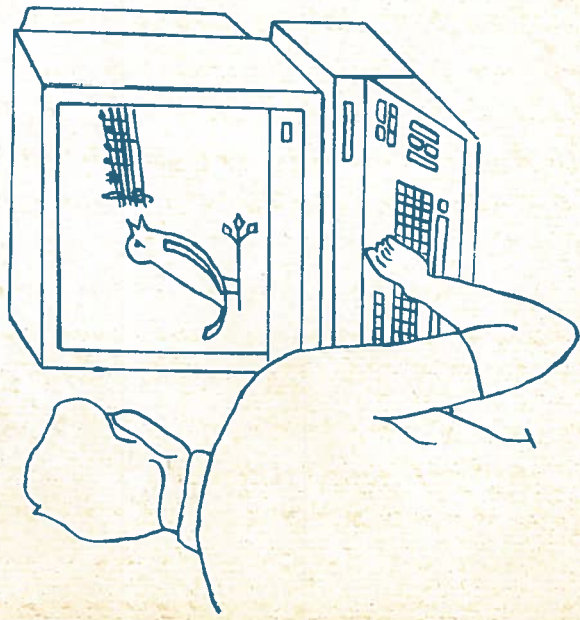


90 коп.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РСФСР
КАРЕЛЬСКИЙ ОРДЕНА "ЗНАК ПОЧЕТА" ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ

ГРАФИЧЕСКИЕ И МУЗЫКАЛЬНЫЕ ВОЗМОЖНОСТИ MSX-БЕЙСИКА



МИНИСТЕРСТВО ОБРАЗОВАНИЯ РСФСР

Карельский орден "Знак Почета"
государственный педагогический институт

ГРАФИЧЕСКИЕ И МУЗЫКАЛЬНЫЕ
ВОЗМОЖНОСТИ MSX-БЕЙСИКА

Учебное пособие
к спецкурсу по информатике

Петрозаводск

1990

ПРЕДИСЛОВИЕ

Одной из замечательных особенностей MSX-Бейсика является наличие превосходной машинной графики и широкого спектра музыкальных возможностей. Используя графические команды, пользователь имеет возможность строить на экране монитор графические фигуры и образы и перемещать их (например, имитировать полет космического корабля). Используя музыкально-ориентированные команды, он может имитировать исполнение музыкальных произведений или программировать музыкальное сопровождение к своим программам.

Излагаемые в пособии приемы программирования графических и звуковых эффектов предполагают знание читателем элементарных сведений о программировании на Бейсике (такие сведения можно найти, например, в [1]). Пособие содержит также раздел, посвященный учебным компьютерным играм по математике и физике, иллюстрирующей полезность применения компьютеров при изучении в школе предметов физико-математического цикла.

Приведенные в пособии программы рассчитаны на реализацию на персональных компьютерах "Ямаха MSX-2".

I. Средства отображения графической информации

MSX-Бейсик содержит графические средства, позволяющие строить на экране монитора простейшие графические элементы (точка, линия, прямоугольник, окружность, эллипс), управлять цветом и яркостью изображения, а также с помощью таких элементов строить различные геометрические объекты и динамические картины. Моделирующие возможности отображаемого объекта.

I.1. Графические экраны

MSX-Бейсик, реализованный на компьютере "Ямаха MSX-2", позволяет работать с двумя типами экранов. Первые два из них являются текстовыми и предназначены для работы с алфавитно-цифровой информацией, остальные - графическими. Выбор экрана осуществляется командой SCREEN (SCREEN - экран) с указанием соответ-

УДК 681.3

Рассмотрено и рекомендовано к печати на заседании кафедры алгебры и информатики

Печатается по решению редакционно-издательского совета Карельского государственного педагогического института

Графические и музыкальные возможности MSX-Бейсика:
 Учебное пособие / В.А.Тарасов; Карел. гос. педагогический институт, Петрозаводск, 1990. - 101 с.
 Табл. 5. Ил. 60. Библ.цитр. 6 назв.

Излагаются основные сведения о графических и звуковых средствах версии языка MSX-Бейсик, реализованной в КВБТ "Ямаха MSX-2".

Предназначено для студентов физико-математических специальностей педагогических институтов и учителей информатики средних школ, ведущих компьютерные кружки.

Рецензенты: В.Т.Вдовиченко, кандидат физико-математических наук, зав. лабораторией ПСА отдела ММАНИП КФ АН СССР;

В.В.Аксенов, научный сотрудник лаборатории микропроцессорных систем отдела ММАНИП КФ АН СССР;

В.А.Токарев, зав. кабинетом информатики Карельского института усовершенствования учителей.

© Карельский государственный педагогический институт, 1990

вущего номера. Так, команда SCREEN 0 и SCREEN 1 определяют текстовые экраны размеров 40x24 и 32x24 символов.

Для работы в графическом режиме предназначены следующие экраны, различающиеся по степени разрешения:

- SCREEN 2 - размер 256x192 точек,
- SCREEN 3 - размер 64x48 блоков,
- SCREEN 4 - размер 256x192 точек,
- SCREEN 5 - размер 256x212 точек,
- SCREEN 6 - размер 512x212 точек,
- SCREEN 7 - размер 512x212 точек,
- SCREEN 8 - размер 256x212 точек.

Графический экран снабжен системой координат, в которой положение точки определяется двумя параметрами - координатами X и Y. Например, в режиме SCREEN 2 координата X может изменяться от 0 до 255, возрастая слева направо, а координата Y - от 0 до 191, возрастая сверху вниз (см. рис. 1.1). Для цветных дисплеев имеется возможность выбора любого из 16 цветов, перечисленных в таблице 1.

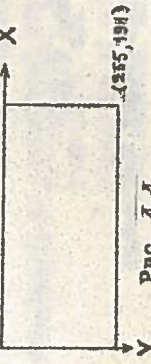


Рис. 1.1

Таблица 1

Номера цветов для цветных дисплеев	
0	прозрачный
1	черный
2	средне-зеленый
3	светло-зеленый
4	темно-синий
5	светло-синий
6	темно-красный
7	голубой
8	средне-красный
9	светло-красный
10	темно-желтый
11	светло-желтый
12	темно-зеленый
13	лиловый
14	серый
15	белый

Выбор цвета осуществляется с помощью оператора COLOR (к^>^e - цвет), имеющего формат COLOR #1, #2, #3,

где #1, #2, #3 - цвета символов, фона и бордюра (некоторые из этих параметров могут быть опущены). Так как компьютер, установленный на рабочем месте пользователя, является монохроматическим, то оператор COLOR вместо цветов передает лишь интенсивности и использовать его мы, как правило, не будем.

1.2. Основные графические операторы

Для установки отдельных точек в графическом режиме используются операторы PSET и PSET. Для рисования прямых линий и многоугольников используется оператор LINE, окружностей, дуг, секторов и эллипсов - оператор CIRCLE. Для построения сложных рисунков используется макрооператор DRAW. Для закраски области, охваченной произвольным контуром, используется оператор RALIT.

При выполнении любых графических команд необходимо задать параметры графической среды, ввода от одного до трех операторов Вейсика - SCREEN, MATH и COLOR. Самым важным в этой группе является оператор SCREEN, создающий формат экрана и указывающий, будет ли изображение цветным.

1.2.1. Оператор PSET (аббревиатура слов POINT SET - указать точку) и PSET дает возможность спроектировать отдельные пиксели (пиксель - минимальный элемент изображения) и имеет следующий синтаксис:

```
PSET (X,Y),C,
PSET(X,Y),C,
```

где X, Y - координаты точки, окрашиваемой в цвет C. Разница между ними заключается в правилах определения параметра C: если в операторе PSET параметр C опущен, то пиксель, заданный своими координатами, приобретает текущий цвет основного изображения, а если параметр C опущен в операторе PSET, то пиксель окрашивается в текущий цвет фона. Если в операторе PSET параметр C указан, то он действует совершенно так же, как и оператор PSET.

К часу ввода, эффективно решаемых с помощью оператора PSET, относятся построение точечных графиков. В качестве иллюстрационного примера приведем программу I, осуществляющую построение на графическом экране графика синусоиды.

Программа 1

```

5 REM СИМУСОИДА
10 SCREEN 2
20 CLS
30 FOR X=5 TO 240
40 PSET (X,90-SIN(X/20))
50 NEXT X
60 GOTO 40

```



Рис.1.2

Классические пояснения к программе 1. Оператор CLS очищает экран (устанавливает экран в режим фона). Оператор GOTO 60 60 прекращает для задержки графического режима (оператор SCREEN 2 сохраняет графический режим на все время выполнения программы, по окончании работы программы при отсутствии 60-й строки экран очищается и переключается в режим SCREEN 0).

Программа 2 рисует на экране мозаику из случайных цветов (для этой цели используется функция RND, генерирующая случайные числа).

Программа 2

```

5 REM МОЗАИКА
10 SCREEN 3
20 FOR I=0 TO 255 STEP 4
30 FOR J=0 TO 191 STEP 4
40 PSET (I,J),RND (1)*16
50 NEXT J
60 NEXT I
70 GOTO 70

```



Рис.1.3

Координаты X, Y в операторе PSET в PSET могут быть не только абсолютными (т.е. заданными непосредственно указанными точки на экране), но и относительными. Относительные координаты задаются с помощью оператора STEP. Так, например, оператор PSET при использовании относительных координат имеет следующий синтаксис:

```
PSET STEP (слит до X, слит до Y),C,
```

где X и Y - координаты последней использованной графической точки. При выполнении этого оператора графический курсор устанавливается в точку с координатами X + слит до X, Y + слит до Y. Иллюстрирующим примером является программа 3.

Программа 3

```

5 REM ПАРАБОЛА
10 SCREEN 5
20 CLS
30 PSET (10,10)
40 FOR X=1 TO 14
50 PSET STEP (2*X, .2*X*X)
60 NEXT X
70 GOTO 70

```

Рис.1.4

1.2.2. Оператор LINE (Line - линия), используемый в формате LINE (X1,Y1)-(X2,Y2),C, строит отрезок, соединяющий точки (X1,Y1), (X2,Y2) цветом C. Указание о цвете может быть опущено. Иллюстрирующим примером является программа 4.

Программа 4

```

5 REM ТРЕУГОЛЬНИК
10 SCREEN 2
20 CLS
30 LINE (50,100)-(150,100)
40 LINE (50,100)-(100,50)
50 LINE (100,50)-(150,100)
60 GOTO 60

```

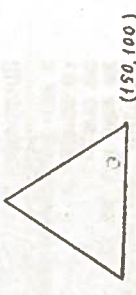


Рис.1.5

Первая пара координат в операторе LINE может быть опущена, в этом случае он принимает вид (указание о цвете также опущено):

```
LINE -(X2,Y2)
```

и строит отрезок, начальной точкой которого является текущая позиция курсора. Такая форма оператора LINE использована в программе 5.

Программа 5

```

5 REM ПРЯМОУГОЛЬНЫЙ ТРЕУГОЛЬНИК
10 SCREEN 2
20 LINE (100,100)-(150,150)
30 LINE (100,150)
40 LINE (100,100)
50 GOTO 50

```

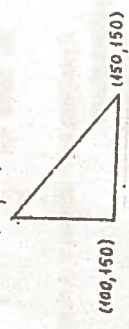


Рис.1.6

Конечная точка в операторе LINE может быть задана приращением относительно начальной точки, в этом случае паре приращений должно предшествовать ключевое слово STEP. При этом оператор LINE (X,Y) - STEP (V,W) эквивалентен оператору LINE (X+V,Y+W). Начальная точка также может быть задана приращением относительно текущего положения курсора. Пример исполь-

зования оператора LINE совместно с оператором STEP приведен в программе 6.

Программа 6

```

5 REM ТРАПЕЦИЯ
10 SCREEN 2
20 LINE (50,100)-STEP(10,-10)
30 LINE -STEP(20,0)
40 LINE -STEP(10,10)
50 LINE -STEP(-10,0)
60 GOTO 60

```



Рис.1.7

Для построения прямоугольника с произвольными вершинами (X1,Y1), (X2,Y2) цветом C оператор LINE используется в формате:

```
LINE (X1,Y1)-(X2,Y2),C,B
```

Параметр C может быть опущен, однако соответствующая вставка в тексте должна указать на его отсутствие.

Программа 7

```

5 REM ВЛАСЕННЫЕ ПРЯМОУГОЛЬНИКИ
10 SCREEN 2
20 CLS
30 FOR K=1 TO 12
40 LINE (KX,5KX)-(1200-10KX,100-5KX),,B
50 NEXT K
60 GOTO 60

```



Рис.1.8

Оператор LINE (X1,Y1)-(X2,Y2),C,BF строит прямоугольник, закрашенный цветом C.

Программа 8

```

10 SCREEN 7
20 CLS
30 LINE (100,100)-(500,500),,B
40 LINE (120,120)-(100,100),,BF
50 GOTO 50

```

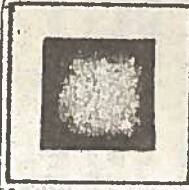


Рис.1.9

Программа 9 строит последовательность уменьшающихся прямоугольников.

Программа 9

```

60 SCREEN 7
70 CLS
80 FOR I=1 TO 10
90 LINE (100-I*I,100-I*I)-STEP(10,10),,B
100 NEXT I
110 GOTO 110

```

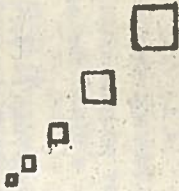


Рис.1.10

1.2.3. Оператор CIRCLE (sac - круг), используемый в формате

```
CIRCLE (X,Y),R,C
```

строит окружность с центром в точке (X,Y) радиуса R цветом C. Программы 10 - 12 иллюстрируют использование оператора CIRCLE. Программа 10

```

5 REM ВЛАСЕННЫЕ ОКРУЖНОСТИ
10 SCREEN 7
20 CLS
30 FOR K=1 TO 15
40 CIRCLE (120,90),SKK
50 NEXT K
60 GOTO 60

```



Рис.1.11

Программа 11

```

5 REM ОРНАМЕНТ
10 SCREEN 2
20 FOR R=5 TO 50 STEP 2
30 CIRCLE (120,90),R
40 NEXT R
50 GOTO 50

```



Программа 12

```

10 SCREEN 2
20 FOR I=1 TO 125
30 CIRCLE (120+70*SIN(I/20),90+70*XSIN(I/20)),32-1/4,I
40 NEXT I
50 GOTO 50

```

Рис.1.12

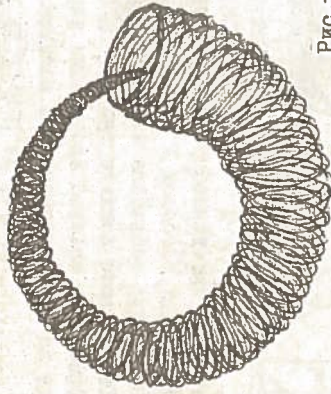


Рис.1.13

Пример использования оператора CIRCLE совместно с оператором STEP приведен в программе 13.

Возможные разновидности оператора CIRCLE приведены в программах I5 - I9.

Программа I5

```

5 REM ВЛОЖЕННЫЕ ЭЛЛИПСЫ
10 SCREEN 7
20 CLS
30 FOR X=1 TO 30
40 CIRCLE (128,95),128-8*X,...,3
50 NEXT X
60 GOTO 60
    
```



Рис. I.17

Программа I6

```

10 SCREEN 2
20 FOR I=40 TO 1 STEP -1
30 Y=90/I : X=3*I
40 CIRCLE (130,30+10*X),X,3,...,3
50 NEXT I
60 GOTO 60
    
```

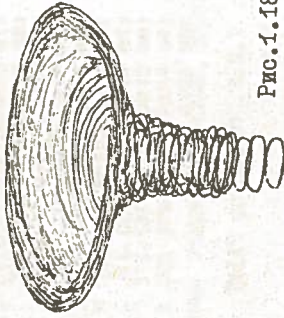


Рис. I.18

Программа I7

```

5 REM CATVPH
10 SCREEN 2
20 P=3.142
30 CIRCLE(120,100),69,3
40 CIRCLE(120,100),70,3
50 C=0: B=.66
60 FOR I=1 TO 30 STEP 3
70 C=(C-0.15): A=C*B
80 B=B*.04
90 CIRCLE(120,93),80+I,3,A,B,.3
100 NEXT I
110 GOTO 110
    
```

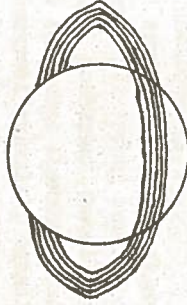


Рис. I.19

Программа I8

```

5 REM ORAL
10 SCREEN 2
20 FOR I=00 TO 24 STEP -4
30 CIRCLE (128,96),96,...,10/I
40 CIRCLE (128,96),96,...,1/10
50 NEXT I
60 GOTO 60
    
```

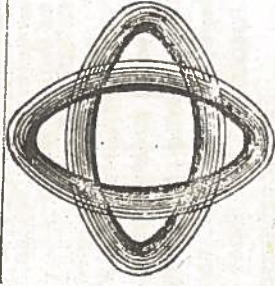


Рис. I.20

Программа I3

```

10 REM СМЕЩЕННЫЕ ОКРУЖНОСТИ
20 SCREEN 7
30 CLS
40 PSET (50,80)
50 FOR K=1 TO 30
60 CIRCLE VTER(S,0),50
70 NEXT K
80 GOTO 80
    
```



Рис. I.14

Если в программе I3 добавить строку

```
55 CIRCLE STEP (5,0),50,0,
```

то нарисованная окружность тут же будет стираться, а затем будет рисоваться следующая окружность. Получаем эффект движущейся окружности.

Оператор CIRCLE, используемый в формате

```
CIRCLE (X,Y),R,C,A1,A2,
```

строит дугу окружности с центром в точке X,Y радиуса R цветом C для значений угла от A1 до A2. По умолчанию A1 = 0, A2 = 2π.

Программа I4

```

5 REM СМЕЩЕННЫЕ ПОЛУОКРУЖНОСТИ
10 SCREEN 2
20 PSET (50,100)
30 FOR K=1 TO 30
40 CIRCLE STEP (5,0),50,.0,3.14
50 NEXT K
60 GOTO 60
    
```



Рис. I.15

Команда

```
CIRCLE (X,Y),R,C,A1,A2,K
```

строит дугу эллипса для значений угла от A1 до A2, получаемого трансформацией окружности радиуса R с коэффициентом сжатия (растяжения) K. Полному эллипсу соответствует A1 = 0, A2 = 2π. Эти значения при выборе оператора CIRCLE могут быть опущены.

Таким образом, эллипс, у которого вертикальная полуось в K раз больше горизонтальной, определяется строкой

```
CIRCLE (X,Y),R,...,K.
```

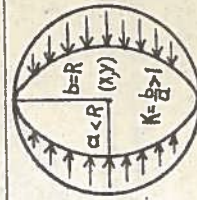


Рис. I.16

Программа 19

```

10 REM ЧЕВУРАШКА
20 SCREEN 5
30 CIRCLE (120, 35), 34
40 CIRCLE (105, 50), 4
50 CIRCLE (135, 50), 4
60 CIRCLE (120, 40), 2
70 CIRCLE (120, 44), 7, 4, 5, 5
80 CIRCLE (85, 30), 30, 9, 5, 1
90 CIRCLE (155, 30), 30, 4, 4, 2, 2
100 CIRCLE (120, 110), 40, 1, 1, 5
110 CIRCLE (145, 155), 20, 1, 1, 5
120 CIRCLE (95, 155), 20, 1, 1, 5
130 CIRCLE (120, 88), 45, 3, 14, 4
140 CIRCLE (85, 90), 7
150 CIRCLE (85, 90), 10, 3, 2, 3
160 CIRCLE (155, 90), 7
170 CIRCLE (155, 90), 10
180 CIRCLE (120, 76), 3
190 PAINT (120, 76)
200 BOTO 200

```

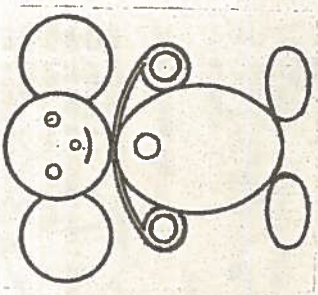


Рис. 1.21

1.2.4. Оператор PAINT (paint - окрашивать), используемый в формате

```
PAINT (X,Y),C
```

окрашивает замкнутую область, начиная от точки (X,Y), в цвет C. При этом цвет границы в цвет, в которую окрашивается область, должен совпадать (иначе цвет раскраски распространится за пределы окрашиваемой поверхности). Программа 20 раскрашивает кольцо, заштрихованное между двумя окружностями, в черный цвет.

Программа 20

```

5 REM КОЛЬЦО
10 SCREEN 7
20 CLS
30 CIRCLE (120, 90), 80
40 CIRCLE (120, 90), 60
50 PAINT (120, 90)
60 BOTO 60

```



Рис. 1.22

1.2.5. Оператор DRAW (draw - рисовать) используется при построении сложных фигур. При этом процедура рисования фигуры предварительно разбивается на отдельные этапы: шаг влево (влево, вверх, вниз), поворот, возврат и т.д.

Оператор DRAW имеет следующий формат:

```
DRAW <строковое выражение >
```

Строковое выражение представляет собой некоторую цепочку команд графического макроязыка (ГМЯ). Каждая команда состоит

из одной буквы, за которой обычно следует аргумент, представляющий собой в большинстве случаев константу, которую можно изменить сенсикой на переменную.

Приведем перечень основных графических команд (точкой отчета для их исполнения является положение графического курсора на экране, установленное перед выполнением оператора ГМЯ).

- Uk - провести отрезок длины k вверх,
- Lk - провести отрезок длины k вниз,
- Lxk - провести отрезок длины k влево,
- Rk - провести отрезок длины k вправо,
- E k - провести отрезок длины k под углом 45°,
- H k - провести отрезок длины k под углом 135°,
- G k - провести отрезок длины k под углом 225°,
- F k - провести отрезок длины k под углом 315°.

Направления движения, определяемые этими командами, изображены на следующей схеме:

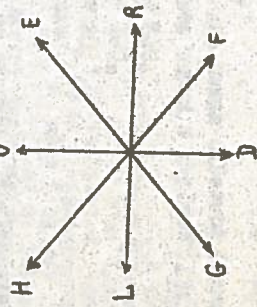


Рис. 1.23

Вышеперечисленные графические команды используются в программе 21.

Программа 21

```

5 REM ДОМ
10 SCREEN 2
20 PSET (140, 112)
30 DRAW "E18R20P12L46R50-46R125R15"
40 DRAW "232R1.5R45040"
50 PSET STEP (-25, 10)
60 DRAW "R502R1.2R125R10R20L1.6R10R20"
70 PSET STEP (0, -45)
80 DRAW "L15L125R14"
90 BOTO 20

```

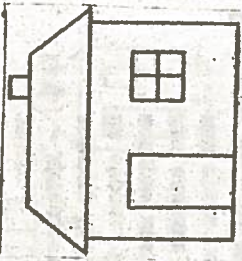


Рис. 1.24

Приведем теперь перечень дополнительных графических команд. Mx, y - перемещение от текущего положения графического курсора до точки с координатами x, y. Если эти координаты

взять со знаками, то координаты точки считаются от текущего положения курсора путем сложения его координат с x и y . Например, если курсор находится в точке (50,50), то в результате выполнения команды DRAW "M-8, -15" он переместится в точку (58,35).

В <команда> - движение без рисования следа. Например, при выполнении команды DRAW "BRK20" пропускается 5 шагов направо и рисуется горизонтальный отрезок длиной 20 пикселей. Заметим, что действие команды DRAW "EMX,Y" равносильно действию команды RESET (X,Y). Это позволяет задавать с помощью оператора DRAW положение графического курсора на экране.

А <команда> - выполнение следующей за префиксом M команды и возврат в исходное положение. Например, команда DRAW "UOM/G5F5" рисует вертикальную стрелку ↑.

Ах - поворот каждого из отрезков, вычерчиваемых следующей стрелкой, на заданный угол против часовой стрелки, где

$$x = \begin{cases} 0, & \text{без поворота,} \\ 1, & \text{поворот на } 90^\circ, \\ 2, & \text{поворот на } 180^\circ, \\ 3, & \text{поворот на } 270^\circ. \end{cases}$$

5х, 1 < х < 255, - изменить все длины в $x/4$ раза. Так, например, S1 означает, что длины отрезков уменьшатся на $1/4$, а S15, - установка цвета ириса отрезков умножатся на 3.

Сх, 0 < х < 15, - установка цвета ириса. Например, DRAW "C206" -рисовать цветом 2 (зеленым) 8 шагов вверх.

Ахф; - интерпретация текста строковой переменной Ахф как команды ГМТ. Точка с запятой является признаком конца строки. Команда X позволяет обратиться к строкам, которые можно объединить и использовать для создания сложных рисунков (строку Ахф можно рассматривать как графическую подпрограмму).

Программа 22 - 25 иллюстрирует действие графических команд.

Программа 22

```

5 REM ЧЕЛОВЕК
10 SCREEN 2
20 CIRCLE (120,80),8
30 DRAW "M120,80
40 DRAW "SMB:BMF15033:R320M720"
50 GOTO 30
    
```



Рис.1.25

Программа 23

```

5 REM СЕКУНДАНАЯ СТЕЛКА
10 SCREEN 2
20 DRAW "M100,100"
30 FOR K=1 TO 20
40 FOR I=0 TO 3
50 DRAW "COMUS9A+I:SIMJ50"
60 NEXT I
70 NEXT K
    
```

Программа 24

```

5 REM ЕЛКА
10 SCREEN 7
20 DRAW "BM250,80C6L3R6L3UD3G3ZF3-
30 FOR I=1 TO 10
40 FOR X=0 TO 3 STEP .3
50 A=250-3X:MSIN(1.57*X)
60 B=80+(10+2X:MSIN(X))X
70 PRESET (A,B),2
80 DRAW "L3R6L3UD3G3ZF3-
90 NEXT X
100 NEXT I
110 GOTO 110
    
```



Рис.1.26

Программа 25

```

5 REM ДОМИКИ
10 SCREEN 2
20 A4="R2PU2BL20D20"
30 B4="E14F14BL20"
40 C4="RSU12L5D12"
50 D4="R4U4L4D4"
60 E4="X4F;B4X4F;BM+8,-8X4F;BM-16,-8X4F;"
70 DRAW "BM20,20C1S2X4F;"
80 DRAW "BM100,100B6C15X4F;"
90 GOTO 90
    
```



Рис.1.27

Задача для самостоятельной работы.

1. Составить программы построения следующих фигур:

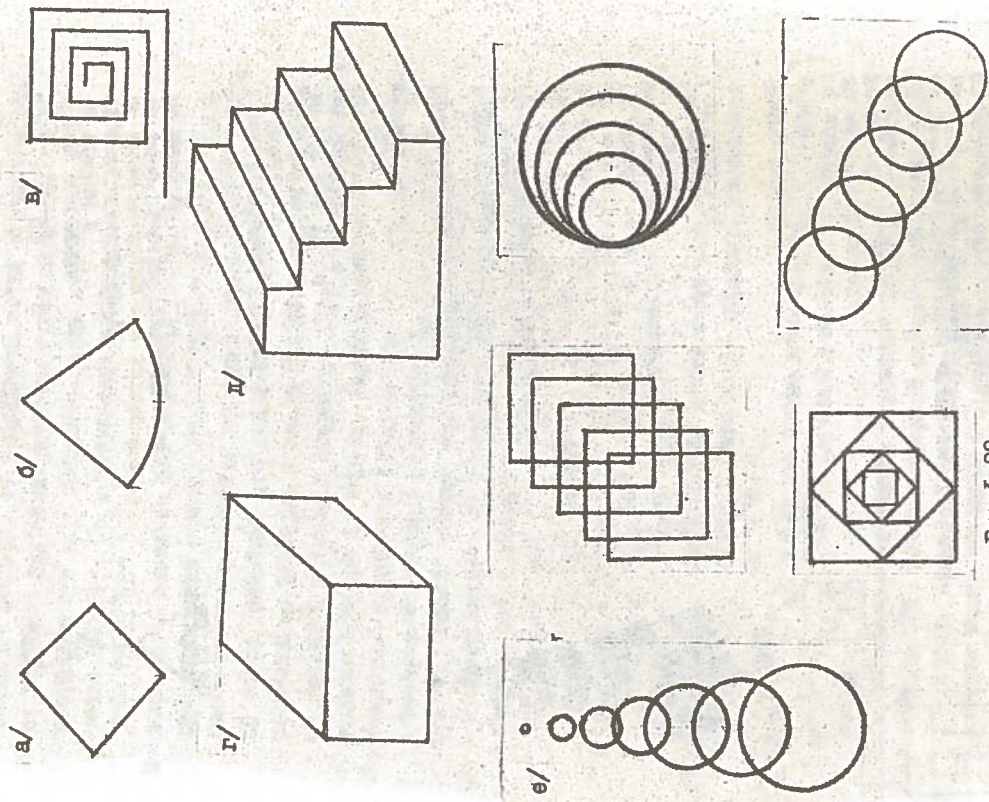


Рис.1.28

2. Неэвклидова программа рисует контур дома, включая оконный проем и трубу.

Программа 26

```

10 SCREEN 7
20 LINE (140, 120) - (500, 200) , B
30 LINE (140, 120) - (220, 50)
40 LINE (220, 50) - (300, 120)
50 LINE (250, 80) - (260, 100) , B
60 LINE (190, 140) - (250, 100) , B
70 GOTO 70

```

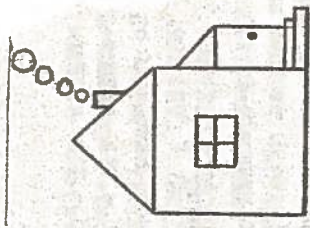


Рис.1.29

Дополнить программу построением других элементов (оконная рама, крыльцо, дверь, дымо из трубы).

