

ОСНОВЫ ПРОГРАММИ- РОВАНИЯ

Что такое алгоритм

**Решение даже самой
простой задачи
обычно
осуществляется за
несколько
последовательных
шагов.**

Процесс покупки хлеба можно представить так:

- Взять у мамы денег;**
- Пойти в магазин;**
- Выбрать нужные
хлебобулочные изделия;**
- Оплатить стоимость покупки;**
- Принести хлеб домой.**

В виде последовательности действий можно описать процессы решения многих задач, с которыми вы имеете дело в школе:

«Вычислить периметр многоугольника»,

«Найти наибольший общий делитель двух натуральных чисел»,

«Определить часть речи»,

«Провести фонетический разбор слова».

**Алгоритм –
последовательность
шагов в решении
задачи**

**Для алгоритма
важен не только
набор действий, но
и то, в каком
порядке они
выполняются.**

Попробуем переставить в известном вам алгоритме нахождения наименьшего общего кратного (НОК) нескольких натуральных чисел четвертое действие на второе место:

1. Разложить исходные числа на простые множители
2. Вписать в звездочку все простые множители разложения одного из чисел
3. Дописать множители недостающие множители из разложений остальных чисел
4. Дописать произведение недостающих множителей из разложений остальных чисел

30

42

2 × 3 × 5

2 × 3 × 7

2 × 3 × 5 × 7

210

1. Разложить исходные числа на простые множители
2. Вписать в выведенный порядок множителей, разложенный одного из чисел
3. Дописать множители, недостающие множители из разложений остальных чисел
4. Дать произведение полученных множители из разложенных остальных чисел

Эту последовательность действий также можно исполнить, но к достижению поставленной цели (нахождению НОК) она не приведет!

**Выполним последовательность действий
для нахождения значения следующего
выражения:**

$$(5,88+5,52)-2,8 : (5 \cdot 0,103 - 0,015)$$

$$1) 5,88+5,52=11,4$$

$$2) 5 \cdot 0,103=0,515$$

$$3) 0,515 - 0,015=0,5$$

$$4) 2,8 : 0,5 = 1,4$$

$$5) 11,4 - 1,4 = 10$$

**Можно ли изменить порядок действий в этом
случае?**

С чем, по вашему мнению, это связано?

**Перед составлением алгоритма
должны быть четко определены
начальные условия
и
то, что предстоит получить.**

Алгоритм –

конечная

последовательность

шагов в решении

задачи, приводящая от

исходных данных к

требуемому результату.

Разрабатывать
алгоритм
может только
человек.



Исполняют алгоритмы



люди и

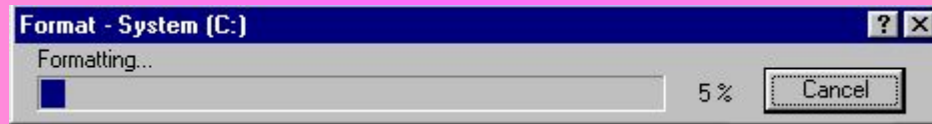


всевозможные устройства

– компьютеры, роботы,
станки, спутники, сложная
бытовая техника ...



