

0011 0010 1010 1101 0001 0100 1011

Графика в QBasic

1
2
4
5

В QBASIC существуют специальные графические операторы для создания изображений, но они требуют переключения в другой режим работы экрана.

Для включения графических режимов используется оператор **SCREEN**.

При переключении режимов очистка экрана производится автоматически.

Пример: оператор **SCREEN 2**

устанавливает графический режим экрана, содержащий 640 точек по горизонтальной оси и 200 точек по вертикальной. Изображение создается точками, и их положение на экране определяется координатной системой. Начало системы координат, т.е. точка с координатами (0,0), расположена в верхнем левом углу экрана. Последняя точка в этом графическом режиме имеет координаты (639,199) и находится в нижнем правом углу.

Для рисования картинки понадобятся точки, линии и окружности. В QBASIC для изображения этих графических примитивов используются соответствующие операторы:

- **PSET, PRESET** - рисование точки;
- **LINE** - рисование отрезка;
- **CIRCLE** - рисование окружности.

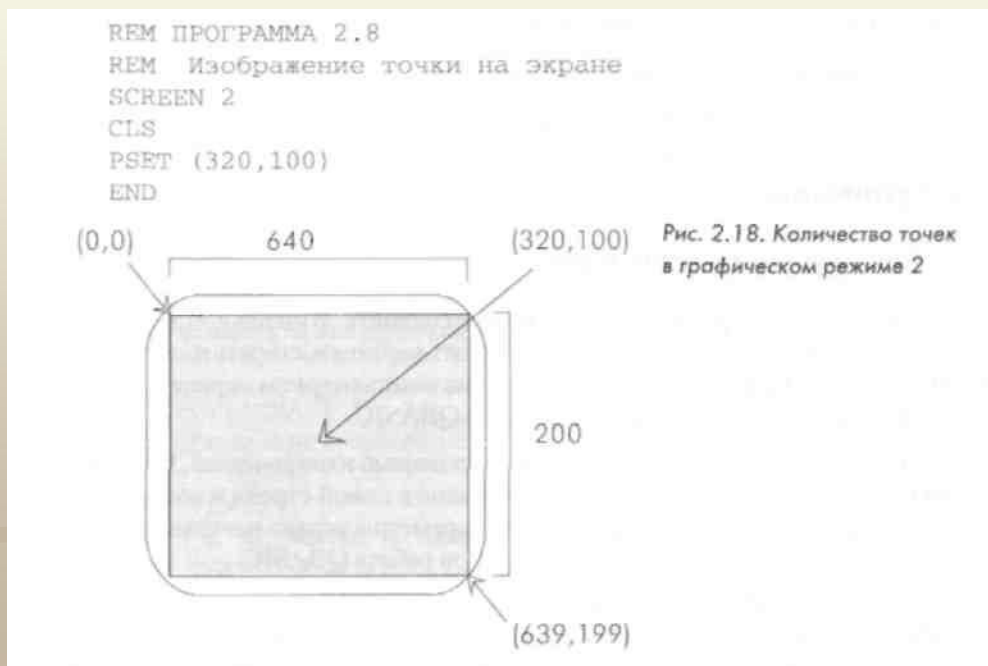


Рис. 2.18. Количество точек в графическом режиме 2



Операторы PSET и PRESET

Оператор **PSET** предназначен для рисования точки на экране путем изменения ее цвета с фонового (черного) на белый. Оператор может иметь следующие формы:

PSET (X, Y) — абсолютная форма

PSET STEP (X, Y) — относительная форма, где

X, Y — абсолютные координаты или смещение точки относительно ТПС.

Кроме того, в QBASIC существует оператор **PRESET**, имеющий аналогичную форму записи:

PRESET (X, Y) - абсолютная форма

PRESET STEP (X, Y) - относительная форма, где

X, Y — абсолютные координаты или смещение точки относительно ТПС.

Пример: В следующей программе показано использование этих операторов:

```
0011 0010
REM ПРОГРАММА 2.9
REM Применение операторов PSET и PRESET
SCREEN 2
CLS
' Изменение цвета 4 точек
PSET (310,90)
PSET (330,90)
PSET (330,110)
PSET (310,110)
' Обратное изменение цвета тех же точек
PRESET (310,110)
PRESET (330,110)
PRESET (330,90)
PRESET (310,90)
' Повторное изменение цвета
PSET (310,90)
PSET (330,90)
PSET (330,110)
PSET (310,110)
END
```

Прямые линии — отрезки

Оператор **LINE** предназначен для рисования отрезка, соединяющего две произвольные точки экрана.