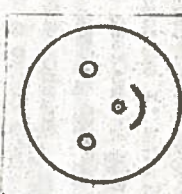
3. Программа 27 рисует смешную рожицу.

Программа 27

```

10 SCREEN 2
20 CIRCLE (128, 96), 80
30 CIRCLE (88, 90), 20
40 CIRCLE (168, 90), 20
50 CIRCLE (128, 120), 15
60 CIRCLE (128, 136), 20, 3, 14, 0
70 GOTO 70

```



Заставьте рожицу попеременно моргать глазами. Рис.1.30

1.3. Спрайт

Английское слово SPRITE (спрайт - спрайт) означает "эльф". Спрайт представляет собой изображение, помещаемое в квадрат определенных размеров. На основе спрайтов конструируются движущиеся объекты. Применять спрайт можно во всех экраных режимах, кроме SCREEN 0.

1.3.1. Объявление размера спрайта

Спрайт могут иметь размеры 8 x 8 и 16 x 16 пикселов (пиксель - минимальный элемент изображения). Размер спрайта указывается во втором операнде оператора SCREEN (первый операнд определяет экраный режим). Например:

```

SCREEN 2, 8 - размер 8 x 8,
SCREEN 2, 2 - размер 16 x 16.

```

SCREEN 2,1 - размер 8 x 8 } с двойным увеличением
 SCREEN 2,3 - размер 16 x 16 } образа при печати.
 По умолчанию спрайт имеет размер 8 x 8 пикселей.

1.3.2. Описание спрайта

Основу описания спрайта составляет его шаблон. Для построения шаблона (образа) спрайта в соответствии с его размером на листе бумаги строится решетка размером 8 x 8 или 16 x 16 квадратов. На этой решетке требуется изображение воспроизводится закрашиванием определенных квадратов решетки. Следующим шагом является кодирование строк шаблона.

Создание спрайта 8 x 8.

Каждой строке шаблона ставится в соответствие двоичное число по следующему правилу: закрашенному квадратику ставится в соответствие 1, незакрашенному - 0. На рисунке 1.31 приведен пример кодирования шаблона 8 x 8.



Рис. 1.31

Заметим, что двоичные коды строк шаблона (всего их восемь) можно преобразовать в десятичные как непосредственным переводом, так и суммированием степеней числа 2, соответствующих закрашенным клеткам шаблона (см. рисунок 1.31).

Каждому шаблону присваивается некоторый номер N , $0 < N < 255$. Фактический шаблон спрайта уславливается, когда коды его строк считываются в строковую псевдопеременную SPRITE\$(N). Переменную SPRITE\$ можно рассматривать как массив, каждый элемент которого содержит некоторую строку шаблона. Поскольку SPRITE\$ имеет строковый тип, числовые данные перед присвоением необходимо преобразовать в строковый формат с помощью

функции CHR\$(CHR - аббревиатура слова Character - символ). Присваивая шаблону, изображенному на рисунке 1.31, нулевой номер, описание этого шаблона можно представить в виде:

SPRITE\$(0) = CHR\$(60) + CHR\$(66) + CHR\$(66) + CHR\$(129) + CHR\$(165) +
 CHR\$(129) + CHR\$(165) + CHR\$(66) + CHR\$(60).

Такое описание можно также осуществить в шкеле с использованием операторов READ и DATA (см. программу 28).

Создание спрайта 16 x 16

На начальной стадии работы строится шаблон 16 x 16, строки которого копируются несколько иначе, чем строки шаблона 8 x 8. С этой целью шаблон 16 x 16 разбивается на четыре блока 8 x 8. Строки каждого из блоков копируются по правилам кодирования шаблона 8 x 8. На рисунке 1.32 приведен пример такой кодировки (десятичные коды строк записаны у каждой строки).

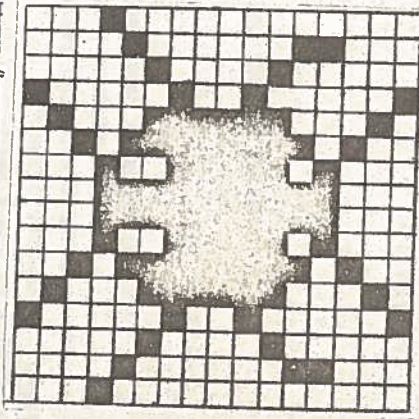


Рис. 1.32

Данные из этих блоков считываются в переменную SPRITE\$ в следующем порядке: верхний левый блок, нижний левый, верхний правый и, наконец, нижний правый.

1.3.3. Воспроизведение спрайта на экране

Спрайт воспроизводится на экране оператором PUT SPRITE, где PUT означает "установить". Оператор имеет следующий формат:

`PUT SPRITE Mnl(X, Y), Mnb, Mшелена,`
 где операнд (X, Y) означает координаты левого верхнего угла квадрата, в который врисован спрайт; операнд M_{nb} - цвет спрайта (он может быть шутен); операнд M_{nl} - номер плоскости, на которой воспроизводится спрайт.

Все экраны, допускающие работу со спрайтами, расслаиваются на 32 воображаемые плоскости, пронумерованные от 0 до 31 (см. рисунок 1.33). На каждой плоскости может находиться не более одного спрайта. Если на плоскости уже находится спрайт, то вывод на нее нового спрайта в любом месте вызывает исчезновение первого. Это позволяет имитировать движение спрайтов, выводя один и тот же спрайт на определенную плоскость и изменяя координаты X, Y . Выводя на одну плоскость в одном месте различные спрайты, можно получить изображение предмета, меняющего форму.

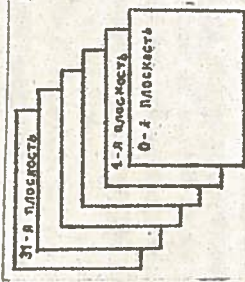


Рис. 1.33

Один и тот же спрайт можно одновременно выводить на несколько плоскостей. При наложении спрайтов полностью выден спрайт, находящийся на плоскости с меньшим номером.

Программа 28 воспроизводит движение спрайта, изображенного на рисунке 1.31.

Программа 28

```
10 SCREEN 2,1
20 DATA 60,66,129,165,129,153,66,60
30 CS=""
40 FOR I=1 TO 8
50 READ A
60 CS=CS+CHR(A)
70 NEXT I
80 SPRITE(0)=CS
90 FOR X=5 TO 240
100 PUT SPRITE 0, (X,96)
110 NEXT X
```

Недостатком этой программы, имитирующей движение спрайта по горизонтальной прямой $Y = 90$, является статичность в том смысле, что, хотя местоположение объекта меняется, его форма

оставалась неизменной.

Для создания полной иллюзии движения необходимо изменить не только местоположение объекта, но и его форму. С этой целью надо, имея некоторый набор спрайтов, заменять их со скоростью 24 кадра в секунду (т.е. с той же скоростью, с какой меняются кадры при демонстрации кинофильмов).

Программа 29 имитирует шагалего человека на основе двух шаблонов, изображенных на рисунках 1.34 и 1.35.

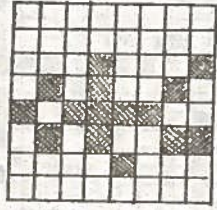


Рис. 1.34

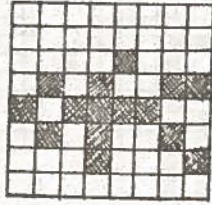


Рис. 1.35

Программа 29

```
3 REM ШАГАЮЩИЙ ЧЕЛОВЕЧЕК
10 SCREEN 2,1
20 FOR I=1 TO 2
30 CS=""
40 FOR J=1 TO 8
50 READ A
60 CS=CS+CHR(A)
70 NEXT J
80 SPRITE(1)=CS
90 NEXT I
100 X=0:H=1:I=1:K=1
110 PUT SPRITE 0, (X,96),:I
120 X=X+H: I=I+1: K=K+1
130 IF I>2 THEN I=1
140 FOR J=1 TO 25
150 NEXT J
160 IF K<240 THEN 140
170 END
180 DATA 16,40,16,60,60,16,40,36
190 DATA 16,40,16,120,20,16,40,72
```

Эта программа имитирует движение человека в горизонтальной полосе $0 \leq Y \leq 240$, $96 \leq X \leq 104$. Частота смены кадров определяется временной задержкой, устанавливаемой строками 140 - 150. После достижения абсциссы $X = 240$ спрайт исчезает с экрана.

Программа 30 имитирует движение человека на основе четырех

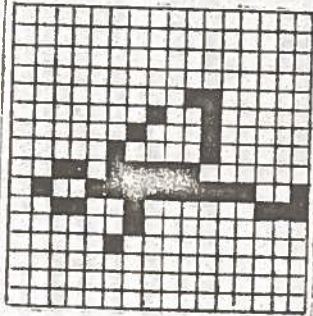


Рис. I.36

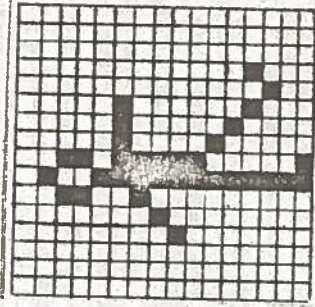


Рис. I.37

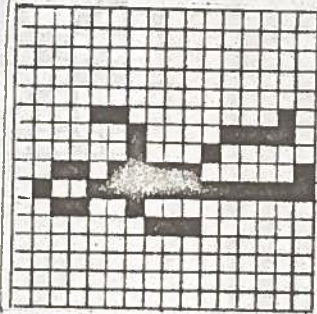


Рис. I.38

Программа 30

```

10 REM MANI
20 SCREEN 2,2
30 FOR I=1 TO 4
40 GOTO 5
50 FOR J=1 TO 32
60 READ A
70 CF=CF+CHR$(A)
80 NEXT J
90 SPRITE$(I)=CF
100 NEXT I
110 X=60
120 M=1
130 I=1
140 PUT SPRITE 0,(X,96),I
150 X=X+H
160 I=I+1
170 IF I>4 THEN I=1
180 FOR J=1 TO 25
190 NEXT J
    
```

```

200 GOTO 140
210 DATA 0,2,5,5,2,19,15,3
220 DATA 3,3,2,2,2,4,4,6
230 DATA 0,0,0,0,0,126,64,52
240 DATA 0,16,240,0,0,0,0,0
250 DATA 0,2,5,5,2,3,7,11
260 DATA 19,3,2,2,2,2,2,3
270 DATA 0,0,0,0,0,224,0,0
280 DATA 0,0,128,64,40,16,0,0
290 DATA 0,2,5,5,2,3,7,11
300 DATA 15,11,2,2,2,2,2,3
310 DATA 0,0,0,0,0,32,192,0
320 DATA 0,0,128,64,64,64,96,0
330 DATA 0,2,5,5,2,3,7,11
340 DATA 19,19,2,2,2,2,2,3
350 DATA 0,0,0,0,0,64,64,192,0
360 DATA 0,0,128,64,128,0,128,0
    
```

Программа 30 зацпает такое движение спрайта, при котором для значения абсциссы $X > 255$ он выходит за правую границу экрана и полностью исчезает, а затем появляется из-за левой границы экрана. Для прекращения процесса движения необходимо выйти из режима работы программы нажатием на клавиши CTRL и STOP.

Программа 31, являющаяся модифицированной программой 30, определяет такое движение человека, при котором он доходит до правой границы экрана, поворачивается и идет к левой его границе, опять поворачивается и т.д. Для имитации такого движения используются восемь спрайтов. При этом движение слева направо определяют спрайты, размещенные в строках 270 - 420, а справа налево - спрайты, размещенные в строках 430 - 580.

Программа 31

```

10 REM MANI
20 SCREEN 2,3
30 FOR I=1 TO 8
40 GOTO 5
50 FOR J=1 TO 32
60 READ A
70 CF=CF+CHR$(A)
80 NEXT J
90 SPRITE$(I)=CF
100 NEXT I
110 X=60 : M=1 : I=1
120 PUT SPRITE 0,(X,96),I
130 X=X+H : I=I+1
140 IF X>196 THEN X=196
150 IF X>4 THEN I=1
160 FOR J=1 TO 25
170 NEXT J
180 GOTO 120
190 X=X-1 : M=-1 : I=5
    
```

```

200 PUT SPRTTE 0, (X, 96), I
210 X=X+H : I=I+1
220 IF X<60 THEN 110
230 IF X=60 THEN 110
240 FOR J=1 TO 25
250 NEXT J
260 GOTO 200
270 DATA 0, 2, 5, 5, 2, 19, 15, 3
280 DATA 3, 3, 2, 2, 2, 4, 4, 6
290 DATA 0, 0, 0, 0, 0, 0, 0, 0
300 DATA 0, 16, 240, 0, 0, 0, 0, 0
310 DATA 0, 2, 5, 5, 2, 5, 7, 11
320 DATA 19, 3, 2, 2, 2, 2, 2, 3
330 DATA 0, 0, 0, 0, 0, 224, 0, 0
340 DATA 0, 0, 128, 64, 40, 16, 0, 0
350 DATA 0, 2, 5, 5, 2, 3, 7, 11
360 DATA 11, 11, 2, 2, 2, 2, 2, 3
370 DATA 0, 0, 0, 0, 0, 0, 0, 0
380 DATA 0, 0, 128, 64, 64, 64, 96, 0
390 DATA 0, 2, 5, 5, 2, 3, 7, 11
400 DATA 19, 19, 2, 2, 2, 3, 2, 3
410 DATA 0, 0, 0, 0, 64, 64, 192, 0
420 DATA 0, 0, 128, 64, 128, 0, 128, 0
430 DATA 0, 0, 0, 0, 0, 0, 0, 0
440 DATA 0, 0, 1, 2, 20, 8, 0, 0
450 DATA 0, 64, 160, 160, 64, 192, 234, 208
460 DATA 200, 192, 64, 64, 64, 64, 64, 192
470 DATA 0, 0, 0, 0, 0, 0, 1, 2, 4
480 DATA 0, 8, 7, 0, 0, 0, 0, 0
490 DATA 0, 64, 160, 160, 64, 192, 240, 200
500 DATA 192, 192, 64, 64, 64, 32, 32, 96
510 DATA 0, 0, 0, 0, 0, 4, 5, 0
520 DATA 0, 0, 1, 2, 3, 2, 1, 0
530 DATA 0, 64, 160, 160, 64, 192, 224, 208
540 DATA 200, 192, 64, 64, 64, 64, 64, 192
550 DATA 0, 0, 0, 0, 0, 2, 3, 0
560 DATA 0, 0, 1, 2, 1, 1, 1, 0
570 DATA 0, 64, 160, 160, 64, 192, 224, 208
580 DATA 200, 192, 64, 64, 64, 192, 64, 192
    
```

Задания для самостоятельной работы. Составить программы движения спрайтов, изображенных на рисунках I.40, I.41, I.42 - I.43.

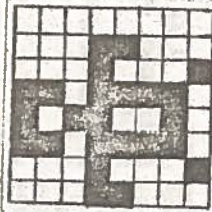


Рис. I.40

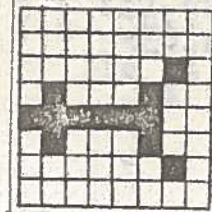


Рис. I.41

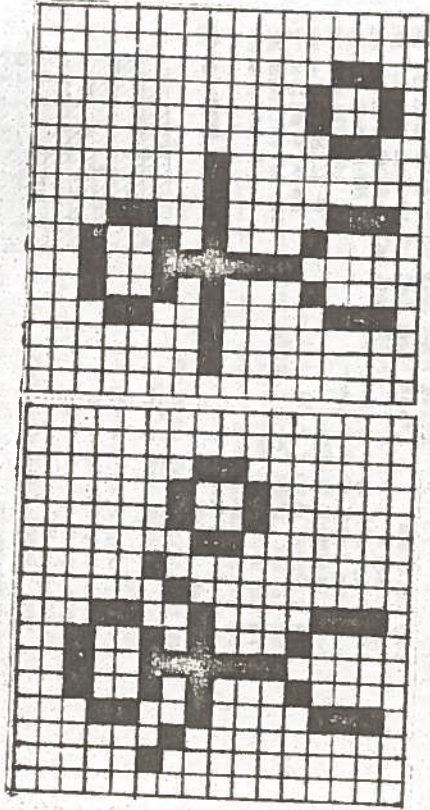


Рис. I.42

Рис. I.43

I.3.4. Управление перемещениями спрайтов

Для реализации интерактивных компьютерных игр необходимо иметь возможность управлять перемещениями спрайтов с помощью курсорных клавиш. Для этой цели используется функция STICK (β) (stick - держерская палочка). Она принимает значения, соответствующие нажатиям курсорных клавишам согласно схеме I.44.

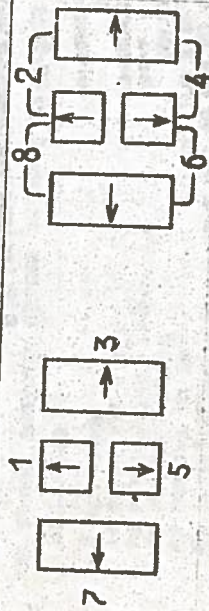


Рис. I.44

Программа 32 управляет перемещениями квадрата, изображенного на рисунке I.45, при помощи курсорных клавиш. Заметим, что числовые коды всех строк шаблона I.45 одинаковы и равны 255₁₆, поэтому описанное шаблон можно получить при помощи функции STRING \$, создающей строку, все символы которой идентичны:

SPRITE\$(0) = STRING\$(8,255)

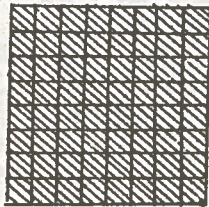


Рис. I.45

Программа 32

```

10 SCREEN 2
20 SPRITE$(0)=STRING$(8,255)
30 X=120 : Y=90
40 PUT SPRITE 0, (X,Y)
50 S=STICK(0)
60 IF S=0 THEN 50
70 IF S=1 OR S=2 OR S=8 THEN Y=Y-3
80 IF S=4 OR S=5 OR S=6 THEN Y=Y+3
90 IF S=2 OR S=3 OR S=4 THEN X=X+3
100 IF S=6 OR S=7 OR S=8 THEN X=X-3
110 GOTO 40
    
```

Программа 32 управляет перемещением лука, изображенного на рисунке I.32.

Программа 33

```

10 REM X,Y
20 SCREEN 2,2
30 DATA 16,8,4,131,69,45,31,15
40 DATA 15,31,47,71,131,4,8,16
50 DATA 8,16,32,48,64,80,96,112,128,144,160,176,192,208,224,240
60 DATA 240,240,240,240,226,193,32,16,8
70 GOTO 11
80 FOR I=1 TO 32
90 READ A
100 G=0:CH$(A)
110 NEXT I
120 SPRITE$(0)=CH$
130 X=120 : Y=90
140 P=STICK(0)
150 PUT SPRITE 0, (X,Y),1
160 IF P=1 OR P=2 OR P=8 THEN Y=Y-3
170 IF P=4 OR P=5 OR P=6 THEN Y=Y+3
180 IF P=2 OR P=3 OR P=4 THEN X=X+3
190 IF P=6 OR P=7 OR P=8 THEN X=X-3
200 GOTO 140
210 END
    
```

Программа 34 управляет перемещением по экрану курсора, оставляющим за собой след.

Программа 34

```

5 REM УПРАВЛЕНИЕ КУРСОРОМ
10 DIM X(10),Y(10)
20 SCREEN 3
30 FOR I=1 TO 10
40 Y(I)=90 : X(I)=120+I*3
50 NEXT I
60 PSET (120,90)
70 IF STICK(0)=0 THEN 70
80 S=STICK(0)
90 IF S=1 THEN U=0 : V=-3
100 IF S=3 THEN U=3 : V=0
110 IF S=5 THEN U=0 : V=3
120 IF S=7 THEN U=-3 : V=0
130 X(0)=X(1)+U : Y(0)=Y(1)+V
140 PSET(X(0),Y(0))
150 FOR I=10 TO 1 STEP -1
160 X(I)=X(I-1) : Y(I)=Y(I-1)
170 NEXT I
180 GOTO 80
    
```

Программа 35 реализует простую компьютерную игру "Питон", в основе которой лежит управление курсором. Содержание игры следующее: на экране разбросаны квадратки желтого цвета, изображающие кроликов, в центре экрана находится квадратик белого цвета - питон. После нажатия на одну из курсорных клавиш питон начинает двигаться в заданном направлении. Нажатием на другие курсорные клавиши направление движения можно изменить. Задача игрока состоит в том, чтобы съесть всех кроликов. Питон, съев кролика, удлиняется на один квадратик. При этом скорость его движения уменьшается. Компьютер выигрывает, если игрок просрочит время, выпустит питона за рамку или питон самопересекается.

Программа 35

```

5 REM ПИТОН
10 DIM X(31),Y(31)
20 P=0:TIME=0:TIME=0:M=0
30 SCREEN 3:COLOR,,14
40 FOR I=1 TO 30
50 X1=INT(256*RND(1))
60 Y1=INT(192*RND(1))
70 C=POINT(X1,Y1)
80 IF C<>10 THEN M=M+1
90 PSET(X1,Y1),10
100 NEXT I
110 C=POINT(127,95)
120 IF C<>10 THEN M=M+1
130 PSET(127,95),15
140 X(1)=127:Y(1)=95:L=1
    
```

```

150 DY=0: DY=0
160 S=8TICK(0)
170 IF S=1 THEN DX=0: DY=-4
180 IF S=3 THEN DX=4: DY=0
190 IF S=5 THEN DX=0: DY=4
200 IF S=7 THEN DX=-4: DY=0
210 A=X(1)+DX: B=Y(1)+DY
220 R=POINT(A, B)
250 IF R=15 AND S<>0 GOTO 370: REM"САМОПЕРЕСЕЧЕНИЕ ПИТОМА"
240 PSET(X(L), Y(L)), 4
250 FOR I=L TO 2 STEP -1
260 X(I)=X(I-1): Y(I)=Y(I-1)
270 NEXT I
280 X(1)=A: Y(1)=B
290 IF R=10 THEN L=L+1: R=4: PSET(A, B), 15: A=X(1)+DX: B=Y(1)+DY: GOTO 250
300 PSET(A, B), 15
310 IF A<0 OR A>255 GOTO 370
320 IF B<0 OR B>191 GOTO 370
330 IF TIME>15000 GOTO 370
340 IF L=N THEN S70: REM"КРОЛИКИ СЪЕДИЛИ"
350 GOTO 160
370 SCREEN 0
380 COLOR , , 4: LOCATE 10, 12, 0
390 PRINT"ИГРА ОКОНЧЕНА": PRINT: PRINT

```

Задача для самостоятельной работы. Составить программу управления с помощью курсорных клавиш перемещения фигур, изображенных на рисунках I.46 - I.49.

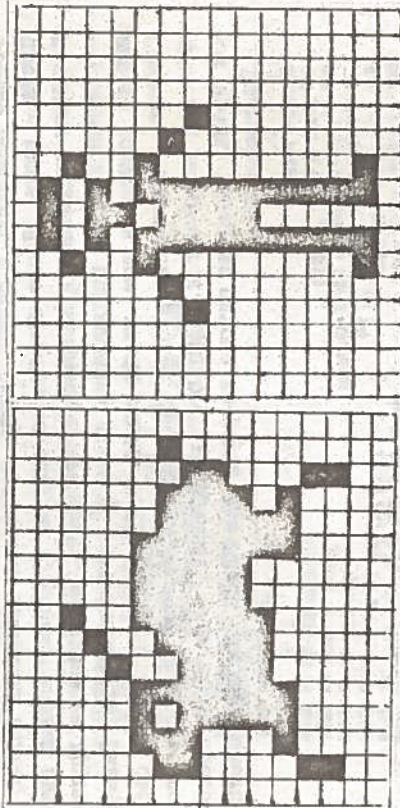


Рис. I.46

Рис. I.47

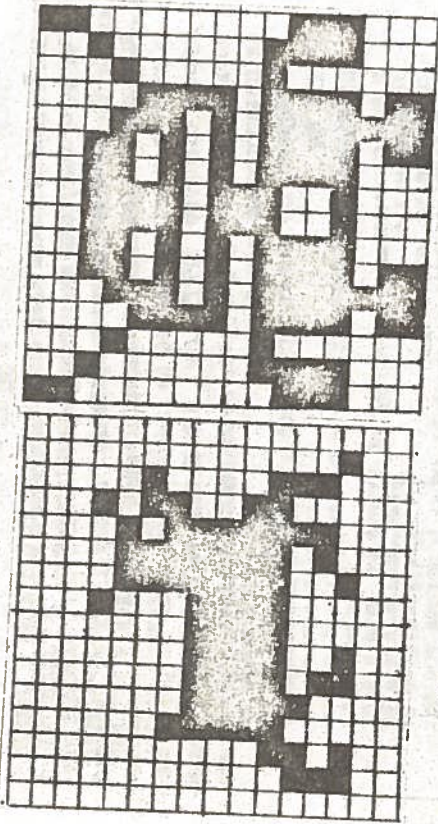


Рис. I.48

Рис. I.49

I.3.5. Управление столкновением спрайтов

Для спрайта считается столкнувшимся, если перекрываются хотя бы по одному их пикселу. Такое событие можно использовать для организации ветвления программы. При его наступлении выполнение основной программы прерывается оператором `SPRITE ON` и осуществляется переход на подпрограмму обработки прерывания (которая, например, воспроизводит столкновение спрайтов вспыхивания, звуковыми сигналами и т.д.). Обращение к подпрограмме обработки инициируется оператором

```
ON SPRITE GO SUB M ,
```

где `M` - номер строки, с которой начинается подпрограмма обработки прерывания.

Оператор прерывания `ON SPRITE` включает в программу обработку в ее начале. Сам он не начинает обработку прерываний, а лишь указывает номер строки `M`, с которой начинается подпрограмма обработки. Включение режима прерывания (режима ловушки) осуществляется оператором `SPRITE ON`. Когда ловушка сработает, произойдет автоматический переход к подпрограмме обработки, после выполнения которой оператор `RETURN` возвращает на место основной программы, где выполнение было прервано и вновь устанавливается оператор `SPRITE ON`.

Движение ON SPRITE можно выключить или приспособить с помощью следующих двух операторов: SPRITE OFF, который полностью отключает режим прерывания, и SPRITE STOP, который предполагает периодически анализировать условие, но результаты обработки непосредственно не проявляются. Если встретит следующий оператор SPRITE ON, MSX-Бейсик "вспоминает", появилось ли условие прерывания, после чего немедленно выполняет следующую подпрограмму обработки.

Программа 36 иллюстрирует использование оператора SPRITE ON.

Программа 36

```

5 REM СТОЛКНОВЕНИЕ СПРАЙТОВ
10 SCREEN 2,1
20 ON SPRITE GOSUB 100
30 SPRITE$(0)=STRING$(10,255)
40 SPRITE ON
50 FOR X=0 TO 255
60 PUT SPRITE$(X,90),15,0
70 PUT SPRITE 1,(240-X,90),1,0
80 NEXT X
90 GOTO 50
100 SPRITE OFF
110 FOR I=1 TO 2:COLOR ,,RND(1)*16:BEER:NEXT I
120 SPRITE ON
130 RETURN
    
```



Рис. 1.50

Программа 36 задает движение двух квадратов - белого и черного - по одной горизонтальной $Y = 90$ (первый из них движется в левую плоскость слева направо, а второй - в первой плоскости справа налево, см. рисунок 1.50). Их столкновение в центре экрана сопровождается генерируемыми подпрограммой обработки прерывания световыми вспышками и звуковыми сигналами, выдаваемыми оператором BEER (бип-бип).

Программа 37 определяет компьютерную игру, в которой пользователь может стрелять из пусковой установки (она может перемещаться вверх и вниз по вертикали $X = 24$ при нажатии на курсорные клавиши \uparrow и \downarrow) по движущейся мишени (она является квадратом, см. рисунок 1.51). Введя установку на нулевой уровень, пользователь призывает выстрел нажатием на клавишу пробела. Попадание выдвинутой

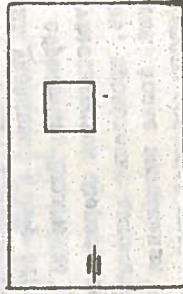


Рис. 1.51

ракеты в цель сопровождается световыми вспышками и звуковыми сигналами. В этой программе прерывания наступают в связи с появлением одного из двух событий - столкновения спрайтов или нажатия клавиши пробела (наступление последнего события проверяется функцией STRIG(0)).