Исполнители вокруг нас

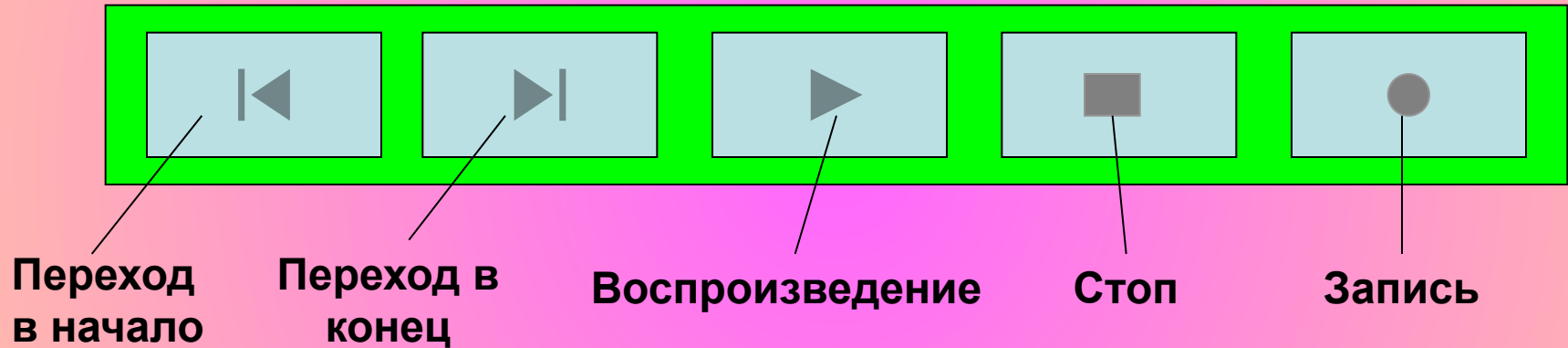


**Исполнитель -
устройство,
способное выполнять
определённый набор
команд.**



**Система команд
исполнителя (СКИ) -
команды, которые
может выполнить
конкретный
исполнитель.**

Пример: Система команд исполнителя-магнитофона содержит следующие команды:



Во многих случаях и сам человек является исполнителем алгоритмов.

Например, каждый из нас при переходе улицы является исполнителем следующего алгоритма:

1. *остановись на тротуаре;*

2. *посмотри налево;*



3. *если транспорта нет, то иди до середины улицы, иначе выполняй п.2;*

4. *посмотри направо;*

5. *если транспорта нет, то иди до противоположного тротуара, иначе выполняй п.4;*



Формы записи алгоритмов

**Алгоритм —
последовательность
шагов, которые
выполняются
человеком при решении
задач, можно записать в
виде списка, таблицы
или БЛОК-СХЕМЫ.**