Отрезок может быть коротким или длинным, вертикальным, горизонтальным или диагональным.

Общая форма записи оператора следующая:

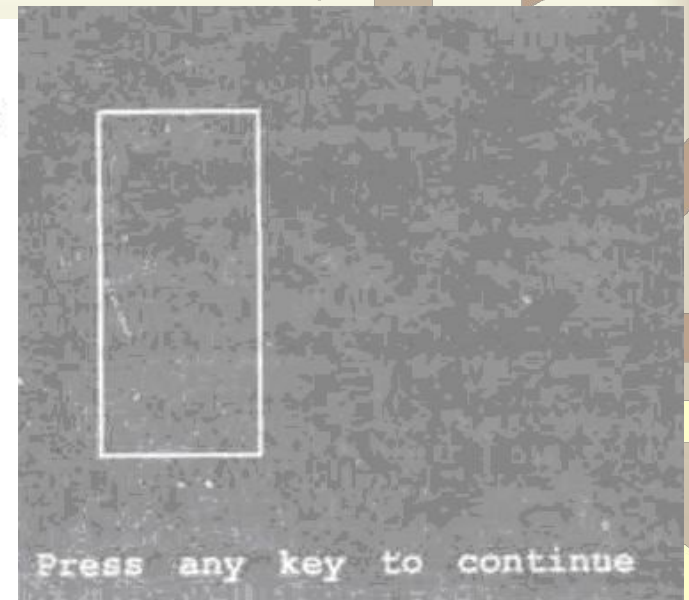
LINE (*X_начало*, *Y_начало*) - (*X_конец*, *Y_конец*), где

X_начало, *Y_начало* — координаты начала отрезка (необязательные параметры);

X_конец, *Y_конец* — координаты конца отрезка (обязательные параметры).

Пример: Запустите данную программу на выполнение и посмотрите результат. Первая команда **LINE** содержит все параметры - координаты начала и конца отрезка. В последующих операторах указаны только координаты конца отрезка. Поэтому каждый отрезок, начиная со второго, начинается в конечной точке предыдущего, образуя ломаную линию

```
REM ПРОГРАММА 2.10
REM Рисование простых отрезков
SCREEN 2
CLS
LINE (50,50) - (50,150)
LINE - (150,150)
LINE - (150,50)
LINE - (50,50)
END
```



В операторе **LINE** можно использовать относительные координаты для начала и/или конца отрезка.

Пример:

```
LINE STEP (10,-10) - (100,165)
LINE (115,120) - STEP (-10,15)
LINE STEP (-5,5) - STEP (5,-5)
```

Программу 2.10 можно переписать следующим образом, используя ТПС:

```
REM ПРОГРАММА 2.11
REM Модификация программы 2.10
'   Внимание!   ТПС = (320,100)
  SCREEN 2
  CLS
  LINE STEP(-270, -50)-STEP(0, 150)
  LINE -STEP(100, 0)
  LINE -STEP(0, -150)
  LINE -STEP(-100, 0)
  END
```

Задание: Попробуйте нарисовать дом или ваш компьютер.⁸

Рисование прямоугольников

Несколько другая форма записи оператора LINE позволяет рисовать прямоугольники:

LINE (*X_начало*, *Y_начало*) - {*X_диагональ*, *Y_диагональ*}, *цвет*, *B*

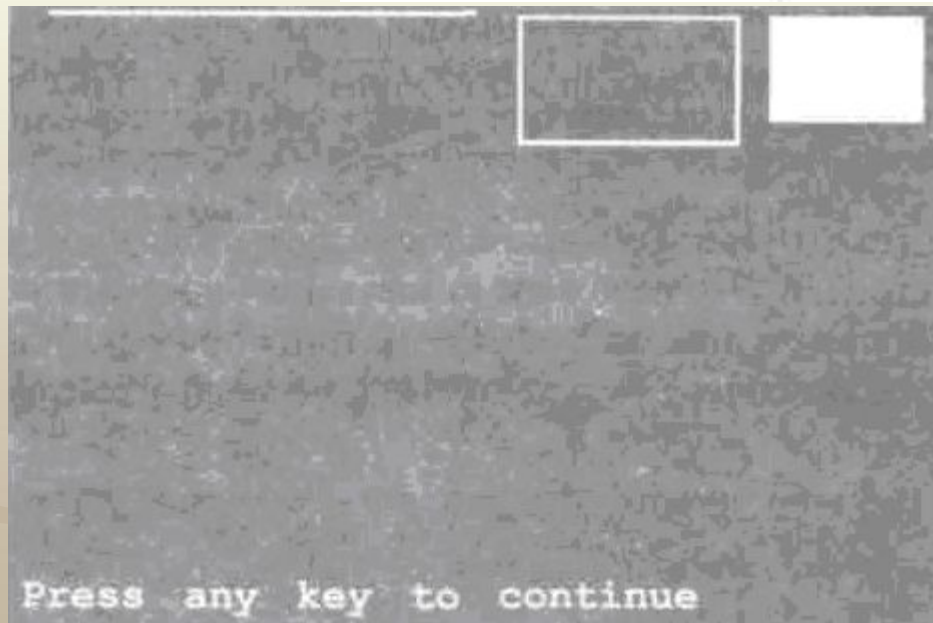
LINE (*X_начало*, *Y_начало*) - {*X_диагональ*, *Y_диагональ*}, *цвет*, *BF*

где

- *B* — параметр, указывающий на рисование прямоугольника;
- *BF* — параметр, указывающий на рисование закрашенного прямоугольника;
- (*X_начало*, *Y_начало*) — координаты левого верхнего угла прямоугольника;
- (*X_диагональ*, *Y_диагональ*) — координаты правого нижнего угла прямоугольника.

Пример:

```
REM ПРОГРАММА 2.14  
REM  Рисование прямоугольников  
SCREEN 1  
CLS  
LINE (10,20)-(135,20),3  
LINE (150,20)-(200,45),3,B  
LINE (230,20)-(280,45),3,BF  
END
```



4
5

Оператор CIRCLE

Оператор **CIRCLE** позволяет рисовать окружность в любом месте экрана. Посмотрите на формы записи оператора:

```
CIRCLE (X_центр, Y_центр) , радиус – абсолютная форма  
CIRCLE STEP (X_центр, Y_центр, радиус – относительная форма,  
где  
X_центр, Y_центр – координаты или смещение центра окружности;  
радиус – радиус окружности.
```

Пример: Следующая программа является примером использования данного оператора:

```
REM ПРОГРАММА 2.12  
REM Рисование окружности  
SCREEN 2  
CLS  
CIRCLE (100,100),25  
END
```



Полученная окружность с центром в точке с координатами (100,100) имеет радиус 25 точек.

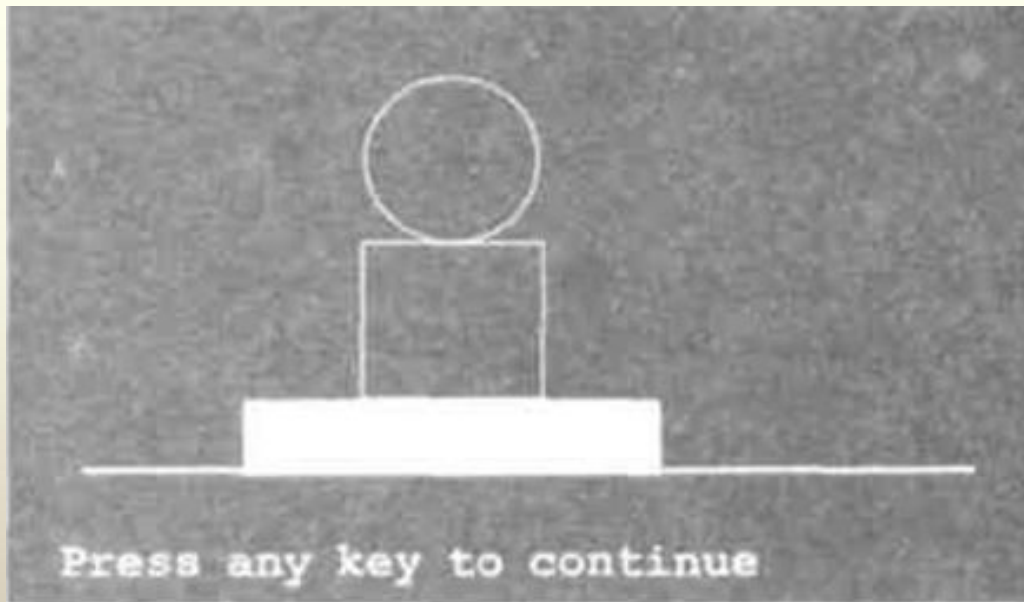
Задание:

1. Составьте программу рисования олимпийских колец.
2. Составьте программу рисования товарного знака фирмы AUDI.
3. Попробуйте запрограммировать свой автопортрет:



4. Составьте программу рисования композиции из следующих фигур:

0011 0010 1010 1101 0001 0100 1011



Рисование дуги, эллипса, сектора

Чтобы нарисовать дугу, эллипс или сектор окружности необходимо добавить новые параметры в оператор CIRCLE.

Полная форма записи оператора CIRCLE выглядит так:

CIRCLE (X,Y), радиус, цвет, начало, конец, коэффициент, где

X, Y — координаты центра окружности;

радиус — радиус окружности;

цвет — ее цвет;

начало — начальная точка дуги, заданная в радианах;

конец — конечная точка дуги, заданная в радианах;

коэффициент — отношение значений Y-радиуса и X-радиуса.

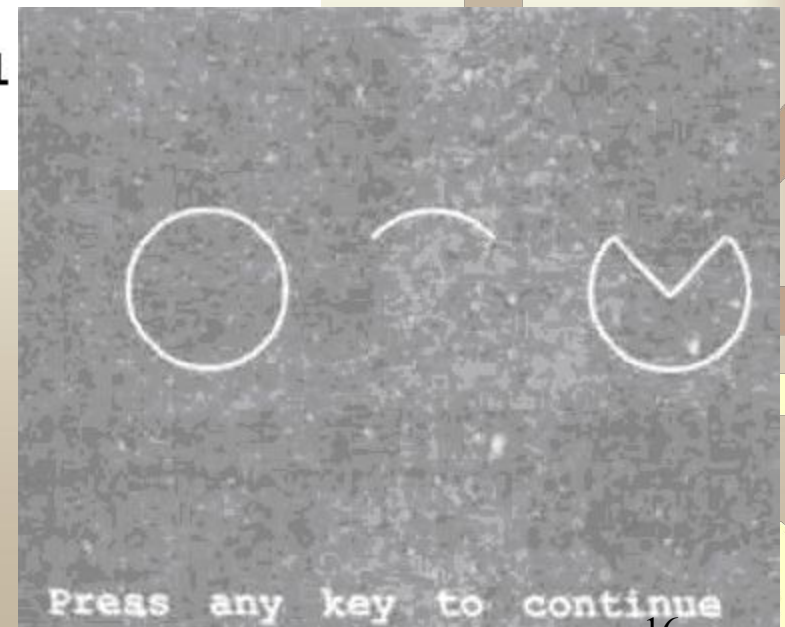
Дуга определяется углом, который вырезается из соответствующей окружности. Значения параметров *начало* и *конец* задаются в не градусах, а в радианах и должны иметь значения между 0 и 2π радиан ($2\pi = 6.28$).

Справка: Радиан — это единица для измерения величины углов, которую можно определить так: один радиан — это угол, вырезающий на окружности дугу, длина которой равна радиусу этой окружности. Радиан составляет 57.28835 градусов. Полная окружность содержит 360 градусов или 2π радиан, а половина (развернутый угол) — 180 градусов или π радиан, где $\pi = 3.1415$.

Пример: QBASIC при рисовании дуг ведется отсчет от начальной точки дуги к конечной в направлении против часовой стрелки.

0011 0010 1010 1101 0001 0100 1011

```
REM ПРОГРАММА 2.15
REM Рисование окружности, дуги и сектора
SCREEN 2
CLS
CIRCLE (100,100), 30
CIRCLE (180,100), 30, 3, 1, 2
CIRCLE (260,100), 30, 3, -2, -1
END
```