Программа 37

```

5 REM ДВИЖУЩАЯСЯ МИШЕНЬ
10 SCREEN 2,0
20 ON STRIG GOSUB 210
30 ON SPRITE GOSUB 230
40 SPRITE$(0)=STRING$(8,255)
50 SPRITE$(1)=CHR$(240)+CHR$(63)+CHR$(240)
60 H=:S:L=:1
70 STRIG(0) ON:SPRITE ON
80 A=0:B=0:X=24:Y=96
90 V=0:W=96:F=0
100 PUT SPRITE 0,(X,Y),15,1
110 PUT SPRITE 1,(V,W),15,0
120 IF STICK(0)=1 THEN Y=Y-2
130 IF STICK(0)=3 THEN Y=Y+2
140 IF Y<=16 THEN Y=16
150 IF Y>=180 THEN Y=180
160 V=V-H:W=W-L
170 IF F=1 THEN GOSUB 280
180 IF Y<0 OR Y>190 THEN H=-H
190 IF W<0 OR W>160 THEN L=-L
200 GOTO 100
210 A=X+8:B=Y+2:F=1
220 RETURN
230 COLOR,,RND(1)*16:BEER
240 V=80:W=96
250 F=0:PUT SPRITE 2,(0,0),4,1
260 X=16:Y=100
270 RETURN
280 PUT SPRITE 2,(A,B),15,1:A=A+B
290 IF A>250 THEN F=0:PUTSPRITE2,(0,0),4,1
300 RETURN
    
```

1.3.6. Ввод - вывод информации в графическом режиме

В графическом режиме ввод данных с клавиатуры осуществляется с помощью специальных функций INPUT\$, INPUT\$, а вывод информации на экран - функцией PRINT#.

Вывод на экран в графическом режиме осуществляется с помощью устройства с именем "GRP:". Это устройство инициализируется оператором OPEN, имеющим следующий формат:
OPEN "GRP:" AS #k,

Программа 38 иллюстрирует применение оператора PRINT #

Программа 38

```
10 SCREEN 2
20 OPEN "GRP1" AS#1
30 PSET (5,5)
40 PRINT#1, "ПРИВЕТ"
50 GOTO 50
```

В программе 39 предлагается ответить на вопрос: "Существует ли треугольник с вершинами A(-2,-3), B(2,5), C(6,1)?" . Ввод ответа "существует" ("не существует") осуществляется нажатием на курсорную клавишу **[↑]** (↑) . Введенный ответ оценивается ("правильно", "неправильно").

Программа 39

```
10 SCREEN 7
20 OPEN "GRP1" AS#1
30 W$="ОТВЕТИТЕ НА ВОПРОС: СУЩЕСТВУЕТ ЛИ ТРЕУГОЛЬНИК"
40 W1$="С ВЕРШИНАМИ A(-2,-3), B(2,5), C(6,1)?"
50 P1$="ОТВЕЧАЯ НА ВОПРОС, НАЖМИТЕ НА КУРСОРНУЮ КЛАВИШУ,"
60 P2$="СООТВЕТСТВУЮЩУЮ ВЫБРАННОМУ ВАРИАНТУ ОТВЕТА"
70 DRAW"BM5,20":PRINT#1,W$
80 DRAW"BM5,30":PRINT#1,W1$
90 DRAW"BM5,50":PRINT#1,P1$
100 DRAW"BM5,60":PRINT#1,P2$
110 DRAW"BM150,120":PRINT#1,"СУЩЕСТВУЕТ"
120 DRAW"BM350,115":PRINT#1,"↑"
130 DRAW"BM350,123":PRINT#1,"↓"
140 DRAW"BM150,145":PRINT#1,"НЕ СУЩЕСТВУЕТ"
150 DRAW"BM350,140":PRINT#1,"↓"
160 DRAW"BM350,148":PRINT#1,"↑"
170 R=STICK(0)
180 IF R=0 THEN 170
190 IF R=1 THEN 220
200 DRAW"BM40,170":PRINT#1,"НЕПРАВИЛЬНО"
210 GOTO 230
220 DRAW"BM40,190":PRINT#1,"ПРАВИЛЬНО"
230 GOTO 230
```

Функция IKEY\$

Эта функция читает символ из буфера клавиатуры. Если символ отсутствует, значением IKEY\$ является пустой символ. В момент нажатия клавиши IKEY\$ получает значение символа из буфера, который будет потерян, если его сразу не сохранить. Для сохранения этого символа в качестве значения переменной A\$ необходимо выполнить присваивание A\$ = IKEY\$. Программа 40 иллюстрирует применение функции IKEY\$.

Программа 40

```
5 REM КЛАВИША
10 SCREEN 2
20 OPEN "GRP1" AS#1
30 LINE (0,20)-(256,100),6,8F
40 PSET (10,40):PRINT#1,"НАЖМИТЕ ЛЮБУЮ КЛАВИШУ"
50 A$=INKEY$
60 IF A$="" THEN 50
70 PSET (10,70):PRINT#1,"ВЫ НАЖАЛИ КЛАВИШУ";A$;" "
80 FOR I=1 TO 40:NEXT I
90 GOTO 30
```

Использование функции IKEY\$ допускает возможность "параллельной обработки" - если клавиша не нажата, можно выполнять какое-либо другое задание. Эта возможность иллюстрируется программой 41, осуществляющей ввод двузначного числа, при этом ожидание ввода цифр числа сопровождается выводом в соответствующем месте экрана мерцающей черты.

Программа 41

```
5 REM МЕРЦАЮЩАЯ ЧЕРТА
10 SCREEN 2,2
20 OPEN "GRP1" AS#1
30 SPRITE$(0)=STRING$(7,0)+CHR$(255)
40 PRESET (10,10)
50 PRINT#1,"ВВЕДИТЕ ДВУЗНАЧНОЕ ЧИСЛО"
60 X=10:Y=30
70 PRESET (X,Y)
80 A$=INKEY$
90 IF A$="" THEN GOSUB 170:GOTO 90
100 A=VAL(A$):PRINT#1,A;
110 X=X+8
120 B$=INKEY$
130 IF B$="" THEN GOSUB 170:GOTO 120
140 B=VAL(B$):PRINT#1,B
150 PRESET(10,50):PRINT#1,"ВВОД ЗАКОНЧЕН"
160 GOTO 150
170 Z=2*X
180 PUTSPRITE 0,(Z,Y+5),3,0
190 FOR I=1 TO 100
200 PUTSPRITE 0,(Z,Y+5),0,0
210 RETURN
```

В программе 41 используется функция VAL(" ") , преобразующая формат числа из строкового в числовой. Так, результатом выполнения функции VAL("243.6") является число 243.6. При вводе пустой строки как аргумента значением функции будет ноль: VAL("") = 0.

Функция INPUT \$

Эта функция выполняет многосимвольный ввод и имеет формат:

A \$ = INPUT\$(K),

где K - число символов, вводимое с клавиатуры. Так, оператор A \$ = INPUT\$(4) присваивает переменной A \$ четыре введенных символа переменной A \$. При этом введенные символы не дублируются на экране. В программе 42 с помощью функции INPUT\$(6) осуществляется ввод шестизначного числа.

Программа 42

```

10 SCREEN 2
20 OPEN "GRP:" AS#1
30 PSET (10,10)
40 PRINT#1,"ВВЕДИТЕ ШЕСТИЗНАЧНОЕ ЧИСЛО"
50 A$=INPUT$(6)
60 A=VAL(A$)
70 PSET (30,30)
80 PRINT#1,"ВЫ ВВЕЛИ ЧИСЛО: ";A
90 GOTO 90

```

Программа 43 осуществляет ввод числа, предложенного в естественной форме записи, с подтверждением введенных цифр на экране.

Программа 43

```

5 REM ВВОД ЧИСЛА С КОНТРОЛЕМ
10 SCREEN 7
20 OPEN "GRP:" AS#1
30 A$=""
40 PRESET(20,10):PRINT#1,"ВВЕДИТЕ ЧИСЛО, ЗАВЕРШАЯ ВВОД НАЖА-"
50 PRESET(20,20):PRINT#1,"ТИЕМ НА КЛАВИШУ ВОЗВРАТА КАРЕТКИ: ";
60 B$=INKEY$
70 IF B$="" THEN 60
80 IF B$=CHR$(13) THEN 110
90 PRINT#1,B$;A$=A$+B$
100 GOTO 60
110 PRESET(20,40):A=VAL(A$):PRINT#1,"ВЫ ВВЕЛИ ЧИСЛО ";A
120 GOTO 120

```

Программа 44 позволяет при вводе числа исправлять неправильно введенные цифры (если ошибка замечена до нажатия на клавишу возврата каретки). Для исправления ошибки необходимо нажать на клавишу **BS** и ввести на место стёртой цифры новую цифру. Ввод числа, как и в предыдущей программе, завершается нажатием на клавишу возврата каретки **↵**.

Программа 44

```

5 REM ВВОД ЧИСЛА С КОНТРОЛЕМ И ИСПРАВЛЕНИЕМ
10 SCREEN 7
20 OPEN "GRP:" AS#1
30 PRESET(10,10):PRINT#1,"ВВЕДИТЕ ЧИСЛО, ЗАВЕРШАЯ ВВОД НАЖА-"
40 PRESET(10,20):PRINT#1,"ТИЕМ НА КЛАВИШУ ВОЗВРАТА КАРЕТКИ: ";
50 A$=""
60 PRESET(300,20)
70 B$=INPUT$(1)
80 IF B$=CHR$(8) THEN 130
90 IF ASC(B$)=13 THEN 180
100 PRINT#1,B$;
110 A$=A$+B$
120 GOTO 70
130 K=LEN(A$)-1:IF K<0 THEN 50
140 A$=LEFT$(A$,K)+" "
150 PRESET(300,20):PRINT #1,A$;A$=LEFT$(A$,K)
160 PRESET STEP (-8,0)
170 GOTO 70
180 A=VAL(A$)
190 PRESET(10,60):PRINT#1,"ВЫ ВВЕЛИ ЧИСЛО ";A
200 GOTO 200

```

В программе 44 используется функция ASC(B \$), преобразующая первый байт строкового выражения в числовой формат (значением этого байта является целое число, соответствующее коду символа). Кроме того, в программе используются функции обработки строк LEN(A \$) и LEFT\$(A \$,K), первая из которых вычисляет длину строки символов, а вторая - выдает K левых символов строкового выражения A \$. Например,

LEN("информатика") = 11,
LEFT\$("информатика",4) = "инбо".

Программой 44 можно воспользоваться не только в седьмом, но и в шестом экранном режиме. Нижеследующая программа 45 может быть использована во втором, четвертом, шестом, седьмом и восьмом экранных режимах.

Программа 45

```

10 SCREEN 2
20 OPEN "GRP:" AS#1
30 DRAW"BM3,20":PRINT#1,"ВВЕДИТЕ ЧИСЛО, ЗАВЕРШАЯ ВВОД"
40 DRAW"BM3,80":PRINT#1,"НАЖАТИЕМ НА КЛАВИШУ ВОЗВРАТА"
50 A$=""
60 B$=INPUT$(1)
70 IF B$=CHR$(8) THEN 110
80 IF B$=CHR$(13) THEN 140
90 PRINT#1,B$;A$=A$+B$
100 GOTO 60

```

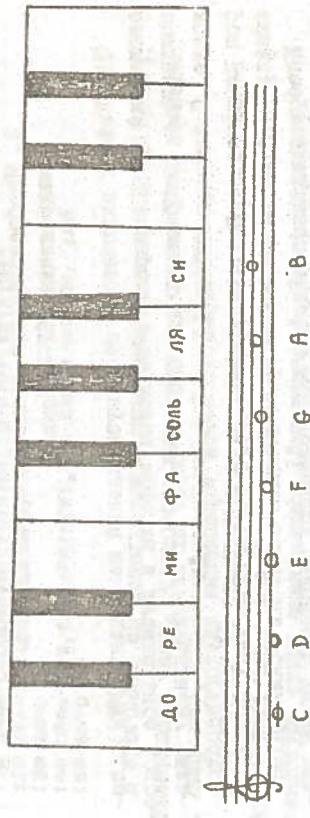



Рис. 2.1

Кодирование октав. Всего МММ допускает восемь октав. Ноты основного музыкального звукоряда исполняются компьютером в четвертой октаве (в общепринятой музыкальной нотации она соответствует первой октаве). Ноты пятой октавы, расположенные над основным звукорядом, изображены на рисунке 2.2 (в общепринятой нотации пятая октава соответствует второй).

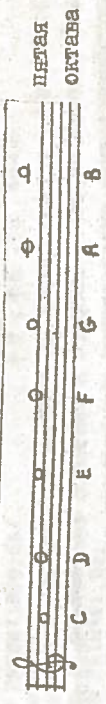


Рис. 2.2

На рисунке 2.3 изображены ноты шестой октавы (соответствующей третьей октаве в общепринятой нотации).

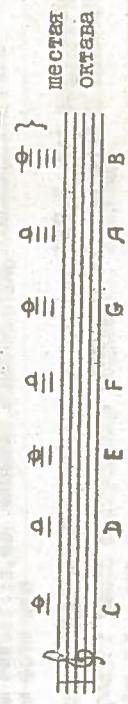


Рис. 2.3

На рисунке 2.4 изображены ноты третьей октавы (соответствующей малой октаве в общепринятой нотации).

В теории музыки ноты малой октавы принято записывать в басовом ключе, как это изображено на рисунке 2.5.

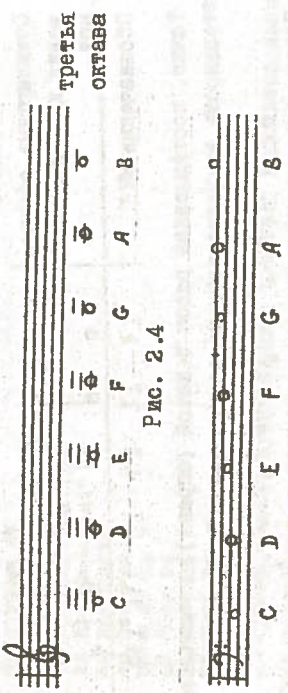


Рис. 2.4

Рис. 2.5

Задание для самостоятельной работы. Постройте графическое изображение нот первой, второй, седьмой и восьмой октав.

При программировании музыкальных текстов октава указывается параметром 0 (буква O) с последующим номером желаемой октавы. Если октава специально не указана, компьютер будет исполнять мелодию в четвертой октаве. Команда 0 является групповой, она влияет на все последующие ноты, пока октава не будет переопределена. Например, команда PLAY "CDEFGABO5CDEFGAB" проигрывает гамму четвертой и пятой октав.

Альтерация (изменение, ступеней музыкального звукоряда осуществляется с помощью знаков бемоля (b - знак понижения, при кодировании обозначается знаком "-") и диеза (# - знак повышения, обозначается знаком "+"). Например: PLAY "CC+DD+EFF+CC+AA+BB"

Пауза (перерыв в звучании). В основной последовательности ноты при звучании непрерывно сменяют друг друга. При необходимости включить паузы между нотами их определяют с помощью параметра P и вводят в строку. Например: PLAY "CCRC04CRC05CR".

Обозначения длительностей нот. Длительности нот в МММ обозначаются параметром L, за которым следует число от 1 до 64. По умолчанию принимается длительность L4, что соответствует четверти. Соответствие между стандартными символами, обозначающими длину ноту, половину, четверть, шестнадцатую, тридцать-вторую, шестидесятчетвертую и обозначениями МММ следующее:

Стандартные обозначения длительности нот

Обозначения ММЯ

L1 L2 L4 L8 L16 L32 L64

Точка, поставленная рядом с нотой (справа), увеличивает длительность звучания ноты на ее половину. Например:

$d. = d + \cdot$

Паузы, так же как и ноты, могут быть различной длительности. Соответствие между стандартными обозначениями длительности паузы и обозначениями ММЯ следующие:

Стандартные обозначения	
Обозначения ММЯ	R1 R2 R4 R8 R16 R32

Можно задавать длительности нот и пауз индивидуально к каждой ноте (паузе), сопроводжая название ноты (паузы) указанием ее длительности. Так, например, команда PLAY "L4CRD18C14CRD" равносильна следующей: PLAY "S4R4D4C8R16R16".

2.3. Программирование одноголосия

При программировании нотных записей необходимо определить названия нот, их длительности, номер октавы, в которых они должны исполняться, наличие бемоля или диэза.

Рассмотрим простой пример нотной записи.

Так как эта запись содержит четыре такта, то для ее кодировки достаточно четырех строчковых переменных. В нижеприводимой программе 47 эти переменные обозначены A#, B#, C#, D#.

Программа 47

```

10 A#="O4E4A4B+4A4"
20 B#="B4A4G4A4"
30 C#="G2F2"
40 D#="F4R4"
50 PLAY A#
60 PLAY B#
70 PLAY C#
80 PLAY D#

```

Программирование музыкального текста с использованием строчковых переменных может повлечь за собой искажение в воспроизведении мелодии там, где встречаются две или более одинаковых по высоте ноты. Избежать этого явления можно, используя для записи кодов нот блок данных. Этот эффект иллюстрирует программа 48, реализующая ту же нотную запись, что и программа 47.

Программа 48

```



10 DATA O4E4,A4,G+4,A4,B4,A4,G4,A4,G2,F2,F4,R4
20 FOR I=1 TO 12
30 READ A#
40 PLAY A#
50 NEXT I

```

В первом такте рассматриваемого нотного отрывка при ноте G выставляется знак диэза. Он относится к так называемым случайным знакам альтерации, т.е. к знакам, стоящим непосредственно перед нотами. Знаки альтерации, стоящие при ключе, называются ключевыми. Они сохраняют свое действие на протяжении всего музыкального произведений и во всех октавах, в то время как случайные знаки альтерации действуют только в пределах данного такта.

В нижеприведенном примере нотной записи используются ключевые знаки альтерации.

В этом отрывке ключевым знаком альтерации является знак бемоля, действующий на ноты B и E. В рассматриваемой нотной записи используются знак лиги (—) или (—) . Лига означает связанное исполнение. Если две ноты или группа нот связаны знаком лиги, то коды таких нот включаются в блок данных без

разделения их записями. Отметим также, что следующие друг за другом отдельные восьмые ноты  для удобства записи объединены в группу . Обсуждаемая нотная запись реализуется программой 49.

Программа 49

```
10 DATA E-8, D1DE, F+8F+8, G8, A8, OSE-8E-8, DE, C4E-1B-8, D8D8, B8
20 FOR I=1 TO 10
30 READ A#
40 PLAY A#
50 NEXT I
```

Программа 50 реализует неперывную нотную запись.



Программа 50

```
5 REM ЗА, УХНЕМ
10 DATA O5C8O4A8, O5D4, O4A4A8, A8, O8C8O4A8, O5D4, O4A4A8, A8
20 DATA O5C4, O5F4, O5E8F16E16, O5D4, O5C8O4A8, O5D4, O4A4A8, A8
30 FOR I=1 TO 15
40 READ F#
50 PLAY F#
60 NEXT I
```

2.4. Регулировка темпа и громкости

Можно изменить фактическую длительность сразу всех нот (не меняя их относительных длительностей), регулируя тем самым темп музыки, с помощью параметра T в следующем за ним числе от 32 до 255. Это число равно числу четвертей, звучащих в течение минуты. По умолчанию темп имеет значение, равное 120. Это означает, что за минуту можно исполнить 120 четвертей или 30 целых нот. Чем больше T, тем быстрее темп. Программа 51 иллюстрирует эффект изменения темпа.

Программа 51

```
5 REM ИЗМЕНЕНИЕ ТЕМПА
10 A#="L8DF+AF+D"
20 PLAY"TZ": PLAY A#
30 PLAY"764": PLAY A#
40 PLAY"128": PLAY A#
50 PLAY"1255": PLAY A#
```

Громкость звучания можно регулировать командой V. Параметр громкости может принимать значения от 0 до 15 (V0 - очень тихо, V15 -громко до крика). По умолчанию принимается V8. Программа 52 иллюстрирует эффект изменения громкости.

Программа 52

```
10 REM ИЗМЕНЕНИЕ ГРОМКОСТИ
20 FOR I=5 TO 13 STEP 2
30 PLAY"U-I;L4F+AF+D"
40 NEXT I
```

Приведем теперь примеры программ, в которых используется регулировка темпа и громкости звучания.

Программа 53 исполняет отрывок из гавота С. Прокофьева, нотная запись которого приведена ниже.



Программа 53

```
10 REM ГАВОТ ПРОКОФЬЕВА
20 DATA B8, A8, B+8, A8, O5A4, O4A4, O5G4, O4G4
30 DATA O5F+4, O4F+4, B4, O5D+4, C+8, D+8, E8, F8, G+4, A+4, B4, B4
40 FOR I=1 TO 20
50 PLAY "T:120V10"
60 READ F#
70 PLAY F#
80 NEXT I
```

Программа 54 исполняет вальс из кинофильма "Джентльмены удачи", нотная запись которого приведена ниже.

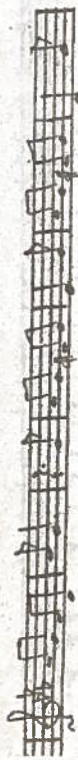


Таблица 2

Такт	1	2	3	4
1 ГОЛОС	04E4 04A4 04B4 05C+4 04B4 04A4 04E2 04F+4 04F+2F+4			
2 ГОЛОС	04E4 04C+4 04E4 04A4 04E+4 04C+4 04D2 04D4 04D2D4			
Такт	5	6	7	
1 ГОЛОС	04F+4 04G+4 04A4 04B4 04G+4 04E4 04E2 04A4			
2 ГОЛОС	04F+4 04G+4 04A4 04C+4 04E4 04D4 04C+2 04C+4			
Такт	8	9	10	
1 ГОЛОС	04A2A4 04G+4 04A4 04G+4 04C+2C+4			
2 ГОЛОС	04C+2C+4 03B4 04C+4 03B4 04C+2C+4			

В реализующей рассматриваемое звукологограмме 55 копн нот голосов записаны в слока данных попарно (по одному от каждого голоса).

Программа 55

- 10 REM ШКОЛЬНЫЕ ГОЛЫ
- 20 DATA E4, E4, A4, C+4, B4, E4
- 30 DATA 05C+4, A4, 04B4, E+4, A4, C+4
- 40 DATA B2, D2, F+4, D4
- 50 DATA F+2F+4, D2D4
- 60 DATA F+4, F+4, G+4, G+4, A4, A4
- 70 DATA B4, G+4, G+4, E4, E4, D4
- 80 DATA B2, C+2, A4, C+4
- 90 DATA A2A4, C+2C+4
- 100 DATA G+4, 03B4, A4, 04C+4, G+4, 03B4
- 110 DATA B4, 04D4, A4, 03B4, G+4, B4
- 120 DATA C+2C+4, 04C+2C+4
- 130 PLAY "V10T200", "V10T200"
- 140 FOR I=1 TO 50 STEP 2
- 150 READ A#, B#
- 160 PLAY A#, B#
- 170 NEXT I

Примером трехголосия является фрагмент из латышской народной песни "Вей, вей, ветерок", нотная запись которого приведена ниже.



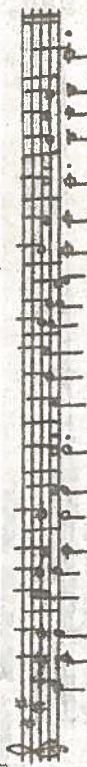
Программа 54

- 10 REM ВАЛЬС ИЗ К-Ф "АЖЕНТ ЛЬМЕНЫ УДАЧИ"
- 20 DATA G#, A-#, G#, F+#, G#, E-4E-#, G#, A-#, G#, F+#, G#
- 30 DATA E-4E-#, G#, 03E-#, D#, C#, 04G#, B-#, A-#, G#, A-#, G#, F+#, G#
- 40 DATA R4R#, F#, B-#, A-#, G#, A-#, F4F#, F#, B-#, A-#, G#, A-#, G#, F+#, G#
- 50 DATA F4F#, F#, B-#, A-#, G#, A-#, F4F#, F#, B-#, A-#, G#, A-#, G#, F+#, G#
- 60 DATA R4R#, G#, A-#, G#, F+#, G#, E-4E-#, G#, A-#, G#, A-#, G#, F+#, G#
- 70 DATA D4D#, G#, A-#, G#, F+#, G#, B-#, A-#, G#, A-#, G#, F+#, G#
- 80 DATA R4R#, 05C#, F#, E-4, D16, E-16, D4D#, C#, F#, E-#, D16, E-16
- 90 DATA D4D#, C#, F#, E-#, F#, E-#, D16, E-16, 04B#, G#, 03D#, C4C#, R#
- 100 PLAY "V10T150"
- 110 FOR I=1 TO 96
- 120 READ F#
- 130 PLAY F#
- 140 NEXT I
- 150 RESTORE
- 160 GOTO 100

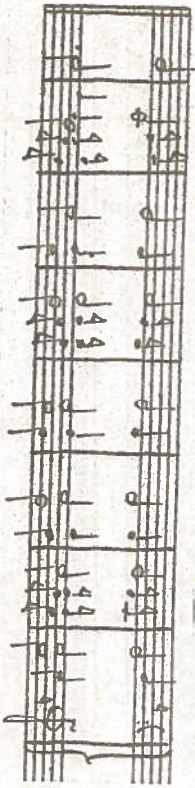
2.5. Программирование многоголосия.

В предыдущих музыкальных программах был задействован лишь один звуковой канал. Исполняя все три канала, можно создавать аккорды при одновременном звучании трех голосов или вести в отдельности мелодию и аккомпанемент. Синхронное звучание трех голосов осуществляется оператором PLAY с тремя выражениями, разделенными запятыми. При программировании многоголосия необходимо соблюдать принцип одинаковой длительности нот, приравняв их одновременно по параллельным каналам.

Приведенный ниже музыкальный фрагмент, состоящий из десяти тактов, является примером двухголосия.



В таблице 2 приведена нотная запись вокалов нот, соответствующая этому фрагменту.



Партии трех голосов, соответствующие этой нотной записи, приведены в таблице 3.

Таблица 3

Такт	1	2	3	4	
1 голос	04F4 05C2	04A8 04C8	04F4 04B-4	05D4 05C2	
2 голос	04F4 04F2	04C8 04E8	04F4 04D4	04E4 04F2	
3 голос	03F4 03A2	04A8 03B-8	05A2 03B-4	03B-2 03B-4 03A2	
Такт	5	6	7	8	
1 голос	04G8 04A8	04B-2 05C4	04F2 04B-8	04A8 04C2	04F2F4
2 голос	04D8 04B8	04C2 04E4	04F2 04D8	04F8 04F4E4	04F2F4
3 голос	03B-8 03A8	03E2 03B-4	03A2 03G8	03A8 04C2	03F2F4

Коды нот этой таблицы расположены в блоке данных программы 56, реализуемой рассматриваемое трехголосие, по столбцам (сначала элементы первого столбца, затем второго и т.д.). В процессе работы программы они считываются по три элемента за цикл, ик значащие присваиваются строковым переменным A\$, B\$, C\$.

Программа 56

```

10 REM BEY, BETEROK
20 DATA F4, F4, 03F4, 05C2, F2, A2, 04A8, 04C8, A8, 68, E8, 03B-8, F4, F4, A2
30 DATA B-4, D4, B-4, 05D2, F2, B-2, D4, E4, B-4, C2, F2, 03A2
40 DATA 04G8, D8, B-8, A8, D8, A8, B-2, G2, E2, 05C4, E4, B-4, 04F2, F2, A2
50 DATA B-8, D8, 68, A8, F8, A8, 62, F4E4, C2, F2F4, F2F4, F2F4
60 PLAY "T100"
70 FOR I=1 TO 18
80 READ A$, B$, C$
90 PLAY A$, B$, C$
100 NEXT I
    
```

Еще одним примером трехголосия является приведенный ниже фрагмент из "Маленькой ночной серенады" Моцарта.



Потактная кодировка нот этого фрагмента приведена в таблице 4.

Таблица 4

Такт	Голос	Код
1	1	05D8 05G8 05E8
	2	F8 F8 F8
	3	F8 F8 F8
2	1	06D8 06D8 05D8 06D8 06D8 06D8 06D8
	2	03E8 04D8 03E8 04D8 03E8 04D8 03E8 04D8
	3	03G8 F8 03G8 F8 03G8 F8 03G8 F8

Продолжение таблицы 4

Такт	Голос	Код									
3	1	05F+8	05F+8	05F+8	05F+8	05G8	05G8	05G8	05G8	05G8	05G8
	2	03A8	03A8	04C8	03B8	04D8	03B8	04D8	03B8	04D8	04D8
	3	03D8	R8	03D8	R8	03G8	R8	03G8	R8	03G8	R8
4	1	06C8	06C8	06C8	06C8	05B8	05B8	05B8	05B8	05B8	05B8
	2	03A8	04D8	03A8	04D8	03B8	04D8	03B8	04D8	03B8	04D8
	3	03F+8	R8	03F+8	R8	03G8	R8	03G8	R8	03G8	R8
5	1	05A8	05A8	R8	R8	R8	05D8	05G8	05G8	05G8	05G8
	2	03F+8	04B8	04A8	04G8	04F+8	04E8	04D8	04C8	04B8	04C8
	3	03D8	R8	R8	R8	R8	R8	R8	R8	R8	R8
6	1	06D8	06D8	06D8	06D8	06D8	06D8	06D8	06D8	06D8	06D8
	2	03B8	04D8	03B8	04D8	03B8	04D8	03B8	04D8	03B8	04D8
	3	03G8	R8	03G8	R8	03G8	R8	03G8	R8	03G8	R8
7	1	05F+8	05F+8	05F+8	05F+8	05G8	05G8	05G8	05G8	05G8	05G8
	2	03A8	04C8	03A8	04C8	03B8	04C8	03B8	04C8	03B8	04C8
	3	03D8	R8	03D8	R8	03B8	R8	03B8	R8	03B8	R8
8	1	05A8	05A8	05A8	05A8	05D8	05D8	05D8	05D8	05D8	05D8
	2	03B8	03G8	03B8	03G8	03F+8	03C8	03F+8	03C8	03F+8	03C8
	3	03C8	R8	03C8	R8	05D8	R8	05D8	R8	05D8	R8
9	1	05G2G8									
	2	03B2B8									
	3	03G2G8									

В соответствии с этой таблицей построена программа 57.