БЛОК-СХЕМЫ –

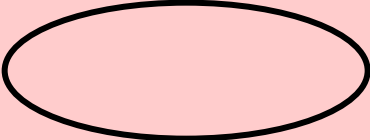

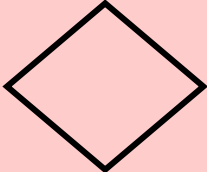

это форма записи

алгоритма в виде

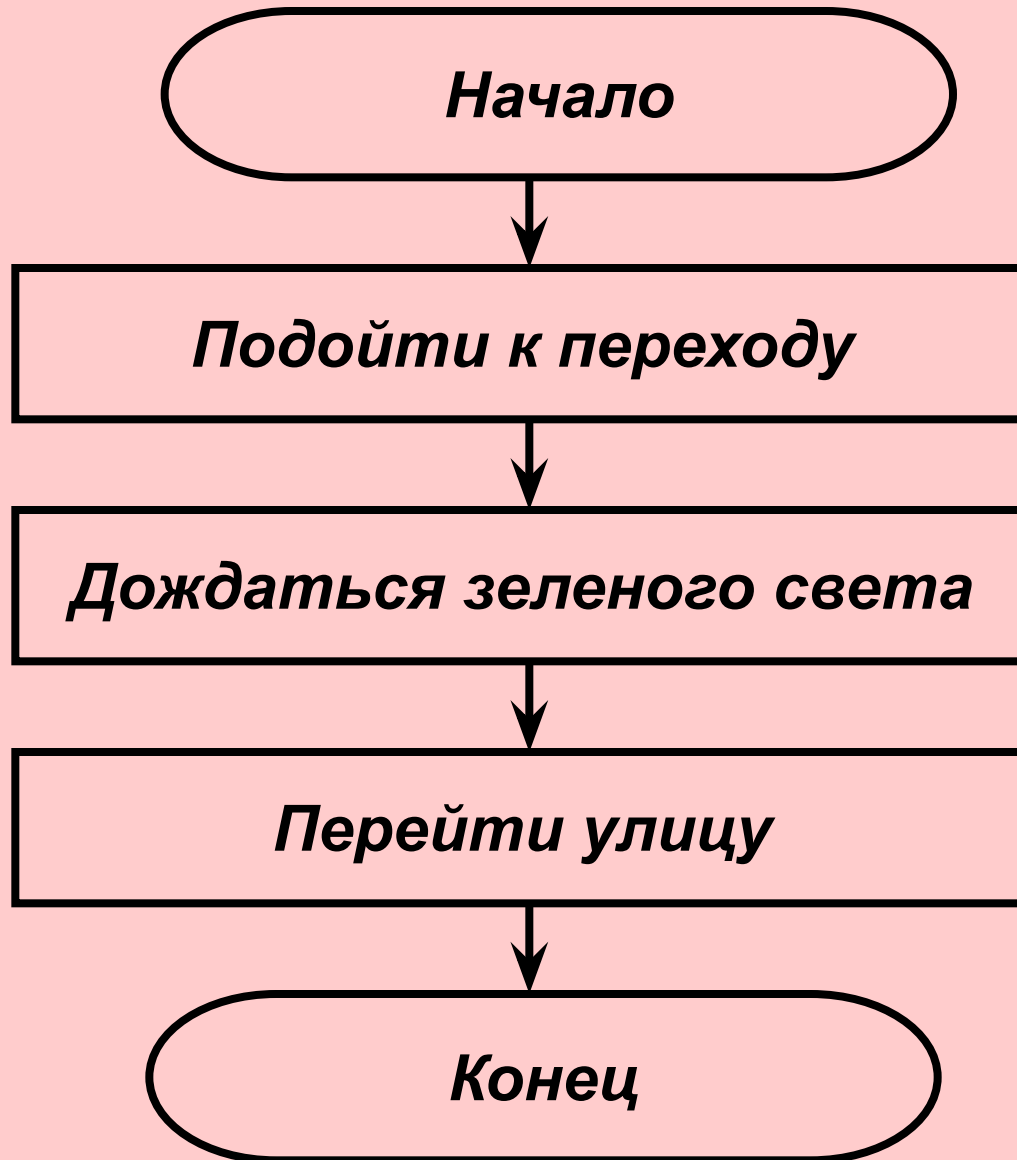
геометрических фигур и

стрелок, указывающих

порядок действий.

Название фигуры	Изображение	Обозначаемый шаг алгоритма
Овал		Начало и конец
Параллелограмм		Ввод или вывод
Ромб		Принятие решения
Прямоугольник		Выполнение действия