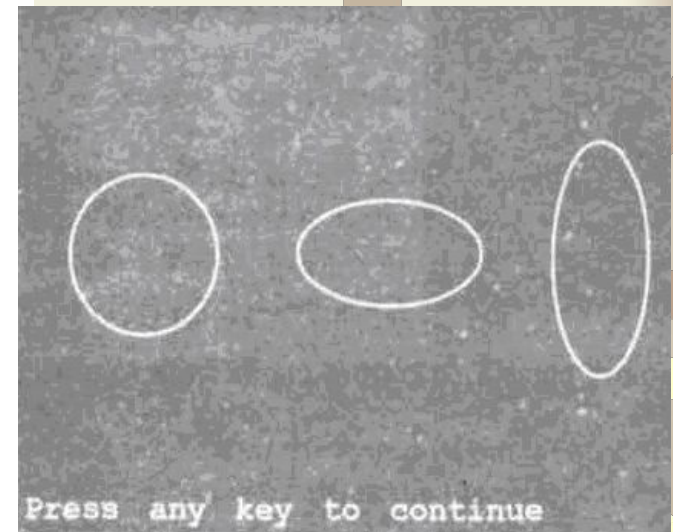


Для рисования эллипса необходимо ввести в оператор CIRCLE *коэффициент* отношения радиусов по осям Y и X. Этот параметр определяет степень сжатия эллипса и может иметь любое положительное значение.

Если параметр *коэффициент* опущен или равен 1, вы получаете изображение окружности.

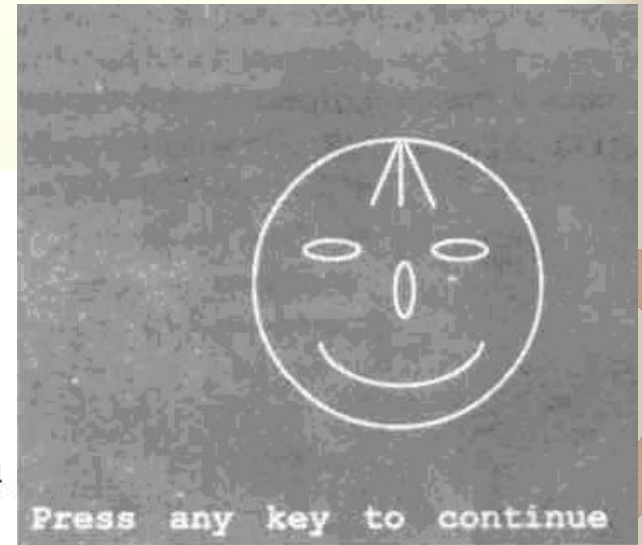
Пример:

```
REM ПРОГРАММА 2.16
REM Рисование эллипсов
SCREEN 2
CLS
CIRCLE (50,90), 30
CIRCLE (150,90), 30, , , , 0.3
CIRCLE (250,90), 30, , , , 1.5
END
```



Задание: Следующая программа представляет оператор CIRCLE с использованием различных параметров. Возможно, этот рисунок вдохновит вас на создание своих собственных произведений.

```
REM ПРОГРАММА 2.17
REM Рисование мордашки
SCREEN 1
CLS
CIRCLE (100,100), 70           'ЛИЦО
CIRCLE (75,75), 10, , , , .2  'глаза
CIRCLE (125,75), 10, , , , .2
CIRCLE (100,100), 5, , , , 2  'НОС
LINE (100,40)-(90,60)         'ВОЛОСЫ
LINE (100,40)-(110,60)
LINE (100,40)-(100,60)
CIRCLE (100,110), 20, , 3.14, 0 'улыбка
END
```



0011 0010 1010 1101 0001 0100 1011

Использование цвета



1
2
4
5

Программируя в QBASIC можно рисовать не только белые изображения на черном фоне. Существуют наборы различных цветов, которые можно использовать как для фона, так и для рисунка.

Цвет, который вы добавите в качестве параметра в операторы PSET, PRESET, LINE и CIRCLE, влияет только на изображение, оставляя фон без изменений.

PSET (X,Y), цвет

PRESET (X,Y), цвет

LINE (X_начало,У_начало) - (X_конец,У_конец), цвет

CIRCLE(X_центр,У_центр), радиус, цвет,

где **цвет** — значение цветового параметра.

В режиме экрана, задаваемом оператором **SCREEN 2**, возможны только два цвета: черный и белый.