Исполняемая "Маленькую ночную серенаду" Моцарта.

Программа 57

5 REM МАЛЕНЬКАЯ НОЧНАЯ СЕРЕНАДА
 10 DATA 05D8, R8, R8, G8, R8, R8, B8, R8, R8
 20 DATA 06D8, 03B8, 03G8, D8, 04D8, R8, D8, 03B8, G8, D8, 04D8, R8
 30 DATA D8, 03B8, G8, D8, 04D8, R8, D8, 03B8, G8, D8, 04D8, R8
 40 DATA 05F+8, 03A8, D8, F+8, 04C8, R8, F+8, 03A8, D8, F+8, 04C8, R8
 50 DATA G8, 03B8, G8, 04D8, R8, G8, 03B8, G8, 04D8, R8
 60 DATA 04C8, 03A8, F+8, C8, 04D8, R8, C8, 03A8, F+8, C8, 04D8, R8
 70 DATA 03B8, 03B8, G8, B8, 04D8, R8, B8, 03B8, G8, B8, 04D8, R8
 80 DATA A8, 03F+8, D8, A8, 04B8, R8, R8, A8, R8, R8, 04B8, R8
 90 DATA R8, F+8, R8, D8, E8, R8, G8, D8, R8, R8, C8, R8
 100 DATA 06D8, 03B8, G8, D8, 04D8, R8, D8, 03B8, G8, D8, 04D8, R8

110 DATA D8, 03B8, G8, D8, 04D8, R8, D8, 03B8, G8, D8, 04D8, R8
 120 DATA 05F+8, 03A8, D8, F+8, 04C8, R8, F+8, 03A8, D8, F+8, 04C8, R8
 130 DATA G8, 03B8, G8, D8, 04D8, R8, G8, 03B8, G8, D8, 04D8, R8
 140 DATA A8, E8, C8, A8, G8, R8, A8, E8, C8, A8, G8, R8, F+8, D8
 150 DATA D8, C8, R8, E8, F+8, D8, F+8, 04C8, R8, G8, D8, 03B8, G8, D8
 160 PLAY "T128"
 170 FOR I=1 TO 66
 180 READ A\$, B\$, C\$
 190 PLAY A\$, B\$, C\$
 200 NEXT I

На компьютере можно исполнять не только трехголосную музыку гомофонно-гармонического склада (мелодия и аккомпанемент), например "Маленькую ночную серенаду" Моцарта, но и полифоническую музыку - трехголосные инвенции и фуги Баха. Возникающие при этом осложнения при кодировании музыкального текста связаны с необходимостью соблюдения одинаковых длительностей для трех голосов. В качестве иллюстрирующего примера приведем программу 58. Исполняющую фугу № 2 И.С. Баха, нотная запись которой приведена ниже.



С целью освоения техники синхронизации партий трех голосов приведем поэтапную кодировку нотной записи фуги Баха (см. таблицу 5).

Таблица 6

1-й такт	2-й такт			3-й такт				
	ГОЛОС			ГОЛОС				
1	2	3	1	2	3	1	2	3
05F8	05C8	05B	05E-8	05B	03G8	05C16	04G16	03E-16
05F16	05C16	04C16	05D16	04A-16	04C16	05C16	04G16	04C16
05D16	05C16	03B16	05D16	04A-16	03B16	05C16	04G16	03B16
05C16	R16	04C16	05E-8	04G8	04C8	05F-1604G16	03A16	
04E-16	R16	04C16	05F8	04F8	04D8	05G16	R16	03G16
04A-16	R16	03G16	04B16	04G16	03G16	05G16	R16	03F16
04G16	R16	03G16	04B16	04G16	03G16	05D16	04B16	03D16
04F8	03A-8	05C16	04F16	03A-16	03A-16	05E-1605C16	03C16	
05A-1605F16	04C16	05C16	04E-16	03A-16	03A-16	05E-1605C16	03E-16	
05B	05E-8	04C8	04F8	03A-8	03G16	05E-1604B16	03D16	
05F8	05D8	04D8	04B16	04D16	03F16	05E-4	05C4	03C4

Программа 58

5 REM OYTA N 2 BAKA
 10 DATA 05F8, 05C8, 05B, F16, C16, 04C16, D16, C14, 03B16, C16, R16, 04C16
 20 DATA 04E-16, R16, C16, A-16, R16, 03G16, B16, R16, G16, F8, 05, A-8
 30 DATA 05A-16, 05F16, 04C16, A-16, F16, 03B16, 05, E-8, 04C8, F8, D8, D8
 40 DATA E-8, 05, 03B8, D16, 04A-16, 04C16, D16, A-16, 03B16, E-8, 05, 04C8
 50 DATA F8, F8, 05, 04B16, G16, 03G16, R16, G16, 03B16, 05C16, F16, A-16
 60 DATA C16, E-16, A-16, 05, F8, A-8, 04B16, D16, G16
 70 DATA B16, D16, F16, 05C16, G16, E-16, C16, G16, 04C16
 80 DATA G16, G16, 03B16, F-16, G16, A16, G16, R16, G16, F16, F16
 90 DATA D16, B16, 03B16, F-16, D16, D16, F-16, 05C16, C16, E-16, C16, D16
 100 DATA E-16, C16, E-16, E-16, 04B16, D16, E-4, 05C4, C4
 110 PLAY "T98"
 120 FOR I=1 TO 37
 130 READ A8, B8, C4
 140 PLAY A8, B8, C4
 150 NEXT I

2.6. Спешальные звуковые эффекты

Одним из средств придания мультимедийному фрагменту большей выразительности является регулирование громкости звучания.

Самым простым способом изменения громкости является команда V. Более интересная возможность состоит в передаче управления громкостью генератору огибающей, программами

которой является период (шка модуляции - несущая частота звукового генератора) и форма (определяющая эволюцию звука).

Период огибающей задается параметром M с последующим числом от 1 до 65535. По умолчанию принимается M255. Форма огибающей устанавливается параметром S с последующим числом 1, 8, 10, 11. Например: P.LAU "M1600S80EFGAB".

Использование параметров M и S позволяет добиваться оттенков "стакато" к звуку. Эта возможность иллюстрируется программой 59.

Программа 59

5 REM КУПТАРСИТА
 10 DATA 04D16E-16F16E-16D16C1605B-16A16, R2, R2, R2, R2
 20 DATA 05B-1606C16D16E-16D16C1605B-16A16, R2, R2, R2
 30 DATA 05C1604B-16A16616F16E-16D16C16, R2, R2
 40 DATA 03D4, 03D4, 03D4, 03D4, 02D4, 04A4, 03D4, 02D4, 04F+4,
 50 DATA 04D8, 03D8, 02D8, 04C+4, 03D4, 02D4, 04E-8, 03D8, 02D8
 60 DATA 04D4, 03E4, 02E4, 03D4, 03E4, 02G4, 04B-4, 03E4, 02E4
 70 DATA 04E4, 03E4, 02E4, 03E4, 02E4, 03E4, 02E4, 04E-8, 03E8, 02E8
 80 DATA 04D4, 03D4, 02D4, 05C4, 03D4, 02D4, 04A4, 03D4, 02D4
 90 DATA 04F+4, 03D4, 02D801AB, R804D8, 03D4, 02D4, 04E-8, 03D8, 02D8
 100 DATA 04D4, 03E4, 02E4, 05D4, 03D4, 02D4, 04E-8, 03D8, 02D8
 110 DATA 04E4, 03E4, 02E4, 03E8, 02E8, 03E8, 02E8, 04C+4, 03E4, 02E4
 120 DATA 04E-8, 03E8, 02E8, 04D8, 03E8, 02E8, 04C+4, 03E4, 02E4
 130 DATA 04E-8, 03E8, 02E8, 04D8, 03E8, 02E8, 04C+4, 03E4, 02E4
 140 DATA 04D4, 03E4, 02E4, 03E8, 02E8, 03E8, 02E8, 04C+4, 03E4, 02E4
 150 DATA 03C4C4C4C8, 05D4D4B-28-4, 04B-4E4F+4E4, 05C4D4E4F+4E4
 160 DATA 03C4C4C4C8, 05D4D4B-28-4, 04B-4E4F+4E4, 05C4D4E4F+4E4
 170 DATA R805C+8D8C+8D4E-4, 04A8B-8A4B-4C4, 03E4E4E4E4E4E4
 180 DATA 05C404A2A4, 04A4D4C+4D4, 03D4D4D4D8
 190 DATA R8E-4D16C16D4B-4F+4, R805C4D4B-16A16E4C4, 03C4R4R403D4
 200 DATA 04E4R8E-8D8C803B-8AB, 03B-4R8R8R2, 02E4R8E-8D8C802B-8AB
 210 DATA 03E405G4R2, 03G4E4R2, 02E4E4R2
 220 DATA 04E8AB8-4B-8E4A16B-16, 03E4E4E4E4E4E4E4
 230 DATA 04A2E8E16B-80SD16G16D1604B-16, 03E4E4E4E4E4E4E4
 240 DATA 04E805G16G16E8F16F16F8E-16E-16E-8D16D16, 03E4E4E4E4E4E4E4
 250 DATA 05D2C8D16F+16A1606D16A16F+16, 03D4D4D4D4D4, 02D4D4D4D4
 260 DATA 05D8A16A16A8E16G16E8F+16F+16F+4, 03D4D4D4D4D4, 02D4D4D4D4
 270 DATA R8E-16E-16E-8D806D2, 03D4D4D4D4D4, 02D4D4D4D4
 280 DATA R805C4C8C404B-8AB, R804F+4F+4F+4F+4, 02D4D4D4D4
 290 DATA 04E4E8D2D4, 03E4E4E4E4E4, 02E4E4E4E4E4
 300 DATA 04E8A8B-8B-4E4A16B-16, 03E4E4E4E4E4E4E4
 310 DATA 04A2E4E16B-1605D16G16D1604B-16, 03E4E4E4E4E4E4E4
 320 DATA 04E805G16G16E8F16F16F8E-16E-16E-8D16D16, 03E4E4E4E4E4E4E4
 330 DATA 02E4E4E4E4E4, 05C1, 03C4C4C4C4C4, 02C4C4C4C4
 340 DATA 05C8E-4C5E8F+8E8E-8, 03C4C4C4C4C4, 02C4C4C4C4
 350 DATA 04B-80SD404B-805E8F+8E8E8E, 03E4E4E4E4E4E4E4
 360 DATA R806D805ABF+8D8C404B-8AB, R808F+8C8D404B+8E-4D8E, D1

SOUND

< номер регистра ПЭТ >, < записываемое значение > .
 При этом допускается запись значений в четырнадцатый регистр -
 ров ПЭТ, перенумерованных от 0 до 13. Содержимое всех этих четыр-
 нацати регистров определяет характер воспроизводимого звука.
 Программа 60 иллюстрирует использование команды SOUND (подро-
 нее об этой команде см. [2], [4]).

Программа 60

```

10 REM MORE
20 FOR I=0 TO 13
30 READ V: SOUND I, V
40 NEXT I
50 END
60 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
70 DATA 16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  
```

В программах, имитирующих э уковне эффекты, рекомендуется включать в качестве начальной команду BEEP (бип). Она индивидуа-
 зирует звуковую микросхему, устанавливая ее регистры по умолча-
 нно в первоначальное состояние. Иллюстрирующим примером являет-
 ся программа 61.

Программа 61

```

3 REM BOMBER
10 BEEP
20 FOR I=0 TO 13
30 READ V: SOUND I, V
40 NEXT I
50 END
60 DATA 200, 14, 220, 14, 240, 14, 0
70 DATA 188, 15, 15, 15, 0, 0, 0, 0
  
```

2.7. Использование переменных и подстрок

Параметр к любой команде ММЯ может записываться переменной,
 перед которой записывается знак равенства, а после имени пере-
 менной - точка с запятой. Например: T = I. В программе 62 пе-
 ременная I используется для регулирования темпа звучания.

Программа 62

```

10 FOR I=120 TO 256 STEP 10
20 PLAY "T=I:DEF".
30 NEXT I
  
```

Предположим теперь, что в музыкальном фрагменте какая-то

```

370 DATA 0464R40564R4, 03B-4R40464R4, 0364R40364R4
380 DATA 046BF+8668ABBB-4AB, 036BF+8668ABBB-4AB
390 DATA 026BF+8668ABBB-4AB, 05614F+1661A16B-16A16B-1606C16D2
400 DATA 0364646464, 0264646464, 04E-8D8E-8F86466E-8, 03C4C4C4C4
410 DATA 02C4C4C4C4, R4R805D806D4R1605616D1604B-16, 04D4036803686464
420 DATA 0264646464, 046BF+8668ABBB-4AB, 036BF+8668ABBB-4AB
430 DATA 026BF+8668ABBB-4AB, 05616F+1661A16B-16A16B-1606C16D2,
0364646464, 0264646464
440 DATA 04E-8D8E-8F86466E-8, 03C4C4C4C4, 02C4C4C4C4
450 DATA 04D8D8E8F+858ABBB-805C8, 0364646464, 0264646464
460 DATA 05D2E-4E-8D8, 0364646464, 0264646464
470 DATA 05D4D8C8, 03D4D4, 02D4D4, 05C8C16E-16D4, 03D4D4, 02D4D4
480 DATA R405C4D4D8C8, 03D4D4D4D4, 02D4D4D4D4
490 DATA 05C4C804B-8, 036464, 026464, 04B-8B-1605D16C4, 036464
500 DATA 026464, R4D4E-4D5C4C804B-8, 0364646464, 0264646464
510 DATA 04B-4B-8AB, 03D4D4, 02D4D4, 04A8A1605C1604B-4, 03D4D4
520 DATA 02D4D4, R4D4A4646BF+8, 03D4D4D4D4, 02D4D4D4D4
530 DATA 04G4R8E-8D8C803B-8AB, 03B-4R8R8R2, 0264R8E-8D8C802B-8AB
540 DATA 03640564R2, 036405R2, 026464R2
550 DATA 04D4, 03D4, 02D4, 05C4, 03D4, 02D4, 04A4, 03D4, 02D4, 04F+4
560 DATA 03D4, 02D801AB, R804D8, 03D4, 02D4, 04E-8, 03D8, 02D8, 04D8
570 DATA 03D8, 02D8, 04C+4, 03D4, 02D4, 04D4, 03D808, 02D808, 04D4, 03G4
580 DATA J264, 05D4, 0364, 0264, 04B-4, 0364, 0264, 0464, 02680368
590 DATA R8D8, 0364, 0264, 04E-8, 0368, 0268, 04D8, 0368, 0268, 04C+4
600 DATA 0364, 0264, 04D4, 0364, 026808, 04D4, 03D4, 02D4, 05C4, 03D4
610 DATA 02D4, 04A4, 03D4, 02D4, 04F+4, 03D4, 02D801AB, R804D8, 03D4
620 DATA 02D4, 04E-8
630 DATA 03D8, 02D8, 04D8, 03D8, 04C+4, 03D4, 02D4, 04D4, 03D8D8
640 DATA 02D8D8, 02D4, 04E-8, 04D4, 0364, 0264, 05D4, 05D4, 0364, 0264, 04B-4
650 DATA 0364, 0264, 0464, 0364, 0364, 0264, 04E-8, 0368, 0268, 04D8
660 DATA 0368, 0268, 04C+4, 03G4, 0264, 04D4, 03G4, 02680368
670 DATA 05E-464F+464, 05C4D464F+464
680 DATA 03C4C4C4C868, R803D8E-8D8E-4F+4, R8D48-8D5C8D4B-8D5C4D4A4
690 DATA 03C4C4C4C8C8, 05D404B-2B-4, 04B-464F+464, 0364646468D8
700 DATA R8D5C+8D8C+8D4E-4, 04A8B-8A4B-4C4, 0364646468
710 DATA 05C404A2A4, 04A4D4C+4D4, 03D4D4D4D8D8
720 DATA R8E-4D16C16D4B-4F+4, R8Q5C4Q4B-16A1664C4, 03C4R4R4D3D4
730 DATA 0464F8E-8D8C803B-8AB, 03B-4R8R8R2, 0264R8E-8D8C802B-8AB
740 DATA 0364056462, 03646462, 02646462
750 PLAY "T160M4000S1", "T160M4000S1", "T160M4000S1"
760 FOR I=1 TO 128
770 READ A$, B$, C$
780 PLAY A$, B$, C$
790 NEXT I
  
```

ММЯ позволяет достаточно хорошо управлять генерированием
 звуковых сигналов, соответствующих стандартной музыкальной
 шкале. Для воспроизведения же звуковых эффектов, подобных шуму
 прибора или sireне, требуется более точное управление программ-
 руемым звуковым генератором (ПЭТ). Для этой цели предназначена
 команда SOUND (sound - звук), обеспечивающая прямой до-
 ступ к ПЭТ. Она имеет формат:

фраза повторяется неоднократно. Вместо того, чтобы каждый раз переписывать эту фразу, можно обозначить ее как строковую переменную, а затем вызывать как подстроку с помощью команды X. Эта команда сопровождается именем строковой переменной и точкой с запятой, например XА ; (см. программу 63).

Программа 63

```
10 A$="116003F32F+3264"
20 PLAY "XA$;XA$;F+4F4D+4C20762"
30 PLAY "AXA$;XA$;F+4F4D+4C2"
```

Подпрограммы на МММ, определяемые строковой переменной, могут быть вложенными. Такую возможность иллюстрирует программа 64.

Программа 64

```
10 H$="C4F8C8808"
20 A$="02XH$;C8D8"
30 L$="03XH$;04C80388"
40 B$="06D4R804D4A805D806A8"
50 C$="05B88D4A405C4"
60 D$="06F16516"
70 PLAY "I:0M300088", "110M300088"
80 FOR I=1 TO 2
90 PLAY A$, B$, C$
100 PLAY A$, B$, "XD$;A1681606C16"
110 PLAY L$, B$, "XD$;B16C1605D4"
120 NEXT I
```

2.9. Задания для самостоятельной работы

В настоящем разделе приведены задания как на программирование музыкальных текстов (одноголосных, двухголосных и трехголосных), так и на отыскание на слух ошибок в мелодии, воспроизводимой компьютером по предложенной программе, и восстановление по ней музыкального текста.

Программирование одноголосия

Запрограммируйте предлагаемый музыкальный текст в соответствии с номером вашего варианта и исполните составленную программу на компьютере. При обнаружении ошибок на слуховое вос-

приятие внесите исправления в составленную программу.

Варианты заданий:

- 1. Русская народная былина "Соловей Будимирович".



- 2. "Высота, высота полнебесная" (из оперы "Садко" Н.А. Римского-Корсакова).



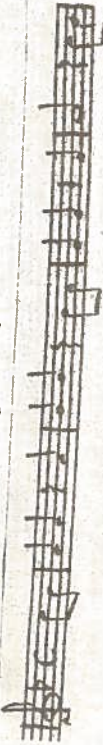
- 3. Русская народная былина "Илья Муромец и Соколов".



- 4. Русская народная былина "Вольга и Микула".



- 5. В. А. Моцарт. "Волшебная флейта".





6. Ф. Шопен. "Вальс".



7. Н.А. Римский-Корсаков. "Шехерсаза".



8. М.И. Глинка. "Жаворонок".



9. И.С. Бач. "Путь".



10. Ф. Шуберт. "Форель".



11. Г.Ф. Гендель. "Пассакалья".



12. С.С. Прокофьев. "Тавор".



Проверка слухового восприятия музыкальных текстов

Ниже приведены коды музыкальных отрывков из детских песен композитора В. Шаинского, содержащие ошибки копирования (вертикальные черточки в тексте означают границы тактов). Составьте, используя эти коды, программу, исправьте ее на компьютере и, обнаружив на слух ошибки, исправьте их. Определите название песни, из которой взят предложенный фрагмент. Восстановите нотную запись фрагмента.

№ п/п	Код музыкального текста
1	BA4A4B4 E-5 B5C5B4 A-4B4A-4B-4 B405D4C404B4 A-4B4C4A4 05 D EI 05C204E2
2	R2R405C304B-8 A4F4G-4B-4 F2 F4B8B 05C4 04F405E-404 A4 05C2R4C3C+8 05E-4C4R4 04A8B8 B4 A4R4B8A8 C4D8C3F4E8C8 F4A-4C4R4
3	R2C4E-4 A8B8 E-4F4E-4D4 C1 C4R4E4 F4 A4B-4 A4B-4
4	D8M404E-4 A-4 06C204F2 R4R8 A8 C8C8R8A-8 F8B8B8 A8 F8F8E-8C8 A4 F8A8 C8C8R8A8 F8F8B8 A8 G+8 A8C8 F8 E4D4
5	M B4A4 G4 A4 E-4A4G4A4 G2E2I E2R2 C4 G4F4G4 A4G4C4F4 C1
6	B4A8B8 A4C8D8 F8D8A+8B8 A4D8E8 C8E-8A+8B8 A4E8F+8 G4A-8B8 A2
7	R2R4A4 E-4A4E-4A4 G4G4 R4G4 E4C4 E4G4 A4A4R2
8	R4R8C8 A-8C8E-8C8 E4D8F8 C8F8D8F8 A-4C8B-8 05C804B-8 C4B-8C8E-8 C4 F8F8 C8F8C8A-8 B-4B-8
9	C8B8R4 C8B8R4 C8C8A-8 C8F8F8R8 C8R8R4 F8R8R4 F8A-8D8B8 E-8 E-8F8E-8 D8C8B-4 D4A-4
10	R2C4E-4 G4C4R2 C4 G2F8G4A-4 C1 R205C4 04E-4 D4D4R2 A2M8B F4G4 E-1 R2
11	G4C4 F4G4 B-2A-4R4 F4 F4E4E4 A-2C4 R4 E-4E-4D4E-4 C2 F4R4 C2A4R4 C2B-4R4
12	R4A8F8 F8E8B8E8D8 D8C8A804F8 F8C8B8E8 D4D8F8 A8A8C8F8 G4 A4 A8B8

Программирование двухголосия

При программировании двухголосия придерживайте синхронизацию длительностей голосов в соответствии с рекомендациями пункта

2.б. Верные записи:

1. Русская народная песня.

Two staves of musical notation for a Russian folk song. The top staff contains a melodic line with various note values and rests. The bottom staff contains a harmonic accompaniment with chords and single notes.

2. Английская народная песня "Гробы Гуд".

Two staves of musical notation for an English folk song. The top staff shows a melodic line, and the bottom staff shows a harmonic accompaniment.

3. Белорусская народная песня "Явля".

Two staves of musical notation for a Belarusian folk song. The top staff features a melodic line, and the bottom staff provides a harmonic accompaniment.

4. К.М. Вебер. Хор из оперы "Осерия".

Two staves of musical notation for a chorus from an opera. The top staff contains a melodic line, and the bottom staff contains a harmonic accompaniment.

5. Г. Хёрселл. Хор из оперы "Дилон и Эней".

Two staves of musical notation for an opera chorus. The top staff shows a melodic line, and the bottom staff shows a harmonic accompaniment.

6. И.С. Бак. Хорак № 102.

Two staves of musical notation for a chorale. The top staff contains a melodic line, and the bottom staff contains a harmonic accompaniment.

7. Д. Верди. Хор из оперы "Аида".

Two staves of musical notation for an opera chorus. The top staff features a melodic line, and the bottom staff features a harmonic accompaniment.

8. М.И. Глинка. Хор из оперы "Руслан и Людмила".

Two staves of musical notation for an opera chorus. The top staff shows a melodic line, and the bottom staff shows a harmonic accompaniment.

9. М.И. Глинка. Хор из оперы "Руслан и Людмила".

Two staves of musical notation for an opera chorus. The top staff contains a melodic line, and the bottom staff contains a harmonic accompaniment.

10. В.А. Моцарт. Хор из оперы "Свадьба Фигаро".

Two staves of musical notation for an opera chorus. The top staff features a melodic line, and the bottom staff features a harmonic accompaniment.

11. Песня Марии Стюарт.



12. М.И. Глинка-Иванов. "На севере диком".



Программирование трехголосия

Используя рекомендацию пункта 2.5, запрограммируйте ниже-приведённые фрагменты согласно номеру в вашего варианта. Особое внимание обратите на синхронизацию трех голосов.

Варианты заданий:

1. М.А. Балакирев. Увертюра к опере "Король Идр".



2. А.Л. Бородин. Опера "Князь Игорь".



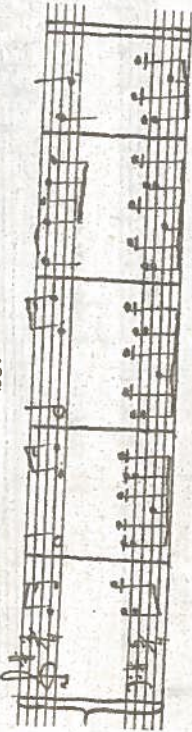
3. А.Л. Бородин. Опера "Князь Игорь".



4. И.С. Бах. "Прелюдия".



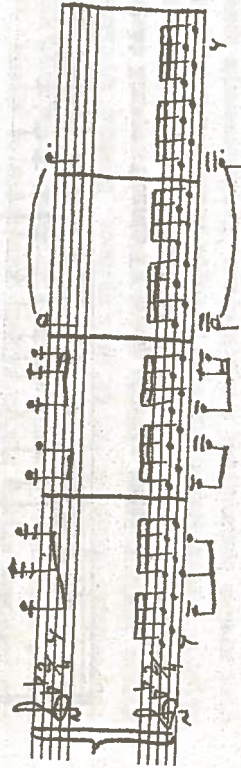
5. Л. Бетховен. Соната № 20.

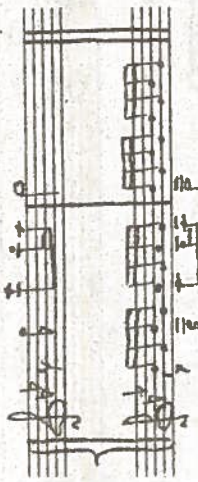


6. П.И. Чайковский. Опера "Пиковая дама".



7. П.И. Чайковский. Первая симфония, первая часть.

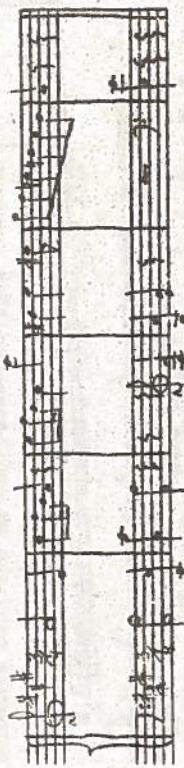




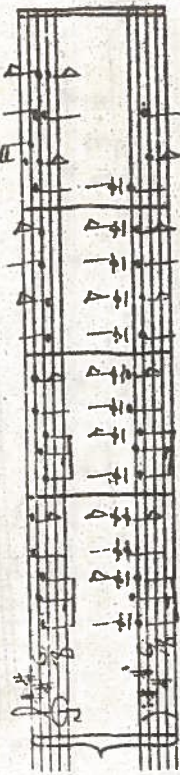
5. П. И. Чайковский. Шестая симфония, первая часть.



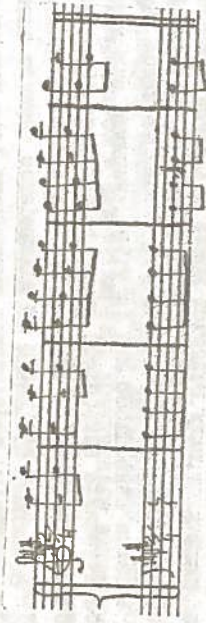
6. Ф. Ш. Шуберт. Соната для фортепиано.



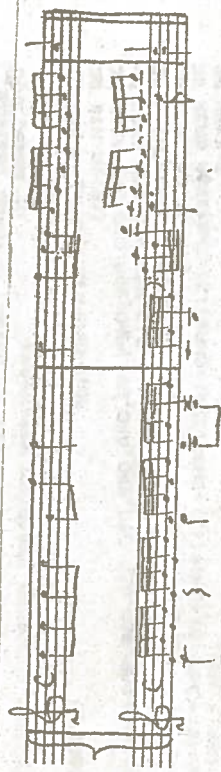
7. Ф. Ш. Шуберт. Соната для фортепиано, первая часть.



8. Ф. Ш. Шуберт. Соната для фортепиано, третья часть.



9. Ф. Ш. Шуберт. Соната для фортепиано, вторая часть.



2.9. Композиция музыки компьютером

ИСК-Бейсик обладает широким спектром музыкальных возможностей. Это позволяет не только исполнять с помощью компьютера музыкальные произведения, написанные человеком (т.е. использовать компьютер как музыкальный инструмент), но и поручить ему функцию "сочинения" музыки с помощью программы, составленной программистом. И эту задачу компьютер успешно решает. Однако компьютерная музыка принципиально отличается от музыки, создаваемой человеком. В компьютерной программе, сочиняющей музыку, удается смоделировать лишь отдельные элементы музыкального творческого процесса, но не весь процесс. Поэтому компьютерная музыка унаследована, лишена вдохновения, творческой фантазии, она безлична. Музыка, создаваемая компьютером, является результатом эстетического освоения автором действительности. Авторская музыка оригинальна и неповторима. Художественную ценность имеет не повторение известного, как в компьютерной музыке, а открытие нового, как в подлинном искусстве.