Алгоритм действий человека при переходе через улицу



Типы алгоритмов

Типы алгоритмов



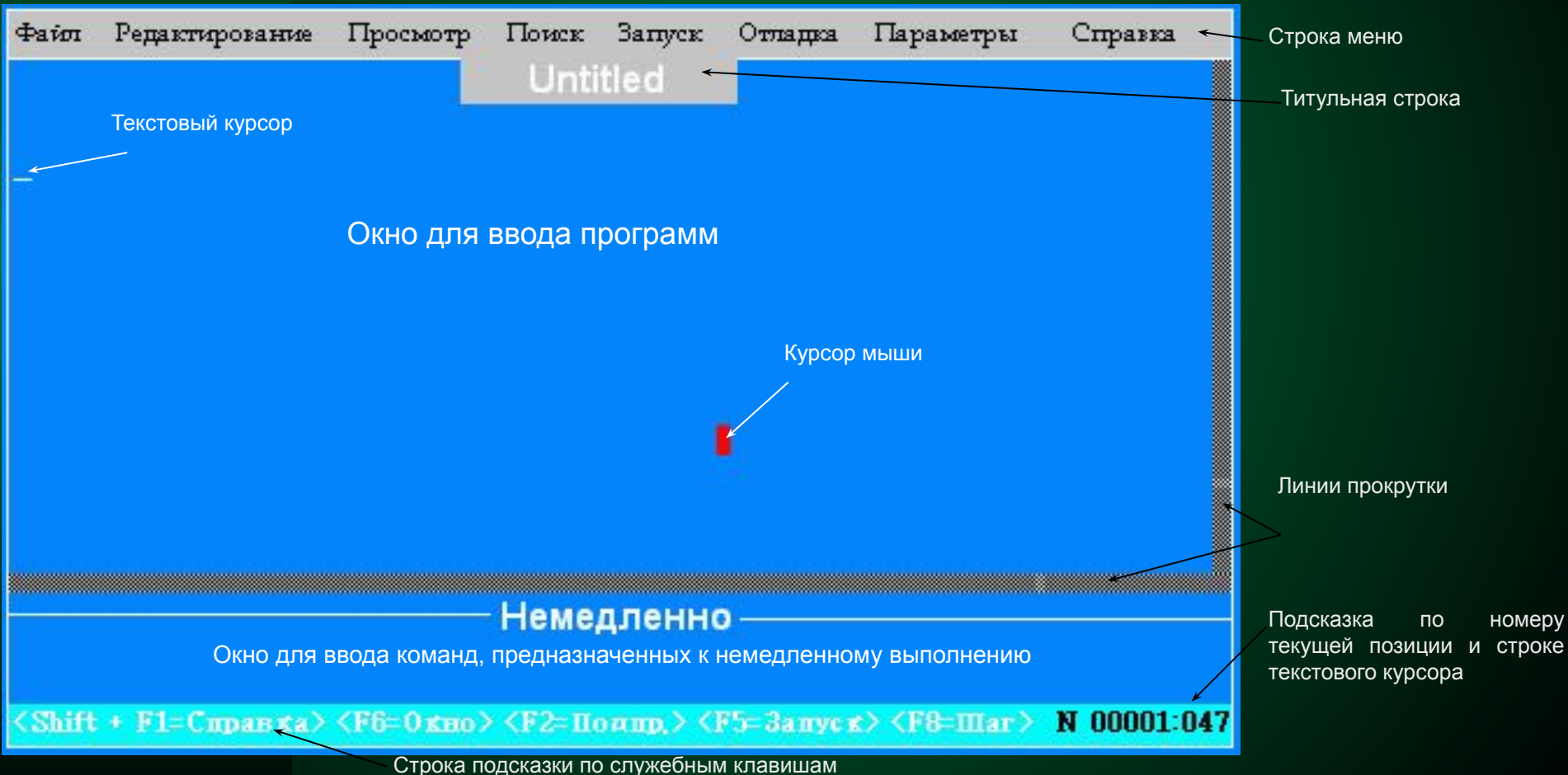


Язык программирования

QBASIC



Графический интерфейс





Алфавит Qbasic:

1. латинские буквы (A – Z, a – z).
Расширенные ASCII-коды
(русские буквы) допускается
использовать только в
символьных константах и
примечаниях.
2. цифры (0 – 9);
3. знаки логических операций;
4. специальные символы.



Программа –



последовательность команд, реализующих тот или иной алгоритм.

Оператор –

минимальная конструкция языка, задающая описание некоторого действия.