Однако режим **1** поддерживает 4 цвета, которым соответствуют значения от 0 до 3 :

значение параметра	цвет
0	черный (фон)
1	голубой
2	пурпурный
3	белый

Чтобы «стереть» какой-либо элемент изображения без очистки всего экрана, можно просто перерисовать этот элемент цветом фона

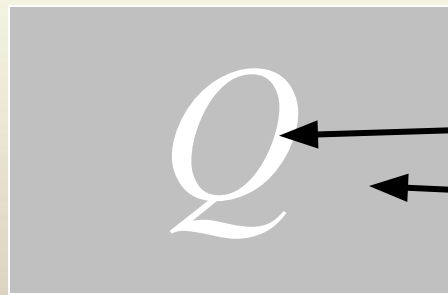
Номера цветов для режима SCREEN 7

<i>номер цвета</i>	<i>цвет</i>
0	<i>черный</i>
1	<i>голубой</i>
2	<i>зеленый</i>
3	<i>бирюзовый</i>
4	<i>красный</i>
5	<i>сиреневый</i>
6	<i>коричневый</i>
7	<i>белый</i>
8	<i>серый</i>
9	<i>светло –голубой</i>
10	<i>светло –зеленый</i>
11	<i>светло –бирюзовый</i>
12	<i>светло –красный</i>
13	<i>светло –сиреневый</i>
14	<i>желтый</i>
15	<i>ярко - белый</i>

Пример: Данная программа рисует две окружности, а затем стирает их изображение. Проверьте:

```
REM ПРОГРАММА 2.13
REM  Стирание изображения
SCREEN 1
CLS
CIRCLE (100,90),30,1      'рисование
CIRCLE (100,100),30,1    'рисование
CIRCLE (100,100),30,0    'стирание
CIRCLE (100,110),30,0    'стирание
END
```

При создании любого изображения программисты оперируют двумя понятиями: **foreground** –цвет рисунка(или переднего плана) и **background** – цвет фона. Цветом рисунка считается цвет каждой точки изображенного объекта. Цветом фона считается цвет всех других точек.



Цвет символа

Цвет фона



Оператор COLOR.

Различные графические режимы позволяют использовать различные наборы цветов. Каждый режим имеет свой набор цветов, называемый **палитрой**.

Способность монитора воспроизводить 16 цветов позволяет выбрать цвет фона в палитре для данного режима работы экрана. Эти изменения могут быть выполнены с помощью оператора **COLOR**.

Данный оператор в графическом режиме отличается от аналогичного оператора в текстовом режиме и имеет следующий вид:

COLOR фон, номер палитры

где

фон – код цвета для фона от 0 до 15;

номер палитры – номер палитры (0 или 1)

Оба эти параметра не обязательны, и команда может быть использована без параметров.

Рисуя цветные образы, включайте цветовые параметры в операторы рисования основных графических объектов:

LINE (X start, Y start) – (X end, Y end), Color

PSET (X, Y), Color

PRESET (X, Y), Color

CIRCLE (X center, Y center), Radius, Color

Оператор PAINТ

0011 0010 1010 1101 0001 0100 1011

Можно улучшить изображение объекта (закрашиваемые области должны иметь замкнутый контур), раскрасив его в различные цвета, используя оператор PAINТ:

PAINТ (X, Y), краска, контур

где

X, Y – координаты любой точки внутри области;

краска – номер цвета для закрашивания;

контур – номер цвета для контура области.

Оператор **PAINT** работает так:

- 1) Процесс закрашивания начинается в точке с координатами (X, Y) и распространяется по всем направлениям. При этом изменяется цвет каждой точки внутри контура.
- 2) В режиме 1 параметры оператора могут принимать значения от 0 до 3. Если значение не распознается или больше 3, по умолчанию будет выбрано 3.
- 3) Если значение присваивается параметру *краска*, а параметр *контур* опущен, то контур будет окрашен цветом *краска*.

Задание: Составьте программу рисования с использованием операторов разукрашивания.

0011 0010 1010 1101 0001 0100 1011



Оператор DRAW

В QBasic существует графический оператор **DRAW**, с помощью которого создают изображения иным способом и который дает такие возможности как движение и поворот рисунка.

Цвет в операторе **DRAW** может быть определен с помощью команды **CP**. Аргумент **n** должен иметь значение, актуальное для установленного графического режима. Цвет ранее нарисованных фигур по этой команде не изменится.

Освоив методы рисования различных изображений, можно попробовать добиться эффекта мультипликации: создавать многократные копии изображения или двигать графическую картинку³⁰.

Команды относительного движения позволяют перемещаться в любом из восьми направлений.