В нижеприводимой программе 65 музыкальный фрагмент создается на основе девяти нот, выбираемых компьютером случайным образом из исходного списка, содержащего 24 ноты (эти ноты не имеют диезов и бемолей, хотя она и могли бы быть включенными в исходный базовый список). Основную роль при построении музыкального фрагмента играют две функции: FNF(X,K) для построения музыкального фрагмента и функции ритма. Носителем построенного музыкального фрагмента является строковая переменная F\$. После завершения построения функции F\$ программа выводит код построенной мелодии на экран и исполняет ее.

Программа 65

```

10 REM МУЗЫКАЛЬНЫЙ ГЕНЕРАТОР
20 DIM A$(21)
30 DATA 04C, 04D, 04E, 04F, 04G, 04A, 04B, 05C, 05D, 05E, 05F, 05G
40 DATA 05B, 05A, 05B, 06C, 06D, 06E, 06F, 06G, 06A, 06B
50 N=INT(RND(-TIME)*13)
60 FOR I=0 TO N+8
70 READ A$(I)
80 NEXT I
90 DEF FNA(X)=X^2+8*X+6
100 DEF FNB(X)=4*X^2+4.25*X+.25*(X+1)*(X-1)^X+2.25
110 DEF FNC(X)=1.5*(X^2+5*X)+7
120 DEF FND(X)=5*X*(X^2+5*X)+1
130 DEF FNE(X)=1-K*(3-K)
140 DEF FNG(X)=1-2*INT(K/2)
150 DEF FNH(K)=(-1)^K
160 DEF FNP(X,K)=1.5+(-1)^K*(X-1.5)
170 DEF FNR(X,K)=1+2*INT(X/3)+INT((X+K)/6)
180 DEF FNF(X,K)=FNA(FNP(X,K))+FNE(K)+FNB(FNR(X,K))+
FNG(K)+FNC(FNP(X,K))+FNH(K)+FND(FNP(X,K))
190 F$=""
200 FOR K=0 TO 3
210 FOR X=0 TO 3
220 I=FNF(X,K)*MID$
230 I=FNR(X,K)
240 IF I<0 THEN I=I+9
250 I=I+N
260 FOR J=1 TO D
270 F$=F$+A$(I)
280 NEXT J
290 NEXT X,K
300 PRINT "N=";N : PRINT "F$=";F$
310 PLAY F$

```

2.10. Обучающая программа по музыке

Приводимая ниже программа 66 предназначена для обучения принципам кодирования музыкальных нотаций на МММ и составления программ исполнения музыкальных текстов. Содержание учебного материала разделено на три раздела, перечисленных в выходящем на экран меню. Каждый из этих разделов может изучаться независимо от других. Программа приводится в упрощенном варианте, не содержащем сведений о специальных звуковых эффектах, а также раздела, контролирующего усвоение учебного материала.

Программа 66

```

5 REM ОБУЧАЮЩАЯ ПРОГРАММА ПО МУЗЫКЕ
10 SCREEN 7:2
20 LINE(45,15)-(455,40):6.BF
30 SPRITE$(0)=CHR$(8)+CHR$(12)+CHR$(254)+CHR$(12)+CHR$(8)
40 OPEN "GRP:" AS#1
50 COLOR 15,4:DRAW"BM47.17"
60 PRINT#1;" ДЛЯ ВЫБОРА РАЗДЕЛА РАБОТЫ ПОДВЕДИТЕ УКАЗАТЕЛЬ "
70 DRAW"BM32.30"
80 PRINT#1;"К НОМЕРУ РАЗДЕЛА И НАЖМИТЕ НА КЛАВИШУ ПРОБЕЛА"
90 FOR I=1 TO 5
100 LINE(40,30+I*20)-(70,40+I*20):10.BF:NEXT I
110 PBET(100,52)
120 PRINT#1;"ХАРАКТЕРИСТИКА МУЗЫКАЛЬНЫХ ВОЗМОЖНОСТЕЙ MSX-BEASИКА"
130 PBET(100,72)
140 PRINT#1;"ПРИНЦИПЫ КОДИРОВАНИЯ МУЗЫКАЛЬНОЙ НОТАЦИИ"
150 PBET(100,92):PRINT#1;"ПРОГРАМИРОВАНИЕ ОДНОГОЛОСИЯ"
160 PBET(100,112):PRINT#1;"ПРОГРАМИРОВАНИЕ МНОГОГОЛОСИЯ"
170 PBET(100,132):PRINT#1;"ВЫХОД ИЗ РЕЖИМА РАБОТЫ"
180 A=132
190 PUT SPRITE 0,(25,A),1,0:FOR I=1 TO 30:NEXT I
200 IF STICK(0)=1 AND A<=70 THEN A=A-20:GOTO 190
210 IF STICK(0)=5 AND A<160 THEN A=A+20:GOTO 190
220 IF STRIS(0)=0 THEN 190
230 IF A=192 GOTO 200
240 SET PAGE 0,1:CLS
250 LINE ( 1,30)-(510,30):,B
260 LINE(1,37)-(510,37):,B
270 LINE(1,44)-(510,44):,B
280 LINE(1,51)-(510,51):,B
290 LINE(1,58)-(510,58):,B
300 CIRCLE(30,51),20,,3,14,0
310 CIRCLE(35,51),15,,0,3,14
320 LINE (10,51)-(40,20)
330 CIRCLE(35,20),5,,0,3,14
340 LINE(30,20)-(35,70)
350 CIRCLE(30,70),5,,3,14,0
360 LINE (50,190)-(440,210):6.BF

```

```

370 DRAW"BM40.195"
380 PRINT#1."ДЛЯ ПРОДОЛЖЕНИЯ РАБОТЫ НАЖМИТЕ НА КЛАВИШУ ПРОБЕЛА"
390 SET PAGE 0:CLS
400 FOR I=1 TO 15: W0=INKEY$: NEXT I
410 ON (I+(A-S)/20) GOTO 430,940,2430,3890,420
420 SCREEN 0: CLS: STOP
430 CLS:SCREEN 7
440 COLOR: DRAW"BM65.5"
450 PRINT#1."КРАТКАЯ ХАРАКТЕРИСТИКА МУЗЫКАЛЬНЫХ ВОЗМОЖНОСТЕЙ"
460 DRAW"BM180.15":PRINT#1."МХ-БЕЙСИКА"
470 COLOR: DRAW"BM15.30":PRINT#1."МХ-БЕЙСИК ОБЛАДАЕТ "
480 PRINT#1."ДОСТАТОЧНО ШИРОКИМ СПЕКТРОМ ЗВУКОВЫХ ВОЗ-"
490 DRAW"BM2.40":PRINT#1."МОЖЕСТЕЙ. ЗВУКОВЫЕ"
500 PRINT#1."СИГНАЛЫ ВЫДАЮТСЯ ПРОГРАММИРУЕМЫМ ЗВУКОВЫМ"
510 DRAW"BM2.50":PRINT#1."ГЕНЕРАТОРОМ (ПЭГ)."
520 PRINT#1."В ДИАПАЗОНЕ ВОСЬМИ ОКТАВ С ИСПОЛЬЗОВАНИЕМ"
530 DRAW"BM2.60":PRINT#1."ТРЕХ НЕЗАВИСИМЫХ ЗВУКОВЫХ КАНАЛОВ ."
540 DRAW"BM10.70":PRINT#1."ОСНОВНЫМ МУЗЫКАЛЬНО-"
550 PRINT#1."ОРИЕНТИРОВАННЫМ ЭЛЕМЕНТОМ БЕЙСИКА ЯВЛЯЕТСЯ"
560 DRAW"BM2.80":PRINT#1."КОМАНДА PLAY (ИГРАТЬ). ОНА "
570 PRINT#1."ИСПОЛЬЗУЕТ МУЗЫКАЛЬНЫЙ МАКРОЯЗЫК ."
580 DRAW"BM2.90":PRINT#1."(МНЯ). ПОВЕЛИВАЮЩИЙ ОПИСАТЬ "
590 PRINT#1."ТАКИЕ ВАЖНЫЕ ХАРАКТЕРИСТИКИ НОТНОГО "
600 DRAW"BM2.100":PRINT#1."ТЕКСТА, КАК ВЫСОТА,
610 PRINT#1."ТОНА, ДЛИТЕЛЬНОСТЬ ЗВУЧАНИЯ, ТЕМП, ГРОМКОСТЬ."
620 DRAW"BM15.110":PRINT#1."ФОРМАТ КОМАНД PLAY СЛЕДУЮЩИЙ:"
630 PRINT#1."PLAY A#.B%.C%. ГДЕ A#.B%.C% -
640 DRAW"BM2.120":PRINT#1."СТРОКИ СИМВОЛОВ, ЯВЛЯЮЩИЕСЯ "
650 PRINT#1."ЗВУКОВЫМИ ПАРТИЯМИ ПЕРВОГО, ВТОРОГО"
660 DRAW"BM 2.130":PRINT#1."И ТРЕТЬЕГО ГОЛОСОВ. СТРОКА:"
670 PRINT#1."ЗАДАЮЩАЯ МЕЛОДИЮ, СОСТОИТ ИЗ "
680 DRAW"BM2.140":PRINT#1."ОДНОБУКВЕННЫХ КОМАНД И УТОЧНЯЮЩИХ "
690 PRINT#1."ИХ ЧИСЕЛ.":DRAW"BM15.150":
700 PRINT#1."В ПРОСТЕЙШЕМ СЛУЧАЕ ВОЗМОЖНО ИСПОЛЬЗОВАНИЕ В "
710 PRINT#1."КОМАНДЕ PLAY " ; DRAW"BM40.160"
720 PRINT#1."ОДНОГО ГОЛОСА, ТОГДА ОНА ИМЕЕТ ВИД:"
730 COLOR: DRAW"BM80.170":PRINT#1."PLAY <СТРОКОВОЕ ВЫРАЖЕНИЕ> "
740 COPY (50.190)-(460.210).1 TO (50.190).0
750 IF INKEY$="" THEN 760 ELSE 750
760 SCREEN 7: COLOR 15
770 DRAW"BM15.180":PRINT#1."ПРИ ВЫПОЛНЕНИИ КОМАНДЫ PLAY "
780 DRAW"BM235.10"
790 PRINT#1."СТРОКОВЫЕ ВЫРАЖЕНИЯ ОБРАБАТЫВАЮТСЯ"
800 DRAW"BM2.20":PRINT#1."И ПРЕОБРАЗУЮТСЯ "
810 PRINT#1."ВО ВНУТРЕННИЙ МУЗЫКАЛЬНЫЙ ФОРМАТ, ОРГАНИЗОВАННЫЙ"
820 DRAW"BM2.30":PRINT#1."В ТРИ ОЧЕРЕДИ "
830 PRINT#1."(ПО ОДНОУ ДЛЯ КАЖДОГО ГОЛОСА), РАЗМЕЩАЕМЫЕ В ПЭГ."
840 DRAW"BM2.40":PRINT#1."ПЭГ АВТОМАТИЧЕСКИ ВОСПРОИЗВОДИТ "
850 PRINT#1."ПО ОДНОЙ НОТЕ ИЗ КАЖДОЙ ОЧЕРЕДИ"
860 DRAW"BM2.50":PRINT#1."ОДНОВРЕМЕННО ДО ТЕХ ПОР, ПОКА ОЧЕРЕДИ "
870 PRINT#1."НЕ ПЛУСТОВАТСЯ. ПОСЛЕ"
880 DRAW"BM2.60":PRINT#1."ЭТОГО В ОСВОБОДИВШЕМСЯ ПРОСТРАНСТВЕ "
890 PRINT#1."РАЗМЕЩАЮТСЯ ДАННЫЕ СЛЕДУЮ-"
900 DRAW"BM2.70":PRINT#1."ЩЕГО ОПЕРАТОРА PLAY."
910 COPY (50.190)-(460.210).1 TO (50.190).0

```

```

920 IF INKEY$="" THEN 930 ELSE 920
930 CLS: GOTO 50
940 CLS:SCREEN 7
950 COLOR: DRAW"BM30.15":PRINT#1."ПРИНЦИПЫ КОДИРОВАНИЯ "
960 PRINT#1."МУЗЫКАЛЬНОЙ НОТАЦИИ"
970 COLOR: DRAW"BM15.30":PRINT#1."ОБЪЯСНЕНИЯ НОТ. В МНЯ ДЛЯ "
980 PRINT#1."НОТ ОСНОВНОГО МУЗЫКАЛЬНОГО ЗВУКО-"
990 LINE (13.30)-(135.30).1
1000 DRAW"BM5.42":PRINT#1."РЯДА ПРИНЯТЫ СЛЕДУЮЩИЕ БУКВЕННЫЕ "
1010 PRINT#1."ОБЪЯСНЕНИЯ:"
1020 COLOR: DRAW"BM40.52":PRINT#1."C(ДО), D(РЕ), E(МИ), F(ФА). "
1030 PRINT#1."G(СОЛЬ), A(ЛЯ), B(СИ) "
1040 COLOR 15: DRAW"BM15.62":PRINT#1."НОТЫ ОСНОВНОГО "
1050 PRINT#1."МУЗЫКАЛЬНОГО ЗВУКОРЯДА ИСПОЛНЯЮТСЯ КОМПЬЮТЕРОМ"
1060 DRAW"BM2.72":PRINT#1."В ЧЕТВЕРТОЙ ОКТАВЕ. (В ОШЕДЕРЖАТОРАХ"
1070 PRINT#1."МУЗЫКАЛЬНОЙ НОТАЦИИ ОНА"
1080 DRAW"BM2.82":PRINT#1."СООТВЕТСТВУЕТ ПЕРВОЙ ОКТАВЕ). "
1090 PRINT#1."НАД ОСНОВНЫМ ЗВУКОРЯДОМ РАСПОЛОЖЕ-"
1100 DRAW"BM2.92":PRINT#1."НЫ НОТЫ ПЯТОЙ, ШЕСТОЙ, СЕДЬМОЙ И "
1110 PRINT#1."ВОСЬМОЙ ОКТАВ, НИЖЕ- ТРЕТЬЕЙ, "
1120 DRAW"BM2.102":PRINT#1."ВТОРОЙ И ПЕРВОЙ ОКТАВ."
1130 DRAW"BM15.112":PRINT#1."ПРИ ПРОГРАММИРОВАНИИ МУЗЫКАЛЬНЫХ "
1140 PRINT#1."ТЕКСТОВ ОКТАВА УКАЗЫВАЕТСЯ "
1150 DRAW"BM2.122":PRINT#1."БУКВОЙ"
1160 COLOR: DRAW"BM55.122":PRINT#1."D"
1170 COLOR: DRAW"BM67.122":PRINT#1."С ПОСЛЕДУЮЩИМ НОМЕРОМ "
1180 PRINT#1."ЖЕЛАЕМОЙ ОКТАВЫ, ЕСЛИ ОКТАВА НЕ"
1190 DRAW"BM2.132":PRINT#1."УКАЗАНА, НОТА ИСПОЛНЯЕТСЯ В "
1200 PRINT#1."ЧЕТВЕРТОЙ ОКТАВЕ. КОМАНДА"
1210 COLOR: DRAW"BM417.132":PRINT#1."O"
1220 DRAW"BM2.142":PRINT#1."ГРУППОВОЙ (ВЛИКАЕТ НА ВСЕ "
1230 PRINT#1."ПОСЛЕДУЮЩИЕ НОТЫ, ПОКА НЕ БУДЕТ "
1240 DRAW"BM2.152":PRINT#1."ПЕРЕОПРЕДЕЛЕНА). НАПРИМЕР, КОМАНДА"
1250 COLOR: DRAW"BM100.162":PRINT#1."PLAY 'O3DEFGBA04CDEFGAB'"
1260 PRINT#1."И ЧЕТВЕРТОЙ ОКТАВ, ДЛЯ ИСПОЛНЕНИЯ "DURAN"BM2.182"
1270 IF INKEY$="" THEN 1310 ELSE 1300
1300 PLAY "O3CDEFGBA04CDEFGAB"
1310 FOR I=1 TO 3000: NEXT I:CLS
1320 CLS:SCREEN 7
1330 FOR B=1 TO 10
1340 DRAW"BM100.15":PRINT#1."ВНИМАНИЕ! НАЖМИТЕ КЛАВИШУ CARB!"
1350 NEXT B
1360 FOR I=65 TO 400 STEP 50
1370 LINE (I,40)-(I+50,100).4,B
1380 NEXT I
1390 FOR I=1 TO 10: NEXT I
1400 FOR I=65 TO 400 STEP 50
1410 LINE (I,40)-(I+50,100).4,B
1420 NEXT I
1430 FOR I=1 TO 10: NEXT I
1440 FOR T=100 TO 350 STEP 50

```

```

2010 DRAW"BM20,150":PRINT#1,"В РАЗНЫХ ОКТАВАХ"
2020 DRAW"BM20,160":PRINT#1,"НАПРИМЕР: 04С*,05F*,02G* И ";
2030 PRINT#1,"НАПРИМЕР: Т.Д. ВВЕДИТЕ КОД ОКТАВЫ."
2040 DRAW"BM20,175":PRINT#1,"НОТЫ И ЗНАКА АЛЬТЕРАЦИИ:"
2050 LINE (250,175)-(320,183),0,BF
2060 GOSUB 1730
2070 DRAW"BM20,190":PRINT#1,"ДЛЯ ПРОДОЛЖЕНИЯ ПРОСЛУШИВАНИЯ ";
2080 PRINT#1,"НАЖМИТЕ КЛАВИШУ 'У', Б"
2090 DRAW"BM20,200":PRINT#1,"ПРОТИВНОМ СЛУЧАЕ - КЛАВИШУ 'N'"
2100 W$=INKEY$
2110 IF W$="" THEN 2100
2120 IF W$="Y" THEN 2050
2130 LINE (0,120)-(512,210),3,BF
2140 DRAW"BM3,130":PRINT#1,"ДЛЯ ИЗМЕНЕНИЯ ДЛИТЕЛЬНОСТИ НОТ В ";
2150 PRINT#1,"МНЯ ИСПОЛЬЗУЕТСЯ ПАРАМЕТР L,"
2160 DRAW"BM3,140":PRINT#1,"ЗА КОТОРЫМ СЛЕДУЕТ ЧИСЛО ОТ 1 ДО ";
2170 PRINT#1,"64. НАПРИМЕР: 03L4C. ТОЧКА,"
2180 DRAW"BM3,150":PRINT#1,"ПОСТАВЛЕННАЯ РЯДОМ С НОТОЙ. ";
2190 PRINT#1,"УВЕЛИЧИВАЕТ ДЛИТЕЛЬНОСТЬ ЕЕ ЗВУ-"
2200 DRAW"BM2,160":PRINT#1,"ЧАНИЯ НА ЕЕ ПОЛОВИНУ. ПАУЗА ";
2210 PRINT#1,"КОДИРУЕТСЯ БУКВОЙ 'С ПОСЛЕДУЮЩИМ"
2220 DRAW"BM3,170":PRINT#1,"ЗА НЕЙ ЧИСЛОМ ОТ 1 ДО 64."
2230 COPY (50,190)-(460,210),1 TO (50,190),0
2240 IF INKEY$="" THEN 2250 ELSE 2240
2250 LINE (0,120)-(512,210),3,BF
2260 DRAW"BM3,140":PRINT#1,"ПРОИГРАЙТЕ НОТЫ ОСНОВНОГО ";
2270 PRINT#1,"ЗВУКОРЯДА С РАЗЛИЧНЫМИ ДЛИТЕЛЬНОСТЯМИ"
2280 DRAW"BM20,150":PRINT#1,"НАПРИМЕР: С2,С5,03D*,07G*
2290 DRAW"BM40,175":PRINT#1,"ВВЕДИТЕ ЖЕЛАЕМЫЙ КОД:"
2300 LINE (250,175)-(400,183),3,BF
2310 GOSUB 1730
2320 DRAW"BM20,190":PRINT#1,"ДЛЯ ПРОДОЛЖЕНИЯ ПРОСЛУШИВАНИЯ ";
2330 PRINT#1,"НАЖМИТЕ КЛАВИШУ 'У'"
2340 DRAW"BM20,200":PRINT#1,"В ПРОТИВНОМ СЛУЧАЕ - КЛАВИШУ 'N'"
2350 FOR I=1 TO 10:INKEY$;NEXT
2360 W$=INKEY$
2370 IF W$="" THEN 2360
2380 IF W$="Y" THEN 2300
2390 CLS: GOTO 50
2400 COPY (50,190)-(460,210),1 TO (50,190),0
2410 IF INKEY$="" THEN 2420 ELSE 2420
2420 FOR I=1 TO 100:NEXT I:CLS:GOTO 50
2430 SCREEN 7
2440 DRAW"BM100,10":PRINT#1,"ПРОГРАММИРОВАНИЕ ОДНОГОЛОСЯ"
2450 DRAW"BM20,30":PRINT#1,"ПРИ ПРОГРАММИРОВАНИИ НОТНЫХ ";
2460 PRINT#1,"ЗАПИСЕЙ НЕОБХОДИМО ОПРЕДЕЛИТЬ КО-"
2470 DRAW"BM1,40":PRINT#1,"ДЫ НОТ, НОМЕРА ОКТАВ, В КОТОРЫХ ОНИ ";
2480 PRINT#1,"ДОЛЖНЫ ИСПОЛНЯТЬСЯ, ДЛИТЕЛЬНО-"
2490 DRAW"BM1,50":PRINT#1,"НОСТИ НОТ, НАЛИЧИЕ ЗНАКОВ ";
2500 PRINT#1,"АЛЬТЕРАЦИИ, ДАЛЕЕ ИЗ ЭТИХ КОДОВ МОГУТ ";
2510 DRAW"BM1,60":PRINT#1,"БЫТЬ СФОРМИРОВАНЫ СТРОКОВЫЕ ";
2520 PRINT#1,"ВЫРАЖЕНИЯ, КОТОРЫЕ ИСПОЛНЯЮТСЯ"
2530 DRAW"BM1,70":PRINT#1,"С ПОМОЩЬЮ ОПЕРАТОРА PLAY."
2540 DRAW"BM20,80":PRINT#1,"РАССМОТРИМ, НАПРИМЕР, ";
2550 PRINT#1,"НИЖЕСЛЕДУЮЩИЙ НОТНЫЙ ФРАГМЕНТ:"
2560 COPY (50,190)-(460,210),1 TO (50,190),0

```

```

1450 IF T=200 THEN 1470
1460 LINE (T,40)-(T+30,80),0,BF
1470 NEXT T
1480 RESTORE 1490
1490 DATA 93,90,C,133,90,D,183,90,E,233,90,F,283,90,G,333
1500 DATA 90,A,383,90,B
1510 FOR J=1 TO 7
1520 READ X,Y,W$
1530 PSET (X,Y)
1540 PRINT#1,W$
1550 NEXT J
1560 DRAW"BM3,140":PRINT#1,"ПРОИГРАЙТЕ НОТЫ ОСНОВНОГО ЗВУКОРЯДА ";
1570 PRINT#1,"В РАЗНЫХ ОКТАВАХ, ВВОДЯ ИХ С"
1580 DRAW"BM2,150":PRINT#1,"КЛАВИАТУРЫ И ЗАВЕРШАЯ ВВОД НАЖАТИЕМ. ";
1590 PRINT#1,"КЛАВИШУ ВОЗВРАТА КАРЕТКИ."
1600 DRAW"BM2,160":PRINT#1,"НАПРИМЕР: 01C,03C,05D 07F И Т.Д."
1610 DRAW"BM20,175":PRINT#1,"ВВЕДИТЕ КОД ОКТАВЫ И НОТЫ:"
1620 LINE (250,175)-(320,183),3,BF
1630 GOSUB 1730
1640 DRAW"BM3,190":PRINT#1,"ДЛЯ ПРОСЛУШИВАНИЯ СЛЕДУЮЩЕЙ НОТЫ ";
1650 PRINT#1,"НАЖМИТЕ КЛАВИШУ 'У', ДЛЯ"
1660 DRAW"BM3,200":PRINT#1,"ПРОДОЛЖЕНИЯ РАССМОТРЕНИЯ ПРИНЦИПОВ ";
1670 PRINT#1,"КОДИРОВАНИЯ - КЛАВИШУ 'N'"
1680 W$=INKEY$
1690 IF W$="" THEN 1680
1700 IF W$="Y" THEN 1610
1710 COLOR 15,4,4
1720 GOTO 1890
1730 A$=""
1740 PRESET (260,175)
1750 B$=INPUT$(1)
1760 IF B$=CHR$(8) THEN 1810
1770 IF B$=CHR$(13) THEN 1860
1780 PRINT#1,B$
1790 A$=A$+B$
1800 GOTO 1750
1810 K=LEN(A$)-1:IF K<0 THEN 1730
1820 A$=LEFT$(A$,K)+" "
1830 PRESET(260,175):PRINT #1,A$;A$=LEFT$(A$,K)
1840 PRESET STEP(-8,0)
1850 GOTO 1750
1860 PLAY A$
1870 FOR I=1 TO 400:NEXT I
1880 RETURN
1890 LINE (0,130)-(512,210),0,BF
1900 DRAW"BM3,140":PRINT#1,"АЛЬТЕРАЦИЯ СТУПЕНЕЙ ОСНОВНОГО ";
1910 PRINT#1,"ЗВУКОРЯДА ОСУЩЕСТВЛЯЕТСЯ С"
1920 DRAW"BM3,150":PRINT#1,"ПОМОЩЬЮ ЗНАКОВ БЕМОЛЯ (b " ЗНАК ";
1930 PRINT#1,"ПОНИЖЕНИЯ, КОДИРУЕТСЯ '-' ) И"
1940 DRAW"BM3,160":PRINT#1,"ДИЕЗА (# - ЗНАК ПОВЫШЕНИЯ, ";
1950 PRINT#1,"КОДИРУЕТСЯ '+') "
1960 COPY (50,190)-(460,210),1 TO (50,190),0
1970 IF INKEY$="" THEN 1980 ELSE 1970
1980 LINE (0,130)-(512,210),3,BF
1990 DRAW"BM20,140":PRINT#1,"ПРОИГРАЙТЕ НОТЫ ОСНОВНОГО ";
2000 PRINT#1,"ЗВУКОРЯДА СО ЗНАКАМИ АЛЬТЕРАЦИИ"

```

```

2570 IF INKEY#="" THEN 2580 ELSE 2570
2580 CLS
2590 COPY (0,5)-(511,60),1 TO (0,5),0
2600 LINE(66,33)-(60,39)
2610 LINE(55,39)-(55,50)
2620 LINE(60,28)-(60,39)
2630 CIRCLE(58,44),3,0,3,14
2640 LINE(61,44)-(55,50)
2650 CIRCLE(80,68),5:PAINT(80,68):PAINT(80,68)
2660 LINE(86,44)-(86,65),,B
2670 LINE(70,65)-(90,65),,B
2680 CIRCLE(76,53),20,0,3,14/3
2690 CIRCLE(110,56),5:PAINT(110,56):PAINT(110,60)
2700 LINE(116,55)-(116,34),,B
2710 CIRCLE(110,40),5,3,14,2X3,14
2720 LINE(140,40)-(140,48)
2730 CIRCLE(170,50),5:PAINT(170,56):PAINT(170,60)
2740 LINE(176,55)-(176,34),,B
2750 CIRCLE(167,43),20,,0,3,14/3
2760 LINE(200,30)-(200,58),,B
2770 CIRCLE(215,52),5
2780 LINE(230,30)-(230,58),,B
2790 CIRCLE(250,62),5:PAINT(250,62)
2800 LINE(256,59)-(256,38),,B
2810 LINE(270,50)-(270,59),,B
2820 LINE(265,50)-(265,59),,B
2830 LINE(262,53)-(273,53)
2840 LINE(262,55)-(273,55)
2850 CIRCLE(280,55),5:PAINT(278,55)
2860 LINE(286,53)-(286,32)
2870 LINE(286,32)-(286,38)
2880 CIRCLE(308,55),5:PAINT(308,55)
2890 LINE(315,54)-(315,33)
2900 CIRCLE(336,51),5:PAINT(336,53):PAINT(336,50)
2910 LINE(343,49)-(343,26)
2920 LINE(315,33)-(344,26)
2930 CIRCLE(364,48),5:PAINT(364,48)
2940 LINE(359,51)-(359,72)
2950 CIRCLE(392,34),5:PAINT(392,34)
2960 LINE(387,37)-(387,58)
2970 LINE(387,58)-(359,72)
2980 CIRCLE(420,34),5:PAINT(420,34)
2990 LINE(415,37)-(415,58)
3000 CIRCLE(448,37),5:PAINT(448,35):PAINT(448,39)
3010 LINE(443,40)-(443,61)
3020 LINE(443,61)-(415,58)
3030 LINE(468,30)-(468,58)
3040 LINE(510,30)-(510,58)
3050 CIRCLE(488,44),5
3060 DRAW"BM20,100":PRINT#1,"РАСМАТРИВАЕМЫЙ ФРАГМЕНТ МОЖЕТ "
3070 PRINT#1,"БЫТЬ РЕАЛИЗОВАН КОМАНДОЙ "
3080 DRAW"BM50,120":PRINT#1,"PLAY"USE-804E-4RBE-8D1DBF+8F+8":
3090 PRINT#1,"GRABOSE-8E-HD9048-1"
3100 DRAW"BM50,140":PRINT#1,"ИСПИШИТЕ ЭТУ КИМАНДУ, НАЖАВ "
3110 PRINT#1,"НА КЛАВИШУ ПРИБЕРА."

