметных областей, различными формами учебных занятия, реализацией дидактических принципов и др. Сформулированы некоторые общие требования, обеспечивающие дидактическую эффективность таких ППС. Библиогр. 15 назв.

УДК 37.01:007+378.147+517.962.8+519.633+681.3.06

Ветхере П.Г., Ветхере У.А., Якович А.Т. ПРОГРАММА ДЛЯ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ И СРАВНИТЕЛЬНОГО АНАЛИЗА СВОЙСТВ РАЗНОСТНЫХ СХЕМ // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 128-137.

В статье описана разработанная авторами учебно-исследовательская программа для ПЭВМ "QL" фирмы "Sinclair", предназначенная для изучения и сравнения свойств различных конечно-разностных схем, используемых для численного решения ряда задач математической физики. Рассматриваются одномерное параболическое уравнение второго порядка и гиперболическое уравнение первого порядка в частных производных. Разработанная программа используется в университетском спецкурсе "Методы конечных разностей", для самостоятельной работы студентов, а также в качестве инструмента в научно-исследовательской работе. Ил. 4, табл. 1, библиогр. 5 назв.

УДК 37.01:007+371.3+519.6+681.3.06

Витиньш М.В. ОБОБЩЕННЫЙ АЛГОРИТМ МЕТОДА ГАУССА ДЛЯ ПРИМЕНЕНИЯ В ОБУЧАЮЩЕЙ ПРОГРАММЕ // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 138-147.

В статье приводится обобщенный алгоритм и программная реализация на языке БЕЙСИК решения систем линейных уравнений методом Гаусса. Обобщение алгоритма заключается в обеспечении машинного решения как для систем, имеющих одно решение, так и для систем, имеющих бесконечное множество решений или совсем не имеющих решения. Библиогр. 1 назв.

УДК 37.01:007+371.3+378.147+543/547+681.3.06

Прикулис А.А., Такерис С.Я. ИСПОЛЬЗОВАНИЕ ЭВМ ПРИ ИЗУЧЕНИИ НЕКОТОРЫХ ТЕМ КУРСА ХИМИИ // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 148-153.

Рассматриваются обучающие программы по темам "Химические элементы", "Уравнения химических реакций", "Неорганический анализ", "Органический анализ", "Волнометрическое титрование", "Конструирование молекул". Обсуждаются также некоторые общие проблемы применения ЭВМ в изучении химии. Библиогр. 6 назв.

УДК 37.01:007+523:52-17-681.3

Рудис А.Ф. ФОРМИРОВАНИЕ ИЗОБРАЖЕНИЙ В ОБУЧАЮЩИХ ПРОГРАММАХ (НА ПРИМЕРЕ КУРСА АСТРОНОМИИ) // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 154-165.

Рассмотрены принципы формирования графических изображений и организации вывода текстовой информации в обучающих программах с учетом возможностей ПЭВМ (в т.ч. школьных). Модель, по которой формируется изображение, должна соответствовать методике предмета и уровню подготовки и восприятия возрастной группы учеников. Приведены примеры изображения из пакетов обучающих программ по курсу школьной астрономии для ПЭВМ "Acorn" и "Электроника БК-0010". Разработан графический редактор, работающий в системе сопряжения ПЭВМ "Acorn" <--> "Электроника БК-0010". Ил. 6, библиогр. 9 назв.

УДК 37.01:007+371.3+537.2+681.3.06

Крест К.А., Порфильев Е.В. ДЕМОНСТРАЦИОННО-ОБУЧАЮЩИЕ ПРОГРАММЫ ПО ФИЗИКЕ для микроЭВМ "Электроника БК-0010" // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 166-173.

Рассматриваются графические возможности микроЭВМ "Электроника БК-0010" для разработки обучающих программ по физике и проблемы, возникающие при организации динамического и управляемого изображения, при размещении рисунка и текста на экране с использованием текстовых и графических "окон", при увеличении скорости выдачи изображения и управления им. Указано на несоответствие цены деления графической ячейки экрана микроЭВМ в направлениях  $x$  и  $y$ . Перечислены некоторые направления разработки перспективных обучающих программ с использованием современных ЭВМ. Ил. 5, библиогр. 6 назв.

УДК 37.01:007+681.3.06

Бульгин Л.Л. ДЕМОНСТРАЦИОННАЯ ПРОГРАММА ДЛЯ СИНТЕЗА ЗВУКА НА ЭВМ // ЭВМ в образовании. Программное обеспечение. Рига: ЛГУ им. П.Стучки, 1988. С. 174-179.

Рассмотрены принципы построения программы для синтеза звука на микроЭВМ "Электроника БК-0010". Программа настроена на темперированный строй и охватывает интервал от "do" контроктавы до "ге" второй октавы. Высота звука рассчитана, исходя из того, что "la" первой октавы соответствует звук с основной частотой 440 Гц. На экране дисплея показано полторы октавы. Нажатие фиксируется окраской соответствующей клавиши в другой цвет. Меню, расположенное в нижней части экрана, позволяет менять октаву, длину и тембр звука. Ил. 3, библиогр. 3 назв.

80194

LU bibliotēka



948009803

527



Цена 1 р. 80 коп.