Каждая из этих команд сопровождается целочисленным аргументом, указывающим длину линии в точках. Например,

Следующие команды также относятся к командам относительного движения:

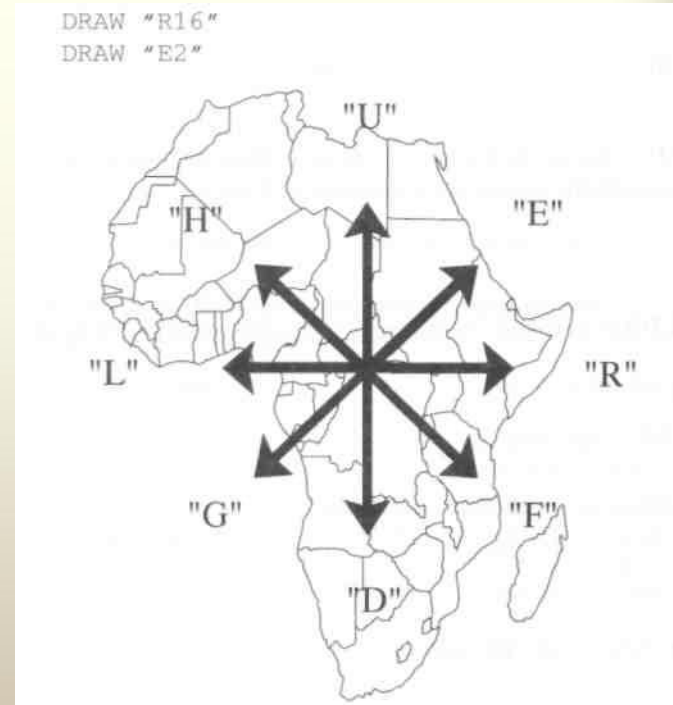
V — перемещение без рисования;

N — перемещение без изменения значения LPR.

Например:

`DRAW "NM8,29"`

Это означает, что можно установить новое значение для LPR без рисования линии (используя команду V) и рисовать линию без изменения прежнего значения LPR (используя команду N). Данные команды дополняют друг друга.