```

```

3120 IF INKEY#="" THEN 3130 ELSE 3120
3130 PLAY "03B-804E-4RBE-8D1DBF+8F+8B8A805E-8E-8D8048-1"
3140 COPY (50,190)-(460,210),1 TO (50,190),0
3150 IF INKEY#="" THEN 3160 ELSE 3150
3160 CLS
3170 DRAW"BM20,5":PRINT#1,"ПРОГРАММИРОВАНИЕ МУЗЫКАЛЬНОГО "
3180 PRINT#1,"ТЕКСТА С ИСПОЛЗОВАНИЕМ "
3190 DRAW"BM1,15":PRINT#1,"СТРОКОВЫХ ВЫРАЖЕНИЯ НЕ ПОЗВОЛЯЕТ "
3200 PRINT#1,"УЧЕТЬ ИМЕЮЩЕЯ В ТЕКСТЕ "
3210 DRAW"BM1,25":PRINT#1,"ЗНАКИ ЛИРИИ, КРОМЕ ТОГО, ОНО МОЖЕТ "
3220 PRINT#1,"ПОВЛЕЧЬ ЗА СОБОЙ ИСКАЖЕНИЯ "
3230 DRAW"BM1,35":PRINT#1,"В ВОСПРОИЗВЕДЕНИИ МЕЛОДИИ ТАМ, ГДЕ "
3240 PRINT#1,"ВСТРЕЧАЮТСЯ ДВЕ ИЛИ БОЛЕЕ "
3250 DRAW"BM1,45":PRINT#1,"ОДИНАКОВЫХ ПО ВЫСОТЕ НОТ, ДЛЯ "
3260 PRINT#1,"ИЗБЕЖАНИЯ ЭТИХ ЯВЛЕНИЙ КОДЫ "
3270 DRAW"BM1,55":PRINT#1,"НОТ СБИРАЮТ В БЛОК "
3280 PRINT#1,"И СОСТАВЛЯЮТ ЦИКЛИЧЕСКИЙ "
3290 PRINT#1,"ВАРИАНТ ПРОГРАММЫ, ОСУЩЕСТВЛЯЮЩАЯ "
3300 PRINT#1,"СЧИТЫВАНИЕ КОДОВ НОТ ИЗ "
3310 DRAW"BM1,75":PRINT#1,"БЛОКА ДАННЫХ И ИХ ВОСПРОИЗВЕДЕНИЕ. "
3320 PRINT#1,"ЭТО РАССМАТРИВАЕМОГО МУЗЫКА "
3330 DRAW"BM1,85":PRINT#1,"ЛЬНОГО ФРАГМЕНТА ТАКАЯ ПРОГРАММА "
3340 PRINT#1,"БУДЕТ СЛЕДУЮЩЕЙ "
3350 DRAW"BM10,100":PRINT#1,"10
3360 DRAW"BM10,110":PRINT#1,"20 DATA 03B-8,04E-4,RB,E-8,D2,DB,F+8
3370 DRAW"BM10,120":PRINT#1,"30 FOR I=1 TO 14"
3380 DRAW"BM10,130":PRINT#1,"40 READ A$:PLAY A$"
3390 DRAW"BM10,140":PRINT#1,"50 NEXT I"
3400 DRAW"BM10,160":PRINT#1,"ДЛЯ ИСПОЛНЕНИЯ ПРОГРАММЫ НАЖМИТЕ НА "
3410 PRINT#1,"КЛАВИШУ ПРИБЕРА."
3420 IF INKEY#="" THEN 3430 ELSE 3420
3430 RESTORE 3440
3440 DATA 03B-8,04E-4,RB,E-8,D2,DB,F+8,AB,05E-8,E-8,DB,04B-1
3450 FOR I=1 TO 14
3460 READ A$:PLAY A$
3470 NEXT I
3480 COPY (50,190)-(460,210),1 TO (50,190),0
3490 IF INKEY#="" THEN 3500 ELSE 3490
3500 CLS
3510 DRAW"BM20,5":PRINT#1,"ТЕМП МУЗЫКИ МОЖНО ИЗМЕНИТЬ С ПОМОЩЬЮ "
3520 PRINT#1,"ПАРАМЕТРА T, ЗА КОТОРЫМ "
3530 DRAW"BM1,15":PRINT#1,"СЛЕДУЕТ ЧИСЛО ОТ 32 ДО 255. ЭТО "
3540 PRINT#1,"ЧИСЛО РАВНО ЧИСЛУ ЧЕТВЕРТЕЙ, "
3550 DRAW"BM1,25":PRINT#1,"ЗВУЧАЩИХ В ТЕЧЕНИЕ МИНУТЫ. ПО "
3560 PRINT#1,"УМОЛЧАНИЮ ПРИНИМАЕТСЯ Т120. ЭТО "
3570 DRAW"BM1,35":PRINT#1,"ОЗНАЧАЕТ, ЧТО ЗА МИНУТУ ИСПОЛНЯЕТСЯ "
3580 DRAW"BM1,45":PRINT#1,"ЦЕЛЫХ НОТ, ЧЕМ БОЛЬШЕ T, ТЕМ "
3590 PRINT#1,"БЫСТРЕЕ ТЕМП."
3610 DRAW"BM20,55":PRINT#1,"ГРОМКОСТЬ ЗВУЧАНИЯ РЕГУЛИРУЕТСЯ "
3620 PRINT#1,"ПАРАМЕТРОМ V, ЗА КОТОРЫМ "
3630 DRAW"BM1,65":PRINT#1,"СЛЕДУЕТ ЧИСЛО ОТ 0 ДО 15 (V0 - "
3640 PRINT#1,"ОЧЕНЬ ТИХО, V15 - ГРОМКО. "
3650 DRAW"BM1,75":PRINT#1,"ДО ХРИПТЫ). ПО УМОЛЧАНИЮ "
3660 PRINT#1,"ПРИНИМАЕТСЯ V8."
3670 DRAW"BM1,85":PRINT#1,"ВКЛЮЧА В РАБЕЕ РАССМОТРЕННУЮ "
3680 PRINT#1,"ПРОГРАММУ РЕГУЛИРОВАНИЕ ТЕМПА"

```

```

3690 DRAW"BM1,95":PRINT#1,"И ГРОМКОСТИ КОМАНДОЙ PLAY 'T150V10'";
3700 PRINT#1," ПОЛУЧИМ ПРОГРАММУ.";
3710 DRAW"BM10,110":PRINT#1,"10 DATA 03B-B,04E-4,RB,E-B,02,08,F+8";
3720 DRAW"BM10,120":PRINT#1,"20 DATA F+8,5B,AB,05E-B,E-B,08,04B-1";
3730 DRAW"BM10,130":PRINT#1,"30 PLAY 'T130V10'";
3740 DRAW"BM10,140":PRINT#1,"40 FOR I=1 TO 14";
3750 DRAW"BM10,150":PRINT#1,"50 READ A$:PLAY A$";
3760 DRAW"BM10,160":PRINT#1,"60 NEXT I";
3770 DRAW"BM20,175":PRINT#1,"ИСТОЯНИТЕ ПРОГРАММУ, НАЖИМАЯ НА ";
3780 PRINT#1,"КЛАВИШУ ПРОБЕЛА";
3790 IF INKEY$="" THEN 3800 ELSE 3790
3800 RESTORE 3810
3810 DATA 03B-B,04E-4,RB,E-B,02,08,F+8,F+8,AB,05E-B,E-B,08,04B-1
3820 PLAY 'T130V10'
3830 FOR I=1 TO 14
3840 READ A$: PLAY A$
3850 NEXT I
3860 COPY (50,190)-(440,210),1 TO (50,190),0
3870 IF INKEY$="" THEN 3880 ELSE 3870
3880 CLS: GOTO 50
3890 CLS: SCREEN 7
3900 DRAW"BM50,10":PRINT#1,"ПРОГРАММИРОВАНИЕ МНОГОГОЛОСНОЙ";
3910 DRAW"BM10,30":PRINT#1,"ПРИ ПРОГРАММИРОВАНИИ ДВУХГОЛОСНЫХ ";
3920 PRINT#1,"И ТРЕХГОЛОСНЫХ МУЗЫКАЛЬНЫХ";
3930 DRAW"BM1,40":PRINT#1,"ТЕКСТОВ С ЦЕЛЬЮ ОБЕСПЕЧЕНИЯ ";
3940 PRINT#1,"СМЯКРОУЩАГО ЗВУЧАНИЯ ГОЛОСОВ";
3950 DRAW"BM1,50":PRINT#1,"НЕОБХОДИМО СОЗДАВАТЬ ПРЯЖИЛЫ ";
3960 PRINT#1,"ОДИНАКОВОЙ ДЛИТЕЛЬНОСТИ НОТ, ПРО-";
3970 DRAW"BM1,60":PRINT#1,"ИЗЫВАЮЩИХ ОДНОВРЕМЕННО ЛЮ ";
3980 PRINT#1,"ПАРАЛЛЕЛЬНЫЙ КАНАЛАМ.";
3990 DRAW"BM20,70":PRI.T#1,"В КАЧЕСТВЕ ИЛЛЮСТРАЦИОННОГО ";
4000 PRINT#1,"ПРИМЕРА ПРИВЕДЕМ НИЖЕСЛЕДУЮЩИЙ";
4010 DRAW"BM1,80":PRINT#1,"ДВУХГОЛОСНЫЙ ФРАГМЕНТ.";
4020 COPY (50,190)-(460,210),1 TO '30,190',0
4030 IF INKEY$="" THEN 4040 ELSE 4030
4040 CLS: COPY (0,5)-(511,80),1 TO(0,5),0
4050 PUT SPRITE0,(25,112),0,0
4060 CIRCLE (30,58),5:PAINT (80,56):PAINT (80,60)
4070 LINE(55,37)-(85,36):LINE(75,59)-(75,60)
4080 CIRCLE (110,48),5:PAINT(110,50)
4090 LINE(115,47)-(115,24)
4100 CIRCLE(110,66),5:PAINT(110,57)
4110 LINE(100,65)-(120,65)
4120 LINE (105,67)-(105,88)
4130 CIRCLE (140,41),5:PAINT(140,42):PAINT(140,46)
4140 LINE(145,43)-(145,22)
4150 CIRCLE (140,58),5:PAINT(140,60):PAINT(140,56)
4160 LINE (135,59)-(135,80)
4170 LINE(160,30)-(160,59)
4180 CIRCLE(180,41),5:PAINT(180,42)
4190 LINE (185,42)-(185,21)
4200 CIRCLE(180,48),5:PAINT(180,46)
4210 LINE(175,49)-(175,70)
4220 CIRCLE(210,44),5:PAINT(210,42):PAINT(210,46)
4230 LINE(215,45)-(215,23)
4240 CIRCLE(210,58),5:PAINT(210,56):PAINT(210,60)

```

```

4250 LINE(205,59)-(205,60)
4260 CIRCLE(240,48),5:PAINT(240,49)
4270 LINE(245,47)-(245,26)
4280 CIRCLE(240,66),5:PAINT(240,67)
4290 LINE (230,66)-(250,66)
4300 LINE(260,30)-(260,58)
4310 LINE(235,67)-(235,88)
4320 CIRCLE(280,44),5
4330 LINE(285,43)-(285,22)
4340 CIRCLE (280,62),5
4350 LINE(275,63)-(275,84)
4360 CIRCLE(310,55),5:PAINT(310,54)
4370 LINE(315,54)-(315,33)
4380 CIRCLE(310,62),5:PAINT(310,62)
4390 LINE(330,30)-(330,58)
4400 LINE(305,63)-(305,84)
4410 CIRCLE(350,58),5
4420 LINE(355,54)-(355,33)
4430 CIRCLE(350,62),5
4440 LINE(345,63)-(345,84)
4450 LINE(370,30)-(370,58)
4460 CIRCLE(360,55),1:CIRCLE(360,62),1
4470 LINE(52,32)-(52,41)
4480 LINE(57,32)-(57,41)
4490 LINE(49,35)-(60,35)
4500 LINE(49,39)-(60,39)
4510 LINE(70,36)-(70,25):LINE(75,36)-(75,25)
4520 LINE(67,28)-(78,28):LINE(67,32)-(78,32)
4530 LINE(73,36)-(73,47):LINE(68,36)-(68,47)
4540 LINE(65,39)-(76,39):LINE(65,42)-(76,42)
4550 DRAW"BM20,90":PRINT#1,"СТРУКТУРОВАВ КОДЫ НОТ В БЛОК ";
4560 PRINT#1,"ДАННЫХ, ПОЛУЧИМ НИЖЕСЛЕДУЮЩИЙ";
4570 DRAW"BM1,100":PRINT#1,"ПРОГРАММУ ИСПОЛНЕНИЯ ";
4580 PRINT#1,"МУЗЫКАЛЬНОГО ФРАГМЕНТА";
4590 DRAW"BM20,120":PRINT#1,"5 REM ШКОЛЬНЫЕ ГОДЫ";
4600 DRAW"BM20,130":PRINT#1,"10 DATA E4,E4,E4,A4,C4,A,B4,E4";
4610 PRINT#1,"OSC+4,A4,04B4,E4,A4,C+4";
4620 DRAW"BM20,140":PRINT#1,"20 DATA B2,D2,F+4,D4,F+2F+4,D2D4";
4630 DRAW"BM20,150":PRINT#1,"70 PLAY'V10T200'";
4640 DRAW"BM20,160":PRINT#1,"80 FOR I=1 TO 9";
4650 DRAW"BM20,170":PRINT#1,"90 READ A$:B$: PLAY A$,B$";
4660 DRAW"BM20,180":PRINT#1,"100 NEXT I";
4670 RESTORE 4680
4680 DATA 04E4,E4,A4,C+4,B4,E4,05C+4,A4,04B4,E+4,A4,C+4
4690 DATA B2,D2,F+4,D4,F+2F+4,D2D4
4700 PLAY'V:0T200'";
4710 FOR I=1 TO 9
4720 READ A$,B$
4730 PLAY A$,B$
4740 NEXT I
4750 COPY (50,190)-(460,210),1 TO (50,190),0
4760 IF INKEY$="" THEN 4770 ELSE 4760
4770 CLS: GOTO 50

```

3. Учебные компьютерные игры

К числу активных методов обучения, призванных содействовать успешному формированию у учащихся знаний и умений, относятся учебные игры. Их познавательные возможности значительно возросли с появлением в сфере образования персональных компьютеров семейства ИХУ, обеспечивающих обменные с учащимися в диалоговом режиме и обладающих хорошо развитыми графическими и мультимедийными возможностями. Учебная компьютерная игра (УКИ), в которой партнером учащегося является компьютер, соединяет эмпатическую привлекательность, присущую игре, с аудиовизуальными, вычислительными, информационными и другими возможностями компьютера и несет в себе большой дидактический потенциал.

Практика использования УКИ показала, что наибольший эффект они дают при изучении математики, прежде всего при контроле и оценке знаний учащихся. Определенные успехи достигнуты также и при использовании компьютеров в обучении, прежде всего младших школьников. При этом включение в учебную деятельность игровых элементов (персонажей, сказок, мультфильмов и т.д.) способствует усилению мотивации обучения. Управление деятельностью учащихся в рамках такой игры носит координатный характер, а обучение ведется в игровой форме, в которой учащиеся лишь после завершения отдельных этапов (как правило, в общей форме, часто - в имитационной). После завершения игры возможны и прямые указания на допущенные ошибки, а также указания на оптимальные способы реализации стратегии игры.

3.1. Игровая программа "Вычитание чисел"

Программа предназначена для контроля правильности выполнения действия вычитания чисел младшими школьниками. Всего учащемуся предлагается выполнить десять управлений. Уменьшаемое и вычитаемое, представляющие собой двузначные числа, генерируются программой.

После запуска программы в работу на экране монитора появляется заглавный кадр «Игровая программа по теме: "Вычитание чисел"», сменяющийся через некоторое время кадром с формулировкой задания

и игровым фоном (см. рисунок 3.1).

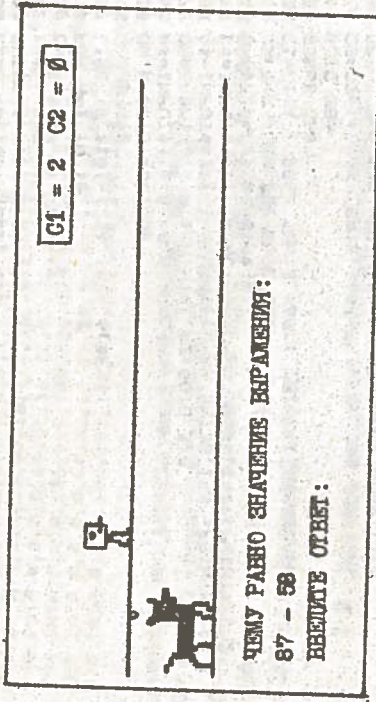


Рис. 3.1

Вводимые учащимся ответы анализируются компьютером с выдачей на экран соответствующих реплик. При правильном ответе на экране появляется подтверждение "ПРАВИЛЬНО", а коллбек переключается по экрану на некоторое расстояние. В случае неправильного ответа на экране появляется "НЕПРАВИЛЬНО", на экран выводится значение правильного ответа, на некоторое расстояние переключается лайна. В правой верхней части экрана выводятся счетчики, на которых индицируется число верных и неверных ответов. После завершения вопроса на экран выдается характеристика успешности выполнения учащимся работы: число правильных ответов и итоговая оценка за выполнение управления.

Программа 67

```

5 REM ВЫЧИТАНИЕ ЧИСЕЛ
10 SCREEN 2,2
20 OPEN "GRP:" AS#1
30 DRAW"BM40,50": PRINT#1,"ИГРОВАЯ ПРОГРАММА"
40 DRAW"BM70,70": PRINT#1,"ПО ТЕМЕ:"
50 DRAW"RM40,90": PRINT#1,"ВЫЧИТАНИЕ ЧИСЕЛ"
60 FOR I=1 TO 400 : NEXT I
70 CLS
80 FOR I=1 TO 4
90 C$=""
100 FOR J=1 TO 32
110 READ A
120 C$=C$+CHR*(A)
130 NEXT J
140 SPRITE$(1)=C$
150 NEXT I
160 LINE (0,46)-(500,46)
    
```

```

170 PUTSPRITE 0, (5,30),1
180 LINE (0,100)-(500,100)
190 PUTSPRITE 1, (5,84),3
200 X=5 : Y=5
210 C=0
220 FOR N=1 TO 10
230 LINE (0,110)-(256,192),6,BF
240 GOSUB 470
250 ON K GOTO 260,280
260 A$=INPUT$(1):A1=0:A2=VAL(A$)
270 DRAW"BM130,140":PRINT#1,A2;GOTO 300
280 A$=INPUT$(1):A1=VAL(A$):DRAW"BM130,140":PRINT#1,A1
290 B$=INPUT$(1):A2=VAL(B$):DRAW"BM130,140":PRINT#1,A2
300 W=A1X10+A2: W$=STR$(W)
310 IF R=W THEN 320 ELSE 350
320 C1=C1+1 : DRAW "BM10,160": PRINT#1,"ПРАВИЛЬНО"
330 GOSUB 560
340 GOTO 380
350 C2=C2+1:DRAW "BM10,160":PRINT#1,"ВЫ ОШИБЛИСЬ. "
360 DRAW"BM10,170":PRINT#1,"ПРАВИЛЬНЫЙ ОТВЕТ: ";R
370 GOSUB 650
380 LINE (150,0)-(250,12),6,BF
390 DRAW"BM155,4":PRINT#1,"C1=";C1;"C2=";C2
400 FOR J=1 TO 400 : NEXT J
410 NEXT N
420 SCREEN 0
430 LOCATE 5,10
440 PRINT "ЧИСЛО ПРАВИЛЬНЫХ ОТВЕТОВ =";C1
450 IF C1=0 OR C1=1 THEN P=21 GOTO 470
460 P=INT(C1/10*5)
470 PRINT "ВАША ОЦЕНКА ";P
480 END
490 A=INT(RND*(-TIME)*95+5)
500 B=INT(RND*(-TIME)*A)
510 R=A-B:K=LEN(MID$(STR$(R),2,2))
520 DRAW"BM10,120":PRINT#1,"ЧЕМУ РАВНО ЗНАЧЕНИЕ ВЫРАЖЕНИЯ"
530 DRAW"BM10,130":PRINT#1,STR$(A)+"-"+STR$(B)
540 DRAW"BM10,140":PRINT#1,"ВВЕДИТЕ ОТВЕТ:"
550 RETURN
560 I=1
570 I=1
580 FOR K=1 TO 15
590 PUT SPRITE 0, (X,30),,I
600 X=X+1 : I=I+1
610 IF I>2 THEN I=1
620 FOR J=1 TO 25 : NEXT J
630 NEXT K
640 RETURN
650 I=3
660 I=3
670 FOR K=1 TO 15
680 PUT SPRITE 1, (Y,84),,I
690 Y=Y+1 : I=I+1
700 IF I=4 THEN I=3
710 FOR J=1 TO 25 : NEXT J
720 NEXT K
730 RETURN
740 DATA 0,7,8,8,8,8,7,1,1,1,2,2,3,0
750 DATA 0,224,16,76,16,112,15,16,224,128,64,64,64,32,0

```

```

760 DATA 0,7,8,8,8,8,7,1,1,1,1,1,1,1,0
770 DATA 0,224,16,80,16,112,16,224,128,64,32,16,8,4,0
780 DATA 0,0,0,64,32,16,31,31,31,40,36,40,0,0
790 DATA 0,0,0,136,112,108,120,240,248,20,36,40,0,0
800 DATA 0,0,0,64,32,16,31,31,31,63,104,68,0,0
810 DATA 0,0,0,136,112,108,120,240,248,20,52,66,0,0

```

Заметим, что используемый в программе оператор **ON** к **DATA** - последовательность номеров строк > осуществляет переход на одну из строк с заданным номером **NS** в зависимости от значения выражения **K**: если **INT(K) = I**, то - **NSI**, если **INT(K) = 2**, то **NS2** и т.д.

3.2. Игровая программа "Формулы приведения"

Программа предназначена для проверки знания учащимися старших классов формул приведения по тригонометрии.

После запуска программы в работу на экране появляется главный кадр «Игровая программа по теме: "Формулы приведения"», сменившийся через некоторое время кадром с игровым фоном и первым заданием на проверку знания формул приведения. В нижней части кадра приводится весь набор возможных ответов для предлагаемых заданий (см. рисунок 3.2).

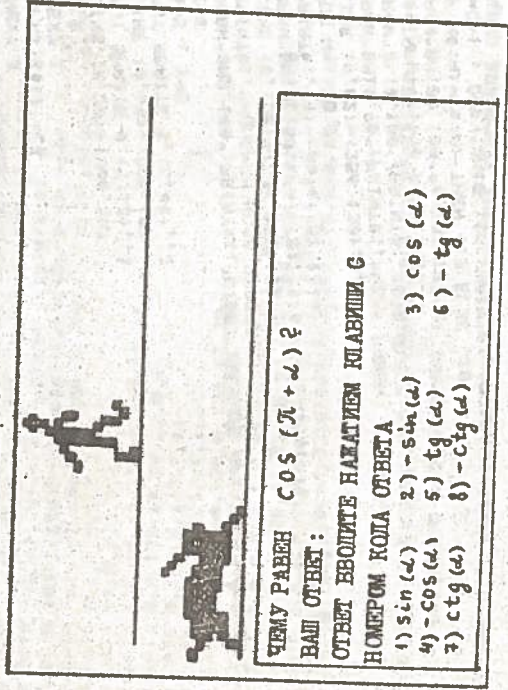


Рис. 3.2

Вводимый учащимся код ответа сравнивается с кодом эталонного ответа. При их совпадении на экран выводится подтверждение "ПРАВИЛЬНО", спринтер перемещается на некоторое расстояние вперед, при несовпадении - выводится реплика "ВЫ ОШИБЛИСЬ", на некоторое расстояние вперед перемещается запяц. После завершения опроса на экран выводится число верных ответов учащегося и итоговая оценка. Приведем теперь текст программы.

Программа 68

```

5 REM ФОРМУЛЫ ПРИВЕДЕНИЯ
10 DIM P$(30), L$(30)
20 FOR I=1 TO 24
30 READ P$(I), L$(I)
40 NEXT I
50 FOR I=1 TO 8
60 READ R$(I)
70 NEXT I
80 SCREEN 2,2
90 OPEN "GRP:" AS#1
100 DRAW"BM40,50": PRINT#1,"ИГРОВАЯ ПРОГРАММА"
110 DRAW"BM70,70": PRINT#1,"ПО ТЕМЕ:"
120 DRAW"BM40,90": PRINT#1,"ФОРМУЛЫ ПРИВЕДЕНИЯ"
130 FOR I=1 TO 200 : NEXT I
140 FOR I=1 TO 4
150 C$=""
160 FOR J=1 TO 32
170 READ A
180 C$=C$+CHR$(A)
190 NEXT J
200 SPRTES$(I)=C$
210 NEXT I
220 CLS
230 LINE (0,30)-(500,30)
240 PUT SPRITE 0,(5,15),1
250 LINE (0,50)-(500,60)
260 PUT SPRITE 1,(5,44),3
270 X=5 : Y=5
280 C=0
290 DRAW"BM2,140":PRINT#1,"ОТВЕТ ВВОДИТЕ НАЖАТИЕМ КЛАВИШИ C";
300 PRINT#1,"НОМЕРОМ КОДА ОТВЕТА:"
310 LINE (0,158)-(500,185),12,BF
320 DRAW"BM10,160":PRINT#1,"1)sin(a) 2)-sin(a) 3)cos(a) 4)tg(a) 5)-cos(a) 6)-tg(a) 7)ctg(a) 8)-ctg(a)";
330 PRINT#1,"4) -cos(a) 5) tg(a) 6) -tg(a) 7)ctg(a) 8) -ctg(a)";
340 FOR N=1 TO 24
350 LINE (0,82)-(256,130),6,BF
360 DRAW"BM10,90":PRINT#1,"ЧЕМУ РАВЕН ";P$(N)
370 DRAW"BM10,100":PRINT#1,"ВАШ ОТВЕТ:"
380 A$=INPUT$(1) : A=VAL(A$)
390 DRAW"BM100,100" : PRINT#1, A
400 IF P$(A)=L$(N) THEN 410 ELSE J=0
410 C=C+1 : DRP="BM10,110":PRINT#1,"ПРАВИЛЬНО"
420 I=1

```

```

430 FOR K=1 TO 10
440 PUT SPRITE 0,(X,13),1
450 X=X+1 : I=I+1
460 IF I>2 THEN I=1
470 FOR J=1 TO 25 : NEXT J
480 NEXT K
490 FOR J=1 TO 200 : NEXT J
500 GOTO 610
510 DRAW "BM150,100" :PRINT#1,"ВЫ ОШИБЛИСЬ"
520 DRAW"BM10,110":PRINT#1,"ПРАВИЛЬНЫЙ ОТВЕТ:";L$(N)
530 I=3
540 FOR K=1 TO 10
550 PUT SPRITE 1,(Y,44),1
560 Y=Y+1 : I=I+1
570 IF I>4 THEN I=3
580 FOR J=1 TO 25 : NEXT J
590 NEXT K
600 FOR J=1 TO 400 : NEXT J
610 NEXT N
620 SCREEN 0
630 LOCATE 5,10
640 PRINT "ЧИСЛО ПРАВИЛЬНЫХ ОТВЕТОВ";C
650 IF C=0 OR C=1 THEN P=2 ELSE P=INT(RND(-TIME)*95+5)
660 LOCATE 5,14
670 PRINT "ВАША ОЦЕНКА";P
680 END
690 DATA cos(pi/2+a),-sin(a),cos(pi+a),-cos(a),cos(3pi/2+a)
700 DATA sin(a),sin(pi/2+a),cos(a),sin(pi+a),-sin(a)
710 DATA sin(3pi/2+a),-cos(a),tg(pi/2+a),-ctg(a)
720 DATA tg(pi+a),tg(a),tg(3pi/2+a),-ctg(a)
730 DATA -tg(a),ctg(pi+a),ctg(a),ctg(3pi/2+a),-tg(a)
740 DATA cos(pi/2-a),sin(a),cos(pi-a),-cos(a),cos(3pi/2-a)
750 DATA -sin(a),sin(pi/2-a),cos(a),sin(pi-a),sin(a)
760 DATA sin(3pi/2-a),-cos(a),tg(pi/2-a),ctg(a),tg(pi-a)
770 DATA -tg(a),tg(3pi/2-a),ctg(a),ctg(3pi/2-a)
780 DATA tg(a),ctg(pi-a),-cos(a),tg(a),tg(a),sin(a)
790 DATA -sin(a),cos(a),-cos(a),tg(a),ctg(a),-ctg(a)
800 DATA 0,1,1,0,1,3,7,11
810 DATA 19,3,2,2,2,4,6
820 DATA 128,64,64,128,0,128,64,64
830 DATA 0,128,64,128,0,32,0,0
840 DATA 0,1,1,0,1,3,7,11
850 DATA 11,3,5,9,17,9,1,1
860 DATA 128,64,64,128,0,128,64,32
870 DATA 0,0,0,0,0,0,0,128
880 DATA 0,0,0,0,0,0,7,95,63,63,31,12,28,32,32,0
890 DATA 0,0,0,128,64,32,28,22,254,248,248,28,2,1,0
900 DATA 0,0,0,0,0,0,7,95,63,63,31,12,28,16,0,0
910 DATA 0,0,0,128,64,32,28,22,254,248,248,28,4,4,0

```

3.3. Игровая программа "Исследование треугольника, заданного координатами своих вершин"

Программа предназначена для проверки знания учащимися старших

классов условий существования треугольника, понятий правильного, равнобедренного, остроугольного, прямоугольного треугольника, умный вычислять периметр и площадь треугольника.