Переключение алфавитов в QBasic

В английский

В русский

Левый Shift +
Правый Ctrl

Левый Ctrl +
Правый Shift



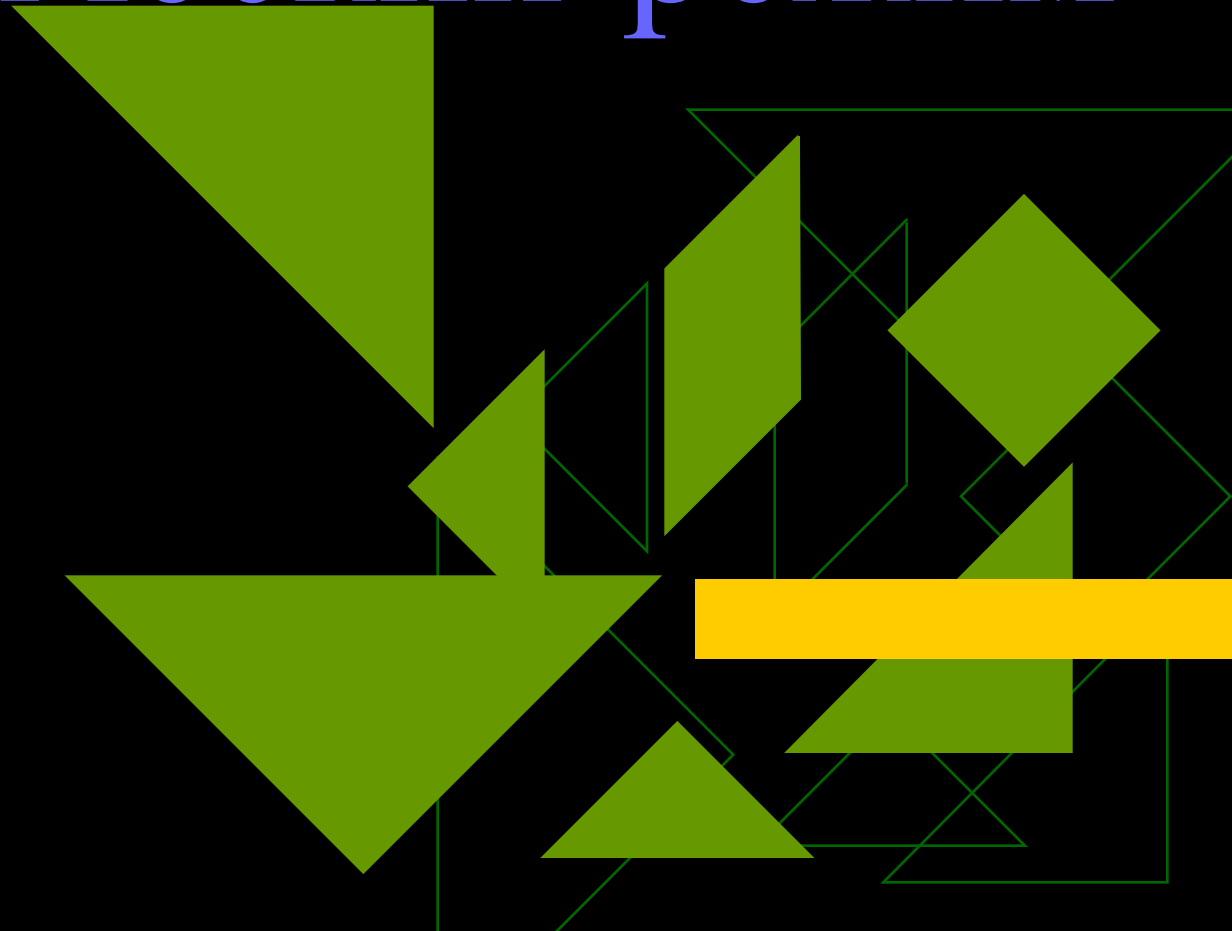
Рабочие клавиши

F4 – просмотр экрана вывода (результата работы программ).

F5 – запуск программы, загруженной в окно программы, на выполнение.

F6 – переключение между окнами.