REM ПРОГРАММА 6.17

REM Использование команды M

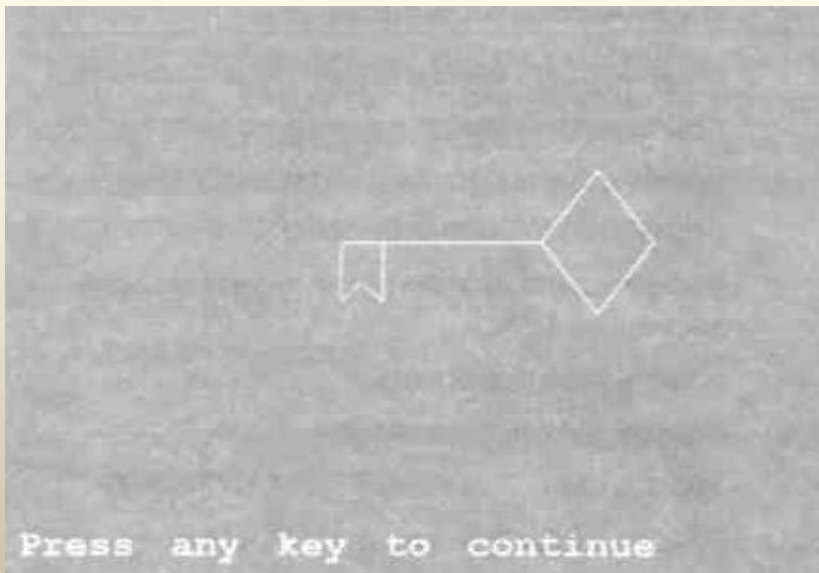
SCREEN 1

' рисование ключа с помощью команды M

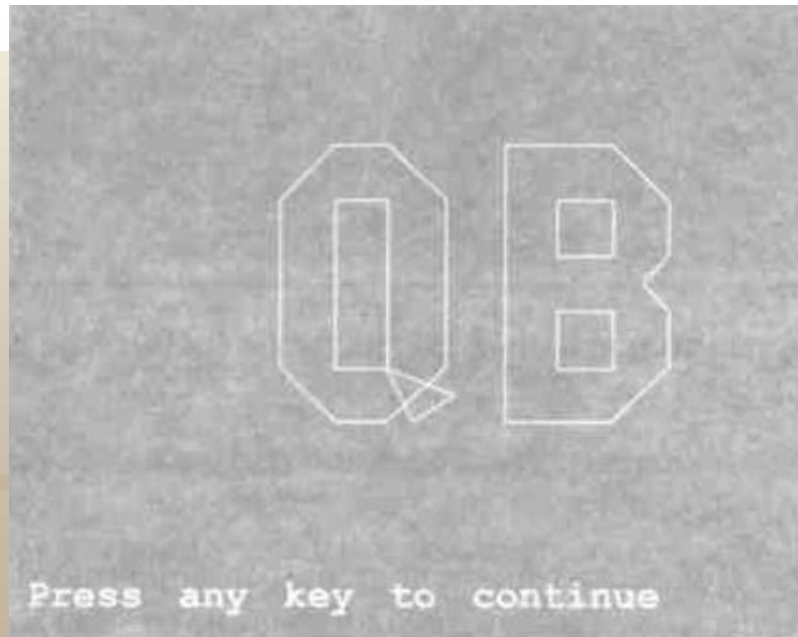
DRAW "M+25,+25 M+25,-25 M-25,-25 M-25,+25 M -60,+0"

DRAW "M+0,+20 M+10,-10 M+10,+10 M-0,-20"

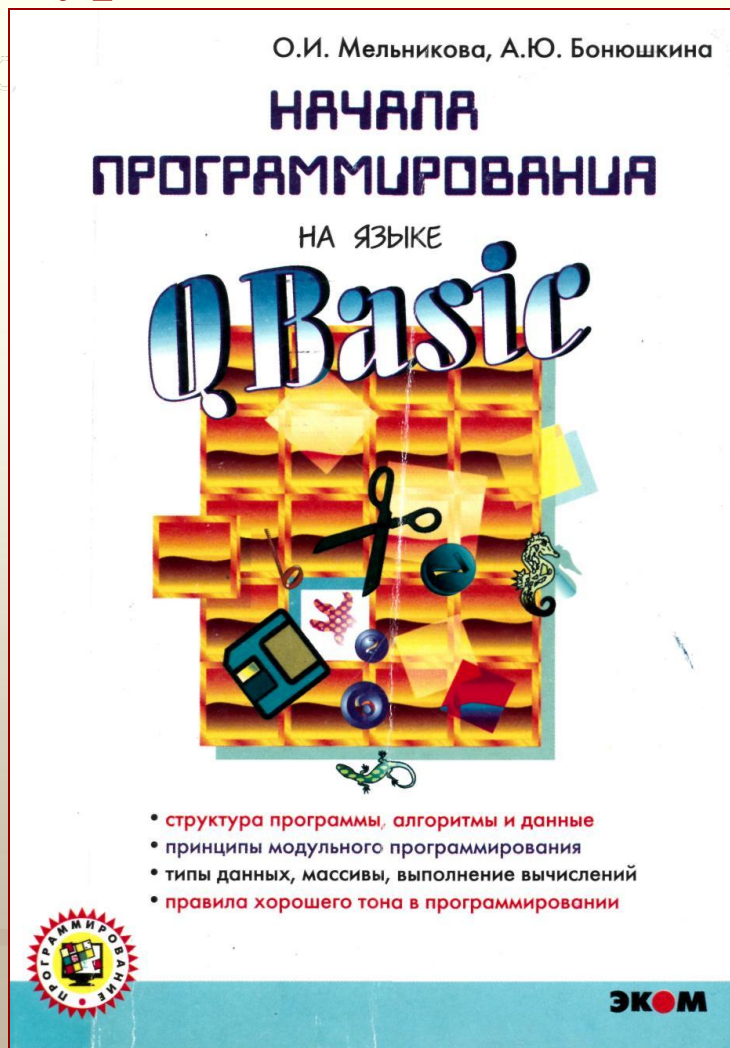
END




```
REM ПРОГРАММА 6.18
REM Использование команд относительного движения
SCREEN 1
CLS
' рисование буквы "Q"
DRAW "BM 100,100 F10 R10 E10 U30 H10 L10 G10 D30"
DRAW "BM+10,+0 R10 U30 L10 D30"
DRAW "BM+10,+0 NM125,112 N M135,110 BM125,112 M 135,110"
' рисование буквы "B"
DRAW "BM+10,+0 R20 E10 U10 H5 E5 U10 H10 L20 D50"
DRAW "BM+10,-10 R10 U10 L10 D10"
DRAW "BM+0,-20 R10 U10 L10 D10"
' конец рисования
END
```



Литература:



1 2
4 5