При запуске программы в работу на экране появляется заглавный кадр "Программа исследования: треугольника" и исполняется музыкальный фрагмент из песни В. Шаинского "Чему учат в школе". После его завершения происходит смена кадра и на экране появляется предложение "ВВЕДИ СВОЕ ИМЯ С КЛАВИАТУРЫ ТЕРМИНАЛА". Затем компьютер приветствует учащегося, появляется предписание об исследовании треугольника, координаты вершин которого генерируются программой и выводятся на экран. Кроме того, на экран выволятся вычисляемые программой на основе этих координат значения длин сторон треугольника в убывающем порядке. Эти значения учащемуся предлагаются записать в тетрадь, т.к. они нужны для дальнейшей работы. Далее на экране появляются поочередно вопросы и варианты ответов на них (см. рисунок 3.3). Учащийся вводит ответ, нажимая на соответствующую курсорную клавишу. При правильном ответе на экране появляется подтверждение "ПРАВИЛЬНО", буллетер перепрыгивает луку, исполняется мажорная музыка. При неправильном ответе выдается реллика "НЕПРАВИЛЬНО", буллетер тонет в луке, звучит минорная музыка. В заключительном кадре на экран выволятся число верных ответов учащегося и итоговая оценка.

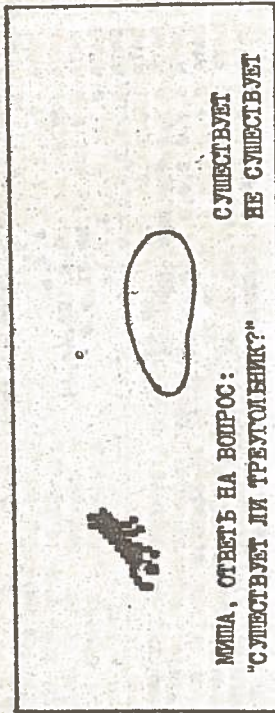


Рис. 3.3

Приведем теперь текст программы.

Программа 69

```

5 REM ИССЛЕДОВАНИЕ ТРЕУГОЛЬНИКА
10 DATA "СУЩЕСТВУЕТ", "НЕ СУЩЕСТВУЕТ", "ПРАВИЛЬНЫЙ", "НЕПРАВИЛЬНЫЙ"
20 DATA "РАВНОБЕДРЕННЫЙ", "НЕРАВНОБЕДРЕННЫЙ", "ОСТРОУГОЛЬНЫЙ"
30 DATA "ТУПОУГОЛЬНЫЙ", "ПРЯМОУГОЛЬНЫЙ", "НЕПРЯМОУГОЛЬНЫЙ"
40 FOR I=1 TO 5
50 READ T$(1), T1$(1)
60 NEXT I
70 S$(1)="СУЩЕСТВУЕТ ЛИ ТРЕУГОЛЬНИК?"
80 S$(2)="ЯВЛЯЕТСЯ ЛИ ТРЕУГОЛЬНИК ПРАВИЛЬНЫМ?"
90 S$(3)="ЯВЛЯЕТСЯ ЛИ ТРЕУГОЛЬНИК РАВНОБЕДРЕННЫМ?"
100 S$(4)="ЯВЛЯЕТСЯ ЛИ ТРЕУГОЛЬНИК ОСТРОУГОЛЬНЫМ?"
110 S$(5)="ЯВЛЯЕТСЯ ЛИ ТРЕУГОЛЬНИК ПРЯМОУГОЛЬНЫМ?"
120 E$(1)="ТРЕУГОЛЬНИК СУЩЕСТВУЕТ, Т.К. L1<L2+L3"
130 H$(1)="ТРЕУГОЛЬНИК НЕ СУЩЕСТВУЕТ, Т.К. L1>L2+L3"
140 E$(2)="ТРЕУГОЛЬНИК ПРАВИЛЬНЫЙ, Т.К. L1=L2=L3"
150 H$(2)="ТРЕУГОЛЬНИК НЕПРАВИЛЬНЫЙ, Т.К. НЕ ВЫПОЛНЕНО УСЛОВИЕ
L1=L2=L3"
160 E$(3)="ТРЕУГОЛЬНИК РАВНОБЕДРЕННЫЙ, Т.К. ДВЕ ЕГО СТОРОНЫ РАВНЫ"
170 H$(3)="ТРЕУГОЛЬНИК НЕ РАВНОБЕДРЕННЫЙ, Т.К. НИ У ОДНОЙ ПАРЫ
ЕГО СТОРОН ДЛИНЫ НЕ РАВНЫ"
180 E$(4)="ТРЕУГОЛЬНИК ОСТРОУГОЛЬНЫЙ, Т.К. L1^2<L2^2+L3^2"
190 H$(4)="ТРЕУГОЛЬНИК ТУПОУГОЛЬНЫЙ, Т.К. L1^2>L2^2+L3^2"
200 E$(5)="ТРЕУГОЛЬНИК ПРЯМОУГОЛЬНЫЙ, Т.К. L1^2=L2^2+L3^2"
210 H$(5)="ТРЕУГОЛЬНИК НЕПРЯМОУГОЛЬНЫЙ, Т.К. L1^2>L2^2+L3^2"
220 V$(6)="ПЕРИМЕТР ТРЕУГОЛЬНИКА РАВЕН:L1+L2+L3"
230 V$(7)="ПЛОЩАДЬ ТРЕУГОЛЬНИКА РАВНА ПОЛОВИНЕ ПРОИЗВЕДЕНИЯ
ОСНОВАНИЯ НА ВЫСОТУ"
240 S$(6)="ЧЕМУ РАВЕН ПЕРИМЕТР ТРЕУГОЛЬНИКА?"
250 S$(7)="ЧЕМУ РАВНА ПЛОЩАДЬ ТРЕУГОЛЬНИКА?"
260 S$(8)="(ВЫСОТА, ПРОВЕДЕННАЯ К ЕГО БОЛЬШЕЙ СТОРОНЕ, РАВНА"
270 S$(9)="ВВЕДИ ОТВЕТ:"
280 S$(10)=" С ДВУХА ЗНАКАМИ ПОСЛЕ ЗАПЯТОЙ"
290 V$(8)="", ЗАВЕРШАЯ ВВОД НАЖАТИЕМ НА"
300 DATA 0,0,0,0,0,0,1,7
310 DATA 15,63,92,156,188,168,40,0
320 DATA 0,0,40,56,104,252,240,224
330 DATA 224,224,160,144,64,0,0,0
340 DATA 0,0,0,0,0,0,1,7
350 DATA 15,63,92,156,188,168,40,0
360 DATA 0,0,40,56,124,252,240,224
370 DATA 224,224,160,144,64,0,0,0
380 SCREEN 7,2
390 FOR J=1 TO 2
400 C$=""
410 FOR I=1 TO 32
420 READ A
430 C$=C$+CHR$(A)
440 NEXT I
450 SPRITE$(J)=C$
460 NEXT J
470 S2=0
480 W$=""
490 Q1$=""
500 Q$=""
510 P$=""

```

ОТВЕТЬ НА ВОПРОС;
 КЛАВИШУ ВОЗВРАТА КАРЕТКИ"
 "ДЛЯ ПРОДОЛЖЕНИЯ РАБОТЫ НАЖМИ"
 "ОТВЕЧАЯ НА ПРЕДЛАГАЕМЫЕ ВОПРОСЫ,"

```

520 P2$="НАЖИМАЙ НА КУРСОРНУЮ КЛАВИШУ."
530 P3$="СОТВЕТСТВУЮЩУЮ ВЫБРАННОМУ"
540 P4$="ВАРИАНТУ ОТВЕТА"
550 CLS: SCREEN 2
560 LINE (30,30)-(225,160),6,BF
570 OPEN "GRP":AS#1
580 PSET (90,50):PRINT#1,"ПРОГРАММА"
590 PSET (70,80):PRINT#1,"ИССЛЕДОВАНИЯ"
600 PSET (70,110):PRINT#1,"ТРЕУГОЛЬНИКОВ"
610 GOSUB 2040
620 FOR I=1 TO 200: NEXT I
630 SCREEN 2
640 DRAW "BMS,50"
650 DRAW "BM5,50":PRINT#1,"ВВЕДИ СВОЕ ИМЯ С КЛАВИАТУРЫ"
660 DRAW "BMB0,70":PRINT#1,"ТЕРМИНАЛА"
670 F$=""
680 PRESET (90,100)
690 V$=INPUT$(1)
700 IF V$=CHR$(13) THEN 740
710 PRINT#1, V$;
720 F$=F$+V$
730 GOTO 690
740 CLS
750 DRAW "BMS0,100":PRINT#1,"ЗОРРАВСТВУЙ, "F$
760 FOR I=1 TO 500: NEXT I
770 CLS
780 DEF FN(LX,Y)=SQR(LXX+YY)
790 DRAW "BM2,5":PRINT#1,"ИССЛЕДУЙ ТРЕУГОЛЬНИК, ЗАДАНЫЙ"
800 DRAW "BM2,15":PRINT#1,"КООРДИНАТАМИ СВОИХ ВЕРШИН"
810 LINE (50,70)-(120,40)
820 LINE (120,40)-(160,100)
830 LINE (50,70)-(160,100)
840 DRAW "BM20,80":PRINT#1,"(A1,B1)"
850 DRAW "BM100,30":PRINT#1,"(A2,B2)"
860 DRAW "BM140,110":PRINT#1,"(A3,B3)"
870 A1=INT(RND(-TIME)*50+25)
880 B1=INT(RND(-TIME)*70+40)
890 A2=INT(RND(-TIME)*120+60)
900 B2=INT(RND(-TIME)*40+20)
910 A3=INT(RND(-TIME)*160+40)
920 B3=INT(RND(-TIME)*100)
930 FOR I=1 TO 500:NEXT I
940 CLS
950 DRAW "BM2,5":PRINT#1,"ПУСТЬ A1=";A1;"A2=";A2;
960 PRINT#1,"B2=";B2;"B3=";B3;"B3=";B3
970 L1=FN(LA2-A1,B2-B1)
980 L2=FN(LA3-A1,B3-B2)
990 L3=FN(LA3-A1,B3-B1)
1000 M=L1:Q=L2:G=L3
1010 IF M>Q THEN 1030
1020 V=M:M=Q:G=V
1030 IF M>G THEN 1050
1040 V=M:M=G:G=V
1050 IF Q>G THEN 1070
1060 V=Q:Q=G:G=V
1070 M=.01*INT(100*M+.5)
1080 Q=.01*INT(100*Q+.5)

```

```

1090 G=.01*INT(100*G+.5)
1100 DRAW "BM1,30":PRINT#1,F$;," ЗАПИШИ В ТЕТРАДЬ ЗНАЧЕНИЯ"
1110 DRAW "BM1,40":PRINT#1," ДЛИН СТОРОН ТРЕУГОЛЬНИКА: "
1120 DRAW "BM1,50":PRINT#1,"L1=";M;"L2=";J;Q;"L3=";G
1130 DRAW "BM2,80":PRINT#1,P1$
1140 DRAW "BM2,95":PRINT#1,P1$
1150 DRAW "BM2,110":PRINT#1,P2$
1160 DRAW "BM2,125":PRINT#1,P4$
1170 IF M<Q THEN P(1)=1 ELSE P(1)=5
1180 IF (M=Q)AND(M=6) THEN P(2)=1 ELSE P(2)=5
1190 IF (M=Q)OR(M=6) THEN P(3)=1 ELSE P(3)=5
1200 F=M*Q-Q*X0-G*X0
1210 IF F<0 THEN P(4)=1 ELSE P(4)=5
1220 IF ABS(F)<1E-10 THEN P(5)=1 ELSE P(5)=5
1230 IF P(1)=1 THEN V$(1)=E$(1) ELSE V$(1)=H$(1)
1240 IF P(2)=1 THEN V$(2)=E$(2) ELSE V$(2)=H$(2)
1250 IF P(3)=1 THEN V$(3)=E$(3) ELSE V$(3)=H$(3)
1260 IF P(4)=1 THEN V$(4)=E$(4) ELSE V$(4)=H$(4)
1270 IF P(5)=1 THEN V$(5)=E$(5) ELSE V$(5)=H$(5)
1280 FOR I=1 TO 500: NEXT I
1290 DRAW "BMS,150":PRINT#1,Q$
1300 DRAW "BMS,165":PRINT#1,Q1$
1310 AS=INKEY$
1320 IF AS="" THEN 1310
1330 CLOSE #1
1340 CLS
1350 SCREEN 7
1360 OPEN "GRP": AS#1
1370 C=0
1380 FOR N=1 TO 5
1390 CLS
1400 GOSUB 2160
1410 DRAW "BM40,130":PRINT#1,F$;M$
1420 DRAW "BM10,140":PRINT#1,S$(N)
1430 DRAW "BM350,128":PRINT#1,T$(N)
1440 DRAW "BM480,123":PRINT#1," "
1450 DRAW "BM480,130":PRINT#1," "
1460 DRAW "BM350,145":PRINT#1,T1$(N)
1470 DRAW "BM480,150":PRINT#1," "
1480 DRAW "BM480,143":PRINT#1," "
1490 R=STICK(0)
1500 E=P(N)
1510 IF R=0 THEN 1490
1520 GOSUB 2210
1530 FOR I=1 TO 800: NEXT I
1540 NEXT N
1550 CLS
1560 N=6
1570 GOSUB 2160
1580 DRAW "BM10,130":PRINT#1,S$(6)
1590 DRAW "BM10,140":PRINT#1,S$(9)
1600 R=h+G+Q
1610 FOR J=1 TO 9: L$=INKEY$: NEXT J
1620 L$=""
1630 DRAW "BM120,140"
1640 T$=INPUT$(1)
1650 IF ASC(T$)=13 THEN 1680

```

```

1660 PRINT#1, T$; L$=L$+T$
1670 GOTO 1640
1680 E=VAL(L$)
1690 GOSUB 2210
1700 IF E<>R THEN DRAW"BM200,200";PRINT#1,"P=";R
1710 FOR I=1 TO 800:NEXT I
1720 CLS
1730 N=7
1740 L=R/2;S1=80R*(L-M)*(L-G)*(L-Q)
1750 H=INT(100*(2*X1)/M+.5)/100
1760 GOSUB 2160
1770 DRAW"BM10,130";PRINT#1,S$(7)
1780 DRAW"BM10,140";PRINT#1,S$(8);H;"")
1790 DRAW"BM10,150";PRINT#1,S$(9)
1800 S=INT(100*(H*X1)/2+.5)/100
1810 Z$=""
1820 DRAW "BM120,150"
1830 T$=INPUT$(1)
1840 IF ASC(T$)=13 THEN 1880
1850 PRINT#1, T$,
1860 Z$=Z$+T$
1870 GOTO 1830
1880 E=VAL(Z$):R=S
1890 GOSUB 2210
1900 IF E<>R THEN DRAW"BM200,200";PRINT#1,"S=";S
1910 FOR I=1 TO 800:NEXT I
1920 CLS
1930 DRAW "BM20,50": PRINT#1, "ЧИСЛО ПРАВИЛЬНЫХ ОТВЕТОВ РАВНО ";S
1940 IF C=7 THEN 1950 ELSE 1960
1950 E=5:GOTO 1990
1960 IF C>4 THEN 1970 ELSE 1980
1970 E=4:GOTO 1990
1980 IF C>2 THEN E=C-1 ELSE E=2
1990 DRAW "BM30,70": PRINT#1, "УВЯ ОЦЕНКА ";E
2000 DRAW "BM50,100";PRINT#1,"ДО СВИДАНИЯ, ";F$
2010 FOR I=1 TO 800: NEXT I
2020 CLS
2030 STOP
2040 REM ШКОЛА
2050 DATA 03AB,04FB
2060 DATA FB,EB,EB,DE,D4,03AB,04FB,FB,EB,EB,DE,D4
2070 DATA 04DB,FB,AB,AB,GB,FB,AB,GB,FB,GB,GA,A4,R4
2080 DATA RB,04DB,05CB,CB,04B-B,B-B,AB,AA,DE,BB,BB,AB,AB,GB,GB
2090 DATA 04FB,GB,B-B,AB,FB,DE,FB,EB,AB,C+B,E4,D4,R4
2100 PLAY "V10T180"
2110 FOR I=1 TO 55
2120 READ F$
2130 PLAY F$
2140 NEXT I
2150 RETURN
2160 LINE (1,5)-(490,120),12,BF
2170 CIRCLE (250,100),60,5,.,1/6
2180 PAINT (250,100),5
2190 PUT SPRITE 0,(50,90),15,1
2200 RETURN
2210 IF R=E THEN 2370
2220 FOR X=50 TO 110 STEP 6

```

```

2230 FOR J=1 TO 2
2240 PUT SPRITE 0,(X,150*(1-.6*X/SIN(X/50))),15,J
2250 FOR K=1 TO 10
2260 NEXT K
2270 NEXT J
2280 NEXT X
2290 DRAW"BM200,160";PRINT#1,"НЕПРАВИЛЬНО"
2300 DRAW"BM50,180" I PRINT#1,V$(N)
2310 E$="T180V120S1M100E16R16M2";PLAY E$
2320 PUT SPRITE 0,(110,50*(1-.6 *X/SIN(X/50))),5,J
2330 FOR X=1 TO 3
2340 CIRCLE (250,100),60-10*X,15,.,1/6
2350 NEXT X
2360 GOTO 2470
2370 FOR X=50 TO 180 STEP 6
2380 FOR J=1 TO 2
2390 PUT SPRITE 0,(X,150*(1-.6*X/SIN(X/70))),15,J
2400 FOR K=1 TO 10
2410 NEXT K
2420 NEXT J
2430 NEXT X
2440 DRAW"BM200,160":PRINT#1,"ПРАВИЛЬНО"
2450 C=C+1
2460 PLAY "VBL4CL8EG0SL16CCCCCCCC160DDDDDDDL4EDL8CD4L8G0SL4C"
2470 RETURN

```

3.4. Контролирующая программа с игровыми элементами по теме: "Видовременные формы глаголов"

Программа содержит упражнения на употребление глаголов в форме PRESENT PERFECT. Всего обучаемому предлагается шесть упражнений, которые генерируются на основе, содержащего десять упражнений. В выводимом на экран монитора предложении глагол, который нужно употребить в правильной форме, приведен в скобках. Обучаемому предлагается ввести ответ в содержательной форме с клавиатуры. Допускается возможность двух попыток ввода ответа. Если обе они будут неудачными, то на экран выводится правильный ответ и осуществляется переход к следующему упражнению. Работа организована в режиме диалога с компьютером и происходит на игровом фоне с участием колобка (см. рисунок 3.4). При правильном ответе колобок покрывает под мажорную мелодию, при неправильном - снижает под грустную мелодию. По завершении работы на экран выводится число верных ответов, попыток повторного ответа и итоговая оценка.



ИСПОЛЬЗУЙТЕ ГЛАГОЛЫ, ДАННЫЕ В СКОБКАХ, В ФОРМЕ ПРЕСЕНТ ПЕРВОГО

ЗАДАНИЕ 1

SHOW (TO SEE) JOHN THIS WEEK

ВВЕДИТЕ ОТВЕТ, ЗАВЕРШАЯ ВВОД НАЖАТИЕМ НА
КЛАВИШУ ВОЗВРАТА КАРЕТКИ

Рис. 3.4

Приведем теперь текст программы.
Программа 70

```

10 DIR P15(10),K15(10),P25(10),K25(10),P35(10),K35(10)
20 SCREEN 2
30 OPEN "GRP:" AS#1
40 DRAW"BM35,60":PRINT#1,"КОНТРОЛИРУЮЩАЯ ПРОГРАММА"
50 DRAW"BM 85,80":PRINT#1,"ПО ТЕПЕРЬ"
60 DRAW"BM10,115":PRINT#1,"ВМОД-ВРЕМЕННЫЕ ФОРМЫ ГЛАГОЛОВ"
70 DATA E4,E4,A4,C+4,B4,E4,C80+4,A4,0A94,E+4
80 DATA A4,C+4,E2,D2,F+4,DA,F+2F+0,D2D4
90 DATA F+4,F+4,G+4,G+4,A4,A4,B4,G+4,G+4,E4,E4
100 DATA A4,08C+4,G+4,A2A4,C+2C+4,G+4,03B4
120 DATA A4,03B4,G+4,B4,C+2C+4,04C+2C+4
130 PLAY "V10T200","V10T200"
140 FOR I=1 TO 25
150 READ A#,B#;PLAY A#,B#
160 NEXT I
170 FOR I=1 TO 2000:NEXT I
180 SCREEN 7,3
190 CLS
200 CIRCLE(100,100),70
210 CIRCLE(45,45),10
220 CIRCLE(115,94),10

```

```

230 CIRCLE(80,110),5
240 CIRCLE(80,116),20,3,5,6
254 DRAW"BM250,50":PRINT#1,"КАК ВАС ЗОВУТ?"
260 DRAW"BM200,85":PRINT#1,"ВВЕДИТЕ СВОЕ ИМЯ С КЛАВИАТУРЫ."
270 DRAW"BM200,100":PRINT#1,"ВКЛЮЧИВ КЛАВИШУ CAPS И РУС"
280 DRAW"BM200,120":PRINT#1,"И ЗАВЕРШАЯ ВВОД НАЖАТИЕМ НА"
290 DRAW"BM200,140":PRINT#1,"КЛАВИШУ ВОЗВРАТА КАРЕТКИ."
300 PRESET (200,180)
310 AS=" "
320 BS=INPUT$(1)
330 IF BS=CHR$(0) THEN 380
340 IF ASC(BS)=13 THEN 430
350 PRINT#1,BS;
360 AS=AS+BS;
370 GOTO 320
380 K=LEN(AS)-1:IF K<0 THEN 310
390 AS=LEFT$(AS,K)+ " "
400 PRESET(200,180):PRINT#1,AS;AS=LEFT$(AS,K)
410 K=K-1
420 GOTO 320
430 FOR I=1 TO 100:NEXT I
440 FOR I=1 TO 4
450 CS=" "
460 FOR J=1 TO 32
470 READ A : CS=CS+CHR$(A)
480 NEXT J
490 SPRTES(1)=CS
500 NEXT I
510 CLS : PUTSPRITE 0,(10,80),,1
520 DRAW"BM200,70":PRINT#1,"ЗДРАВСТВУЙТЕ.":AS;:
530 DRAW"BM150,100"
540 PRINT#1,"КАС ПРИВЕТСТВУЕТ КОМПЬЮТЕР ЯМАХА MSX-2!"
550 PRINT#1,"НАДЕМСЬ,ЧТО НАШЕ ЗНАКОМСТВО БУДЕТ ПРИЯТНЫМ"
570 FOR I=1 TO 900:NEXT I
580 CLS
590 DRAW"BM120,60":PRINT#1,"ВАМ БУДЕТ ЗАДАН РЯД ВОПРОСОВ.
600 DRAW"BM120,80"
610 PRINT#1,"ОТВЕТ ВВЕДИТЕ ПРИ ВКЛЮЧЕННОЙ КЛАВИШЕ РУС"
620 DRAW"BM120,100":PRINT#1,"И ВКЛЮЧЕННОЙ КЛАВИШЕ CAPS. ЕСЛИ В"
630 DRAW"BM120,120":PRINT#1,"ПРОЦЕССЕ ВВОДА ВОЗНИКНЕТ НЕОБХОДИ-"
650 DRAW"BM120,160":PRINT#1,"СИМВОЛ, НАЖМИТЕ НЕВЕРНО ВВЕДЕННЫЙ"
660 DRAW"BM120,190":PRINT#1,"ИЛИТАК, МЫ НАЧИНАЕМ ..."
670 FOR I=1 TO 5:PLAY"K3400B3070C64C36":NEXT I
680 FOR I=1 TO 2000: NEXT I
690 GOSUB 1450
700 CLS : PUTSPRITE 0,(90,30),,1
710 LINE (0,85)-(512,100),6,BF
720 DRAW"BM10,90":PRINT#1,BI$;BZ$
730 C=0:F=0
740 T=1
750 FOR J=1 TO 6
760 N(0)=0
770 NEXT J

```

```

780 N=INT(RND*(-TIME)*10+1)
790 FOR J=1 TO 6
800 IF N=N(J) THEN 790
810 NEXT J
820 N(I)=N
830 F1=0
840 LINE (50,105)-(450,140).L0,BF
850 LINE (100,202)-(400,215).L2,BF
860 DRAW"BM170,110":PRINT#1,"ЗАДАНИЕ"IT
870 DRAW"BM70,125":PRINT#1,P1$(N(I))
880 DRAW"BM70,140":PRINT#1,W$
890 DRAW"BM70,150":PRINT#1,W1$
900 LINE (100,190)-(400,212).L2,BF
910 LINE (100,170)-(400,200).L2,BF
920 A$=""
930 PRESET (170,180)
940 B$=INPUT$(1)
950 IF B$=CHR$(6) THEN 1000
960 IF ASC(B$)=13 THEN 1050
970 PRINT#1,B$
980 A$=A$+B$
990 GOTO 940
1000 K=LEN(A$)-1:IF K<0 THEN 940
1010 A$=LEFT$(A$,K)+ "
1020 PRESET (170,180):PRINT#1,A$:
1030 A$=LEFT$(A$,K):PRESET STEP(-6,0)
1040 GOTO 940
1050 W0$=A$:A$=R1$(N(I))
1060 S0$UB 1450
1070 IF F0=0 THEN 1160
1080 F=F+F0
1090 F1=F+F0
1100 IF F1>1 THEN 1140
1110 DRAW"BM170,203":PRINT#1,"ВВЕДИТЕ ОТВЕТ СНОВА"
1120 FOR J=1 TO 800: NEXT J
1130 GOTO 910
1140 DRAW"BM120,203":PRINT#1,"ПРАВИЛЬНЫЙ ОТВЕТ: "(R1$(N(I)))
1150 FOR J=1 TO 800:NEXT J
1160 I=I+1
1170 IF I<=6 THEN 780
1180 CLS
1190 PUT BFRITE 0,(80,50).:1
1200 DRAW"BM100,10":PRINT#1,"ЧИСЛО ВЕРНЫХ ОТВЕТОВ = "I
1210 DRAW"BM100,20"
1220 PRINT#1,"ЧИСЛО ПОПЫТОК ПУРТОРНОГО ВВОДА ОТВЕТА = "I:F
1230 PRESET(80,120)
1240 IF F=0 THEN 1360
1250 IF F<=3 THEN 1340
1260 IF F<=5 THEN 1320
1270 IF F<=6 THEN 1300
1280 PRINT#1,"ВАШИ ЗНАНИЯ ПО ТЕМЕ НЕУДОВЛЕТВИТЕЛЬНЫЕ"
1290 GOTO 1420
1300 PRINT#1,"РАБОТА ЗАЧТЕНА. ВАШИ ЗНАНИЯ НЕДОСТАТОЧНО ТРЕВНЫЕ"
1310 GOTO 1420
1320 PRINT#1,"ВАШИ ЗНАНИЯ ПО ТЕМЕ УДОВЛЕТВИТЕЛЬНЫЕ"

```

```

1330 GOTO 1420
1340 PRINT#1,"ВАШИ ЗНАНИЯ ПО ТЕМЕ ХОРОШИЕ"
1350 GOTO 1420
1360 PRINT#1,"МОЛОДЕЦ: РАБОТА ВЫПОЛНЕНА УТИЛЧНО"
1370 DRAW"BM0,170":PRINT#1,"СОБИРАТЕ СВОЮ ОЦЕНКУ УЧИТЕЛЯ"
1380 DRAW"BM20,190"
1390 PRINT#1,"ДЛЯ ВЫХОДА ИЗ ПРОГРАММЫ НАЖМИТЕ НА КЛАВИШУ ПРОБЕЛА"
1400 IF INKEY$="" THEN 1420 ELSE 1400
1410 CLS
1420 DRAW"BM150,150":PRINT#1,"ДО СЕИДАННЯ!"
1430 FOR I=1 TO 1000: NEXT I
1440 END
1450 REM БЛОК АНАЛИЗА ОТВЕТА
1460 F0=0
1470 IF W0$<>K$ THEN 1580
1480 DRAW "BM170,190":PRINT#1,"ПРАВИЛЬНО"
1490 C=C+1
1500 PLAY"MI228130SC404AB.A+160SC404F4L166AA+0500AA+8AB62"
1510 FOR K=1 TO 16
1520 PUT SPRITE 0,(90,14).:2
1530 FOR J=1 TO 25:NEXT J
1540 PUT SPRITE 0,(90,14).:3
1550 FOR J=1 TO 25:NEXT J
1560 NEXT K
1570 GOTO 1630
1580 DRAW"BM170,190":PRINT#1,"НЕПРАВИЛЬНО"
1590 F0=F0+1
1600 PLAY"MI1001250S050AC60"
1610 PUT SPRITE 0,(90,46).:4
1620 FOR J=1 TO 400:NEXT J
1630 PUT SPRITE 0,(90,30).:1
1640 RETURN
1650 K1$="УПОТРЕБИТЕ ГЛАГОЛЫ, ДАННЫЕ В СКОБКАХ, В ФОРМЕ "
1660 B2$="ПРЕБЕНТ PERFECT"
1670 W$="ВВЕДИТЕ ОТВЕТ. ЗАБЕДШАЯ ВВОД НАЖАТИЕМ НА"
1680 W1$="КЛАВИШУ ВЪЗЪРАТА КАРЕТКИ"
1690 P1$(1)="I (TO SPEAK) TO HIM ABOUT IT SEVERAL TIMES"
1700 K1$(1)="HAVE SPOKEN"
1710 P1$(2)="WE (TO LEARN) MANY NEW WORDS IN THIS COURSE"
1720 K1$(2)="HAVE LEARNED"
1730 P1$(3)="HE (TO MAKE) THE SAME MISTAKE SEVERAL TIMES"
1740 K1$(3)="HAS MADE"
1750 P1$(4)="I (TO HEAR) THAT STORY BEFORE"
1760 K1$(4)="HAVE HEARD"
1770 P1$(5)="I AM AFRAID THAT I (TO LOSE) MY CAR KEYS"
1780 K1$(5)="HAVE LOST"
1790 P1$(6)="SHE (TO SEE) JOHN THIS WEEK"
1800 K1$(6)="HAS SEEN"
1810 P1$(7)="SHE (TO OPEN) JUST THE DOOR"
1820 K1$(7)="HAS JUST OPENED"
1830 P1$(8)="WE (TO BUY) SOME MAGAZINES"
1840 K1$(8)="HAVE BOUGHT"
1850 P1$(9)="THE PASSENGER (TO COME) TO THE REFRESHMENT"
1860 K1$(9)="HAS COME"
1870 P1$(10)="THEY (TO GET) NEWSPAPER"

```

```

1800 R1$(10)="HAVE 50T"
1890 RETURN
1900 DATA 0,1,3,7,8,16,20,49,80,83,8,7,4,4,28,0
1910 DATA 0,128,192,224,16,8,40,140,10,202,16,224,32,32,56,0
1920 DATA 1,3,7,8,16,20,49,80,147,8,7,4,4,28,0,0
1930 DATA 128,192,224,16,8,40,140,10,201,16,224,32,32,56,0,0
1940 DATA 0,1,3,7,8,144,84,49,16,19,8,7,4,4,28,0
1950 DATA 0,128,192,224,16,9,42,140,8,200,16,224,32,32,56,0
1960 DATA 0,0,1,3,4,8,10,24,40,41,36,35,4,56,0,0
1970 DATA 0,0,192,224,16,8,40,140,10,139,18,226,16,14,0,0
1980 PLAY"MI2251305C40448.A+1605C404F4L166AA+USC04A+BA862"

```

4. Имитационное моделирование

При изучении математики и физики в средней школе широко используется математическое моделирование - исследование объектов познания с помощью математических моделей, представляющих собой некоторую совокупность уравнений, отображающих в упрощенной форме структуру, взаимосвязь и отношения между элементами исследуемого объекта. Новые знания об этом объекте получают в результате математического анализа таких моделей. Проводить этот анализ в условиях происходящих в настоящее время компьютеризации учебного процесса наиболее эффективно с помощью персональных компьютеров семейства MSX.