Графический режим



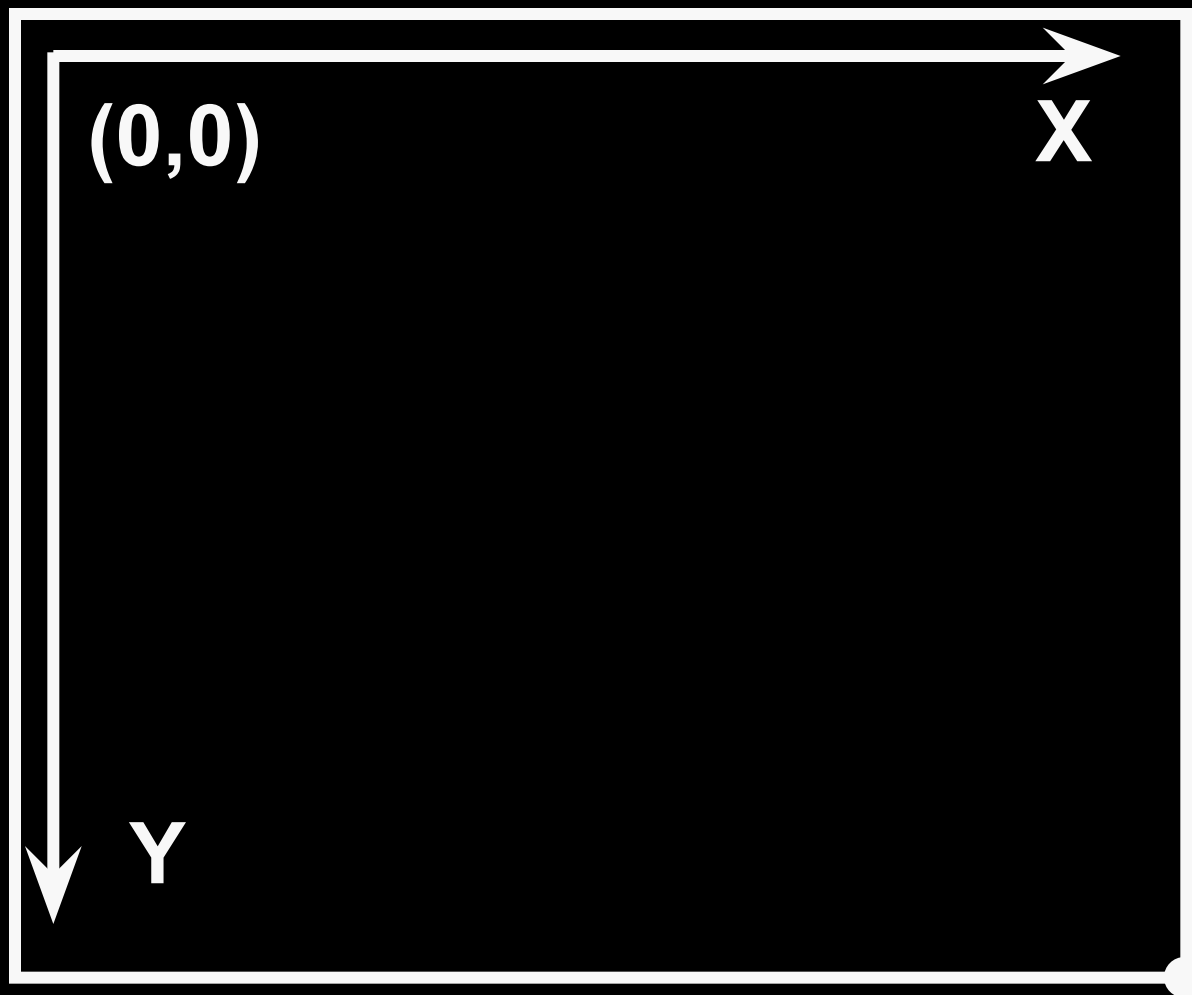
Загрузка графического режима

SCREEN $_n$

Здесь n – номер графического режима

- 7 режим 320 × 200 пиксель, цветной
- 9 режим 640 × 350 пиксель, цветной
- 12 режим 640 × 480 пиксель (квадратный),
цветной

Экран графического режима



(319,199) 7 режим
(639,349) 9 режим
(639,479) 12 режим

Основные операторы



Оператор точки

PSET _ (X,Y)[,C]

Здесь и далее:

(X,Y) – координаты точки;

C – номер цвета из палитры цветов.

Палитра цветов

0	– черный	8	– серый
1	– синий	9	– ярко-синий
2	– зеленый	10	– ярко-зеленый
3	– голубой	11	– ярко-голубой
4	– красный	12	– ярко-красный
5	– лиловый	13	– ярко-лиловый
6	– коричневый	14	– желтый
7	– белый	15	– ярко-белый

(по умолчанию)

Пример

**Зажечь точки красным, желтым
и зеленым цветом.**

SCREEN 12

PSET (20,20), 4

PSET (100,100),14

PSET (200,200),2

Решение

Вычерчивание отрезка

LINE $_ _ (X_H, Y_H) - (X_K, Y_K) [, C]$

Здесь:

X_H, Y_H – координаты точки начала отрезка;

X_K, Y_K – координаты точки конца отрезка.

Пример

**Построить отрезки красным,
желтым и зеленым цветом.**

SCREEN 12

LINE (20,20)-(50,50), 4

LINE (100,100)-(50,50), 14

LINE (20,20)-(100,100), 2

Решение

Вычерчивание отрезка

LINE $_$ $-(X_k, Y_k)[, C]$

Здесь:

X_k, Y_k – координаты точки конца отрезка.

Началом считается текущая точка

Пример

Построить треугольник с
разноцветными сторонами.

SCREEN 12

LINE (120,20) – (200,200), 4

LINE –(100,100),14

LINE –(120,20),2

Решение

Вычерчивание прямо- угольного контура

LINE \perp $(X_n, Y_n) - (X_k, Y_k)$, [C],
B

Пример

**Построить лиловый
прямоугольный контур**

Решение

SCREEN 12

LINE (20,20)-(250,250), 5,B

Пример

**Построить прямоугольный
контур текущим цветом**

Решение

SCREEN 12

LINE (20,20)-(250,250),,B

Вычерчивание закрашенного прямо-угольного контура

LINE \perp (Xн, Yн)–(Xк, Yк), [C], BF

Пример

**Построить серый закрашенный
прямоугольник**

Решение

SCREEN 12

LINE (20,20)-(250,250), 8,BF

Оператор заливки замкнутого контура

PAINT (X,Y)[,[C1]][,[C2]]

Здесь:

C1 – цвет заливки;

C2 – цвет ограничивающего контура.

Если C2 отсутствует, то полагается, что C1=C2.

Пример

**Построить синий треугольник,
залить его серым цветом**

Решение

```
SCREEN 12
```

```
LINE (120,20)-(200,200), 9
```

```
LINE -(100,100),9
```

```
LINE -(120,20),9
```

```
PAINT (105,100),8,9
```

Оператор окружности

CIRCLE \sqsubset (X,Y),R[,C,U_н,U_к,A]

Здесь:

(X,Y) – координаты центра окружности;

R – радиус окружности;

C – цвет окружности;

U_н – начальный угол дуги окружности (по умолчанию 0);

U_к – конечный угол дуги окружности (по умолчанию 2π);

U_к

U_н

x

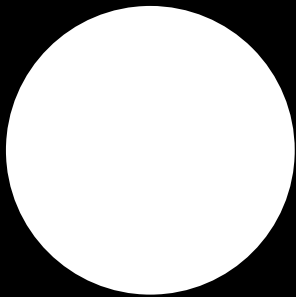
Оператор окружности

CIRCLE \sqsubset (X,Y),R[,C,U_н,U_к,A]

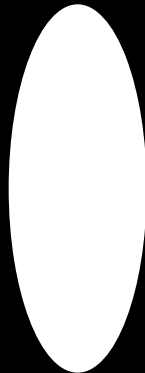
Здесь:

A – коэффициент «расплющивания» эллипса
(используется только при изображении эллипсов,
по умолчанию равно 1)

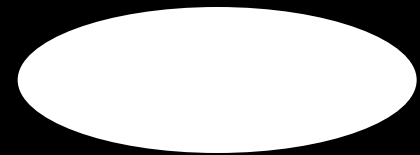
A = 1



A > 1



A < 1



Пример

Построить месяц

SCREEN 12

CIRCLE (300, 220), 100, 14

PAINT (300, 220), 14, 14

CIRCLE (400, 220), 150, 0

PAINT (400, 220), 0

Решение

Пример

Построить солнышко

Решение

```
SCREEN 12
CIRCLE (300, 220), 100, 14
PAINT (300, 200), 14
FOR i = 0 TO 6.28 STEP .15
LINE (300, 220)-STEP(200 * SIN(i), 200 * COS(i)), 14
NEXT i
```