Работа в диалоговом режиме на персональных компьютерах семейства MSX делает возможным имитационное моделирование - исследование изучаемого явления с помощью компьютерной программы с отображением на экране монитора динамики этого явления. Такие программы предусматривают активное участие учащегося в управлении протекающим процессом с помощью клавиатуры терминала. Чем выше уровень взаимодействия учащегося с программой, тем больше вариантов позволяет она проследить при варьировании параметров, влияющих на ход работы программы. Это способствует выработке у учащегося умений самостоятельно приобретать новые знания, развивать исследовательских навыков. Имитационное моделирование способствует повышению мотивации в обучении и активности обучаемого, особенно если имитационные программы содержат игровые элементы.

Способен интерес имитационное моделирование представляет для

курса физики, так как не все изучаемые в этом курсе закономерности могут быть без затруднений продемонстрированы в условиях класса, в некоторых случаях такая демонстрация вообще невозможна.

Приведем пример простейших имитационных программ.

4.1. Исследование движения тела под действием силы тяжести

Программа исследования представляет собой пакет программ, предназначенных для изучения движения тела, брошенного вертикально вверх, горизонтально и под углом к горизонту. На начальном этапе работы программы на экран выводится меню, из которого учащемуся предлагается выбрать курсором интересующий его раздел. При работе по каждому из разделов учащийся вводит в соответствии с запросами программы исходные данные (например, при работе по второму разделу - начальную скорость V_0 , с которой бросается тело, и высоту H_0 тела над поверхностью земли). В соответствии с этими данными программа рассчитывает таблицу координат (x, y) точек, через которые проходит траектория движения тела, и выводит ее на экран, а также строит на экране эту траекторию в графическом режиме.

Приведем теперь текст программы.

Программа 71

```

5 REM ИССЛЕДОВАНИЕ ДВИЖЕНИЯ ТЕЛА ПОД ДЕЙСТВИЕМ СИЛЫ ТЯЖЕСТИ
10 KEY OFF:KEY1,"МЕНЮ":KEY2,"ВЫХОД"
20 G=9.81: P=3.1416
30 F#="!## !#####.### !#####.### !"
40 CLS: SCREEN 0: I=10
50 LOCATE 16,3:PRINT "ИМИТАЦИОННАЯ ПРОГРАММА ПО ТЕМЕ:"
60 LOCATE 12,5:PRINT "ДВИЖЕНИЕ ТЕЛА ПОД ДЕЙСТВИЕМ СИЛЫ ТЯЖЕСТИ"
70 LOCATE 14,8:PRINT"Выберите курсором нужный раздел:"
80 LOCATE 10,10:PRINT"1. Движение тела по вертикали"
90 LOCATE 10,11:PRINT"2. Движение тела по горизонтали"
100 LOCATE 10,12
110 PRINT"3. Движение тела, брошенного под углом к горизонту."
120 R=STICK(0)
130 IF R=1 AND I>10 THEN I=I-1:LOCATE 8,I+1:PRINT" "
140 IF R=5 AND I<12 THEN I=I+1:LOCATE 8,I-1:PRINT" "
150 LOCATE 8,I:PRINT"!" :BS=INKEY$:FOR A=1 TO 50:NEXT
160 IF CHR$(13)<>BS THEN 120 ELSE ON I-9 GOTO 170,430,650
170 CLS: LOCATE 10,0
180 PRINT"1. Движение тела по вертикали"
190 LOCATE 5,3:PRINT"Введите начальную высоту H0(м):"
200 PRINT"Тела над поверхностью Земли!"
210 INPUT H0
220 TI=SQ(2*H0/G): V1=5*TI

```

```

230 SCREEN 7: OPEN "GRP:" AS#1
240 PSET (10,0):PRINT#1,"Время падения тела T1=";CSNG(T1)
250 PSET (10,9):PRINT#1,"Скорость тела в момент ";
260 PRINT#1,"удара о землю V1=";CSNG(V1)
270 PSET (10,10):PRINT#1,"Положение тела через ";
280 PRINT#1,"промежутки времени .1XТ1"
290 DRAW"BM350,151R120NG3H3":PSET(400,150):PRINT#1,"OX"
300 PSET(0,32):PRINT#1," "
310 PSET(1,41):PRINT#1," | N | T | H | "
320 PSET(0,50):PRINT#1," | "
330 L=59
340 FOR N=0 TO 10
350 T=.1XТИXN
360 H=H0-.5GXТХТ
370 PSET(1,L):PRINT#1,USING F$;N,T,H:L=L+9
380 PSET(400,150-H/H0X110)
390 NEXT N
400 ON KEY GOSUB 40,990:KEY(1) ON:KEY(2) ON
410 PSET(250,200):PRINT#1,"F1-меню F2-выход":CLOSE #1
420 GOTO 420
430 CLS:M=1
440 LOCATE 10,0:PRINT"2. Движение тела, брошенного горизонтально"
450 LOCATE 5,3
460 INPUT"Введите начальную скорость V0(м/с) и высоту H(м)":V0,H0
470 T=SQRT(2*H0/G):S=V0XТ
480 SCREEN 7:OPEN "GRP:"AS#1:L=45
490 PSET(10,0):PRINT#1,"Время падения T=";CSNG(T)
500 PSET(10,9):PRINT#1,"Дальность полета S=";CSNG(S)
510 PSET(474,153)
520 PRINT#1,"s(m)":DRAW"e3Nh3L210U130NG5F3":PRINT#1,"H(m)"
530 K1=(V0XТ)/200:K2=H0/100
540 IF K1>K2 THEN K=K1 ELSE K=K2
550 PSET(0,18):PRINT#1," | N | X | Y | "
560 PSET(1,27):PRINT#1," | "
570 PSET(0,36):PRINT#1," | "
580 FOR N=0 TO 15
590 X=V0XТ/15XN:Y=H0-.5XGX(X/V0)X(X/V0)
600 PSET(1,L):PRINT#1,USING F$;N,X,Y:L=L+9
620 ON KEY GOSUB 40,990:KEY(1) ON:KEY(2) ON
630 PSET(250,200):PRINT#1,"F1-меню F2-выход ":CLOSE 1
640 GOTO 640
650 CLS:P$="(M) ":M=2:LOCATE 10,0
660 PRINT"2. Движение тела, брошенного под углом к горизонту"
670 LOCATE 5,3
680 INPUT"Введите начальную скорость V0(м/с)":V0
690 LOCATE 5,6
700 INPUT"Введите значение угла броска α(град), θ<α<P":L
710 IF V0>=10000 THEN V0=V0/1000:P$="(KM)":ELSE 730
720 IF V0>=1000 THEN V0=V0/1000:P$="(KM)":ELSE 730
730 L=L*P/180
740 T=2*XV0XСIN(L)/G:H=(V0XСIN(L))^2/2/G:S=V0XCOS(L)*T
750 SCREEN 7:OPEN "GRP:" AS#1:L1=54
760 PSET(10,0):PRINT#1,"Время движения T=";CSNG(T)
770 PSET(10,9):PRINT#1,"Максимальная высота H=";CSNG(H)
780 PSET(10,18):PRINT#1,"Дальность броска S=";CSNG(S)

```

```

790 IF L<P/2 THEN B10
800 PSET(510,150):DRAW"NH3NG3NL210U10U10NB3F3":L=200:GOTO 830
810 PSET(474,153)
820 PRINT#1,"s(m)":DRAW"e3Nh3L210U130NG5F3":PRINT#1,"H(m)"
830 K1=ABB(B/200):K2=H/100:IF K1>K2/2 THEN K=K1 ELSE K=K2
840 PSET(0,27)
850 PRINT#1," | N | X | Y | "
860 PSET(1,36)
870 PRINT#1," | N | X | Y | "
880 PRINT#1," | "
890 PSET(0,45):PRINT#1," | "
900 FOR N=0 TO 15 :X=V0XТ/15XN:Y=L
910 Y=YXСIN(L)*X/15XN-GX(T/15XN)^2/2
920 PSET(1,L1):PRINT#1,USING F$;N,X,Y:L1=L+9
930 PSET(300+0*X/K,150-Y/K/2):NEXT N
940 CLOSE#1
950 OPEN "grp:"AS#1
960 PSET(250,200):PRINT#1,"F1-меню F2-выход":CLOSE #1
970 ON KEY GOSUB 40,990:KEY(1) ON:KEY(2) ON
980 GOTO 980
990 KEY1,"color ":KEY2,"auto ":CLS:END

```

4.2. Исследование гармонических колебаний

Программа исследования представляет собой пакет программ, предназначенных для изучения на экране формы результирующего колебания от соотношения периодов и начальных фаз складываемых гармонических колебаний. На начальном этапе работы программы на экран выводится меню, включающее в себя следующие аспекты исследования гармонических колебаний:

- 1) Сложение колебаний с кратными периодами,
- 2) Сложение колебаний с близкими периодами (близки),
- 3) Сложение взаимно перпендикулярных колебаний.

Выбор соответствующего раздела осуществляется с помощью курсора. При работе с программами учащиеся анализируют амплитуд, периодов и фаз складываемых гармоник в каждом из разделов программы строят график результирующего колебания. Каждый раздел начинается с краткого пояснения влияния соотношения периодов, амплитуд и начальных фаз складываемых колебаний на форму результирующего колебания.

Приведен теперь текст программы.

Программа 72

```

5 REM ГАРМОНИЧЕСКИЕ КОЛЕБАНИЯ
10 P=6.28: MIDN 00: COLOR 15.4.4
20 KEY OFF:KEY1."МЕНЮ":KEY2."ВЫХОД"
30 DEF FNL(I)=A1*X1SIN(P*I/T1+F1)
40 DEF FNA(I)=M*A1*X1SIN(P*X1/T1+F2)
50 CLS: SCREEN 0
60 LOCATE 16,3:PRINT "ИМИТАЦИОННАЯ ПРОГРАММА ПО ТЕМЕ:"
70 LOCATE 16,7:PRINT "СЛУШЕНИЕ ГАРМОНИЧЕСКИХ КОЛЕБАНИЙ"
80 FOR J=1 TO 500: NEXT J
90 CLS: LOCATE 3,1
100 PRINT "РАССМОТРИМ ДВА ГАРМОНИЧЕСКИХ КОЛЕБАНИЯ:"
110 LOCATE 12,3:PRINT "X1=A1*X1SIN(2*PI*T1+F1)"
120 LOCATE 12,4:PRINT "X2=A2*X1SIN(2*PI*T2+F2)"
130 LOCATE 10,6:PRINT"ГДЕ A1,A2 - АМПЛИТУДЫ, T1,T2 - ПЕРИОДЫ, "
140 PRINT "F1,F2 - НАЧАЛЬНЫЕ ФАЗЫ КОЛЕБАНИЙ, ПИИ"
150 PRINT "СЛУШЕНИЕ ТАКИХ КОЛЕБАНИЙ, ПРИ "
160 PRINT "БАНИИ ЗАВИСИТ ОТ СООТНОШЕНИЯ ФОРМ РЕЗУЛЬТИРУЮЩЕГО КОЛЕ":
170 PRINT "ПЕРИОДОВ И НАЧАЛЬНЫХ ФАЗ СКЛАДЫВАЕМЫХ ГАРМОНИК."
180 LOCATE 3,10: PRINT "ПРОВОДЕМ ИССЛЕДОВАНИЕ ФОРМ РЕЗУЛЬТИРУЮЩ":
190 PRINT "ЩЕГО КОЛЕБАНИЯ ДЛЯ КОЛЕБАНИЙ С КРАТНЫ":
200 PRINT "МИ ПЕРИОДАМИ, БЛИЗКИМИ ПЕРИОДАМИ И ВЗАИМНО ПЕРПЕНДИКУ":
210 PRINT "ЛЯРНЫМИ КОЛЕБАНИЯ."
220 LOCATE 3,19
230 PRINT"ДЛЯ ПРОВОЖДЕНИЯ РАБОТЫ НАЖМИТЕ НА КЛАВИШУ ПРОВОДА"
240 IF INKEY#="" THEN GOTO 240 ELSE 240
250 SCREEN 0: I=16
260 LOCATE 10,8:PRINT"Выверите курсором нужные разделы:"
270 LOCATE 10,16:PRINT"1. СЛУШЕНИЕ КОЛЕБАНИЙ С КРАТНЫМИ ПЕРИОДАМИ"
280 LOCATE 10,17:PRINT"2. СЛУШЕНИЕ КОЛЕБАНИЙ С БЛИЗКИМИ ПЕРИОД":
290 PRINT"ДАМИ (ВНЕШНЯЯ)": LOCATE 10,18
300 PRINT"3. СЛУШЕНИЕ ВЗАИМНО ПЕРПЕНДИКУЛЯРНЫХ КОЛЕБАНИЙ"
310 KEYSTICK(0)
320 IF A=1 AND I>16 THEN I=I-1:LOCATE 8,I+1:PRINT" "
330 IF A=5 AND I<18 THEN I=I+1:LOCATE 8,I-1:PRINT" "
340 LOCATE 8,1
350 PRINT"В":BS=INKEYS:FOR A=1 TO 50:NEXT A
360 IF CHR$(13)>BS THEN 310 ELSE UN I-15 GOTO 370,650,1010
370 CLS: LOCATE 5,1
380 PRINT"2. СЛУШЕНИЕ КОЛЕБАНИЙ С КРАТНЫМИ ПЕРИОДАМИ"
390 LOCATE 3,5:PRINT "ПРОВОДИТЕ ИССЛЕДОВАНИЕ ФОРМ РЕЗУЛЬТИРУЮЩ":
400 PRINT "ЩЕГО КОЛЕБАНИЯ ДЛЯ ЗНАЧЕНИЙ"
410 PRINT "УТОШНЕНИЯ T1,T2 РАВНЫХ 1/2,1/3,1/4 И Т.Д. И РАЗНОСТИ":
420 PRINT "ФАЗ F1-F2 РАВНОЙ 0,PI/2,PI И Т.Д. НАЧИТЕ "
430 PRINT "ИССЛЕДОВАНИЕ ДЛЯ A1=A2=1,T1=1,T2=2,F1=1.57,F2=0."
440 LOCATE 3,10:INPUT "ВВЕДИТЕ АМПЛИТУДУ A1,A2":A1,A2
450 PRINT "ВВЕДИТЕ ПЕРИОДЫ T1,T2 "
460 INPUT " (КРАТНЫЕ ДРУГ ДРУГУ)":T1,T2
470 INPUT "ВВЕДИТЕ НАЧАЛЬНЫЕ ФАЗЫ F1,F2":F1,F2
480 F=1/T2: M=A2/A1
490 SCREEN 5: OPEN "GRP:" AS#1:COLOR 15.4.4
500 DRAW "BMS:150R210NH353"
510 FOR I=0 TO 5 STEP .02
520 Y=Y1+25*I:Y1=00-15*FNL(I):SET(X,Y)
530 PSE(X,Y),1

```

```

540 Y2=00-15*FNA(I)
550 PSE(X,Y2),7
560 Y=Y1+Y2-00
570 PSE(X,Y),1
580 NEXT I
590 DRAW"BMS:150":PRINT#1,"ЦВЕТА ГАРМОНИК: ПЕРВОЙ - 3, "
600 DRAW"BMS:150":PRINT#1,"ВТОРОЙ - 1, РЕЗУЛЬТИРУЮЩЕЙ - 1"
610 COLOR 15.4.4
620 ON KEY GOSUB 250,1370: KEY(I) ON: KEY(2) ON
630 DRAW"BMS:150":PRINT#1,"F1 - меню F2 - выход":CLOSE#1
640 GOTO 640
650 CLS: LOCATE 10,1:PRINT"2. ВНЕШНЯЯ"
660 LOCATE 3,3:PRINT"КОГДА ПЕРИОДЫ СЛАГАЕМЫХ КОЛЕБАНИЙ МАЛО РАЗЛИК":
670 PRINT"ЧАЮТСЯ, ТО В НЕКОТОРЫЕ ПРОМЕНУТКИ ВРЕМЕНИ КОЛЕБАНИЯ "
680 PRINT"ОКАЗЫВАЮТСЯ ПУЧТИ СОВПАДАЮЩИМИ ПО ФАЗЕ, И В ЭТО ВРЕМЯ ОНИ"
690 PRINT"УСИЛИВАЮТ ДРУГ ДРУГА. В ДРУГИЕ ПРОМЕНУТКИ ВРЕМЕНИ "
700 PRINT"КОЛЕБАНИЯ ОКАЗЫВАЮТСЯ ПОЧТИ "
710 PRINT"ПРОТИВОПОЛОЖНЫМИ ПО ФАЗЕ И ТОГДА ОНИ ГАСЯТ ДРУГ ДРУГА. "
720 PRINT"ТАКИЕ УСИЛЕНИЯ И ОСЛАБЛЕ":
730 PRINT"НИЯ РЕЗУЛЬТИРУЮЩЕГО КОЛЕБАНИЯ, ЧЕРЕДУЮСЯ, СЛЕДУЮТ ДРУГ ЗА "
740 PRINT"ДРУГОМ С ЧАСТОТОЙ."
750 PRINT"РАВНОЙ ЧАСТОТЫ НОСЯТ СЛАГАЕМЫХ КОЛЕБАНИЯ."
760 PRINT"ЭТИ ЯВЛЕНИЯ МОЖАТ НАЗЫВАТЬСЯ ВНЕШНЯЯ."
770 LOCATE 3,10:PRINT"ПРОВОДИТЕ ИССЛЕДОВАНИЕ ФОРМ РЕЗУЛЬТИРУЮЩЕГО "
780 PRINT"КОЛЕБАНИЯ ДЛЯ КОЛЕБАНИЙ ОДИНАКОВОГО АМПЛИТУДУ С ОТНОШЕНИЕМ "
790 PRINT"ПЕРИОДОВ T1:T2 РАВНЫМ 7:6,8:7,9:8 И Т.Д. И F1-F2 "
800 PRINT"РАВНОМ 0.1,0.2 И Т.Д. НАЧИТЕ ИССЛЕДОВАНИЕ ДЛЯ A1=A2=1, "
810 PRINT"Т1=.4, T2=.45, F1=F2=0."
820 LOCATE 0,16
830 INPUT "ВВЕДИТЕ АМПЛИТУДУ A1,A2":A1,A2
840 PRINT "ВВЕДИТЕ ПЕРИОДЫ T1,T2 "
850 INPUT " (БЛИЗКИЕ ДРУГ К ДРУГУ)":T1,T2
860 INPUT "ВВЕДИТЕ НАЧАЛЬНЫЕ ФАЗЫ F1,F2":F1,F2
870 K=T1/T2: M=A2/A1
880 SCREEN 2:OPEN "GRP:" AS#1:COLOR 15.4.4
890 DRAW "BMS:210U210NG3F3"
900 DRAW "BMS:30R210NH353"
910 DRAW "BMS:80R210NH353"
920 DRAW "BMS:150R210NH353"
930 FOR I=0 TO 5 STEP .03
940 X=X1+25*I:Y1=30-15*FNL(I):PSE(X,Y1)
950 Y2=00-15*FNA(I):PSE(X,Y2)
960 Y=Y1+Y2+40:PSE(X,Y),1
970 NEXT I
980 ON KEY GOSUB 250,1370: KEY(I) ON: KEY(2) UN
990 PSE(50,185):PRINT#1,"F1 - меню F2 - выход": CLOSE#1
1000 GOTO 1000
1010 CLS:LOCATE 15,1
1020 PRINT"3. СЛУШЕНИЕ ВЗАИМНО ПЕРПЕНДИКУЛЯРНЫХ КОЛЕБАНИЙ"
1030 LOCATE 3,3:PRINT"УЧАСТВУЯ В ДВУХ ВЗАИМНО ПЕРПЕНДИКУЛЯРНЫХ "
1040 PRINT "КОЛЕБАТЕЛЬНЫХ ДВИЖЕНИЯХ ОДИНАКОВОГО ПЕРИОДА, ТОЧКА "
1050 PRINT "СОВЕРШАЕТ РЕЗУЛЬТИРУЮЩЕЕ ДВИЖЕНИЕ ПО ЭЛЛИПТИЧЕСКОЙ "
1060 PRINT "ТРАЕКТОРИИ":PRINT "ВИД ЭТОГО ЭЛЛИПСА ЗАВИСИТ ОТ "
1070 PRINT "РАЗНОСТИ ФАЗ КОЛЕБАНИЙ. ЕСЛИ РАЗНИЦА ФАЗ "
1080 PRINT "РАВНА 0,PI,2PI, ТО ЭЛЛИПС ВЫРЯДКАЕТСЯ В ПРЯМУЮ ЛИНИЮ, "
1090 PRINT "ЕСЛИ ОНА РАВНА PI/2 ИЛИ 3PI/2,ТО - В ОКРУЖНОСТЬ."

```

```

1100 PRINT "ВОЗЬЕ СЛОЖИЕ ТРАЕКТОРИИ ПОЛУЧАЮТСЯ В ТЕХ СЛУЧАЯХ, КОГДА"
1110 PRINT "ПЕРИОДЫ СКЛАДЫВАЕМЫХ КОЛЕБАНИЙ НЕ ОДИНАКОВЫ. В "1
1120 PRINT "ЗЕРНИСТОСТИ ОТ СООТНОШЕНИЯ"; PRINT "ПЕРИОДОВ. АМПИЛИТУД "1
1130 PRINT "И НАЧАЛЬНЫХ ФАЗ СКЛАДЫВАЕМЫХ КОЛЕБАНИЙ ТРАЕКТОРИЯ"
1140 PRINT "РЕЗУЛЬТИРУЮЩЕГО ДВИЖЕНИЯ ПРИМЕТ ВИД ЗАМЫСЛОВАТЫХ "1
1150 PRINT "КРИВЫХ, ИЗВЕСТНЫХ ПОД НАЗВАНИЕМ ФИГУР ЛИССАЖУ."
1160 LOCATE 3,13
1170 PRINT "НАЧИНИТЕ ИССЛЕДОВАНИЕ ДЛЯ A1=A2=1, T1=2, T2=3, F1=F2=0."
1180 LOCATE 0,15
1190 INPUT "ВВЕДИТЕ АМПИЛИТУДЫ A1 И A2":A1,A2
1200 INPUT "ВВЕДИТЕ ПЕРИОДЫ T1,T2":T1,T2
1210 PRINT "ВВЕДИТЕ НАЧАЛЬНЫЕ ФАЗЫ F1,F2 "1
1220 INPUT "(РАЗНИЦА КРАТНА ПИ/2)":F1,F2
1230 IF A1<1 THEN A1=1
1240 IF A1>2 THEN A1=2
1250 K=T1/T2: M=A2/A1
1260 SCREEN 2: OPEN "GK": AS#1
1270 DRAW "BMS,100R250MH353"
1280 DRAW "BM110,100U180NE3F3"
1290 FOR I=1 TO 4 STEP .01
1300 X=FN(1)*K+0+128
1310 Y=MFNA(I)*K+0+106
1320 PSET(X,Y)
1330 NEXT I
1340 ON KEY GOSUB 250,1370: KEY(1) ON: KEY(2) ON
1350 PSET (50,195): PRINT#1,"F1 - меню F2 - выход": CLOSE#1
1360 GOTO 1360
1370 KEY1="COLOR": KEY2="AUTO": CLS: END

```

Л И Т Е Р А Т У Р А

1. Уолш Б. Программирование на Бейсике. - М.: Радио и связь. 1987.
2. УАМАНА. Справочное руководство по языку программирования Бейсик для комплектов учебной вычислительной техники на базе персональных компьютеров "Ямаха MSX-2". УАМАНА CORPORATION. 88 03 CP Printed in Japan.
3. Степанов М.Е. Практические занятия на ПЭВМ ЯМАХА. - М.: изд. школ МИ РОФСР, 1987.
4. Блэнд Г. Основы программирования на языке Бейсик в стандарте MSX. - М.: финансы и статистика, 1989.
5. Компьютерные игры / Под ред. М.Б. Игнатьева. - Л.: Ленивдат, 1988.
6. Грабов Д. Графические средства персонального компьютера. - М.: МПР, 1989.

Указатель операторов и функций

ASC	35
BEEP [#:p]	53
CHR \$ I9	
CIRCLE [c:k:l]	9
COLOR [k:l:c]	4
DRAW [d:r:l]	12
INKEY \$ [':n:k:e:l]	32
INPUT \$ [':n:p'u:t]	34
LEFT \$	35
LEN	35
LINE [l:a:i:n]	7
MID \$ [m:c:d]	36
ON x GOTO	37
PAINT [p:a:i:n:t]	12
PLAY [p'l'e:l]	37
PRESET	5
RESET	5
PUT SPRITE	19
SCREEN [s:c:r:e:n]	3
SOUND [s'a:u:n:d]	52
SPRITE ON	29
SPRITE OFF/STOP	30
SPRITE \$	18
STEP [s'te:p]	6
STICK(0)	25
STRIG(0)	31
STRING \$	25
VAL	33

СОДЕРЖАНИЕ

Предисловие	3
1. Средства отображения графической информации	3
1.1. Графические экраны	3
1.2. Основные графические операторы	5
1.2.1. Оператор RESET	5
1.2.2. Оператор LINE	7
1.2.3. Оператор CIRCLE	9
1.2.4. Оператор PAINT	12
1.2.5. Оператор DRAW	12
1.3. Спрайты	17
1.3.1. Описание размера спрайта	17
1.3.2. Описание спрайта	18
1.3.3. Воспроизведение спрайта на экране	19
1.3.4. Управление перемещениями спрайтов	25
1.3.5. Управление столкновениями спрайтов	29
1.3.6. Ввод - вывод информации в графическом режиме	31
2. Изобразительные средства мультимедиа	37
MSX-Бейсика	37
2.1. Команда PLAY	37
2.2. Кодирование музыкальной нотации	37
2.3. Программирование огибающих	40
2.4. Регулировка темпа и громкости	42
2.5. Программирование многоголосия	44
2.6. Специальные звуковые эффекты	50
2.7. Использование переменных и подстрок	53
2.8. Задачи для самостоятельной работы	54
2.9. Композиция музыки компьютером	63
2.10. Обучающая программа по музыке	65
3. Учебные компьютерные игры	74
3.1. Игровая программа "Вычитание чисел"	74
3.2. Игровая программа "Формулы приведения"	77
3.3. Игровая программа "Исследование треугольника, заданного координатами своих вершин"	79

3.4. Контролирующая программа с игровыми элементами по теме "Видоременные формы глаголов"	85
4. Имитационное моделирование	90
4.1. Исследование плавности тела под действием сил тяжести	91
4.2. Исследование гармонических колебаний	93
Литература	97
Указатель операторов и функций	98

Владимир Александрович Тарасов

Графические и музыкальные
возможности MSX-Бейсика

Редактор В.А.Ульянова

Сдано в набор
Формат 60x84 1/16 Заказ 516 Тираж 1500.
Цена 90 к.

Ротапринт Карелупрестат. Петрозаводск.