

Графика в C++



Включение графики

Монитор ПК может работать в двух режимах текстовый и графический. В этих режимах по-разному представляется видео-память. Переход из режима в режим очищает экран.

Простейшая графическая программа

Графическая программа на Си имеет структуру сэндвича:



Для работы с библиотекой
графики ее необходимо
ПОДКЛЮЧИТЬ.

- ▶ `#include <graphics.h>`
- ▶ `#include <conio.h>`

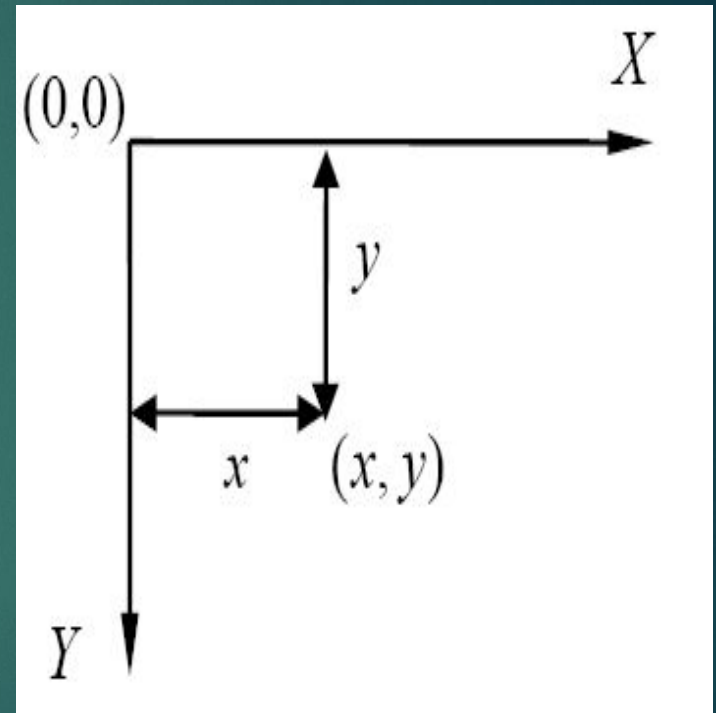
Простейшая графическая программа

Она открывает специальное окно для рисования, ждет нажатия клавиши и закрывает это окно. Программа эта так же неполноценна.

```
#include <graphics.h>
#include <conio.h>
main()
{
    initwindow ( 400, 300 ); // открыть окно для графики 400 на 300
    // ... здесь можно рисовать на экране («мясо»)
    getch(); // ждем нажатия клавиши
    closegraph(); // закрыть окно
}
```

Система координат

- ▶ Начало координат, точка $(0,0)$, находится в левом верхнем углу окна.
- ▶ Ось X направлена вправо, ось Y — вниз (в отличие от общепринятой математической системы координат).
- ▶ Для любой точки координата x — это расстояние до левой границы окна, а y — расстояние до верхней границы.



УСТАНОВИТЬ ЦВЕТ ДЛЯ РИСОВАНИЯ И ФОНА

- ▶ `setcolor(<цвет>)` - Установить цвет рисования
- ▶ `setbkcolor(<цвет>)` - Установить цвет фона

После установки цвета фона графическое окно нужно очистить цветом фона.

- ▶ `clearviewport();` - Очистить текущее окно цветом фона

16 СТАНДАРТНЫХ ЦВЕТОВ

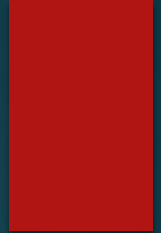
Код	Название	Какой цвет	Код	Название	Какой цвет
0	BLACK	черный	8	DARKGRAY	темно-серый
1	BLUE	синий	9	LIGHTBLUE	светло-синий
2	GREEN	зеленый	10	LIGHTGREEN	светло-зеленый
3	CYAN	морской волны	11	LIGHTCYAN	светлый морской волны
4	RED	красный	12	LIGHTRED	светло-красный
5	MAGENTA	фиолетовый	13	LIGHTMAGENTA	светло- фиолетовый
6	BROWN	коричневый	14	YELLOW	желтый
7	LIGHTGRAY	светло-серый	15	WHITE	белый

Полная палитра



- ▶ В этом случае цвет строится из трех составляющих: красной (**R**), зеленой (**G**) и синей (**B**).
- ▶ Каждая из этих составляющих – целое число от 0 до 255 (256 вариантов).
- ▶ Цвета строятся с помощью функции **COLOR**, у нее в скобках перечисляются через запятую значения составляющих **R**, **G** и **B** (именно в таком порядке).

Пример цветов



- ▶ `COLOR(0,0,0)` черный
- ▶ `COLOR(255, 0, 0)` красный
- ▶ `COLOR(0, 255, 0)` зеленый
- ▶ `COLOR(0, 0, 255)` синий
- ▶ `COLOR(255, 255, 255)` белый
- ▶ `COLOR(100, 100, 100)` серый
- ▶ `COLOR(255, 0, 255)` фиолетовый
- ▶ `COLOR(255, 255, 0)` желтый

ЛИНИИ

Отрезок можно нарисовать с помощью команды **line**:

▶ **line (x1, y1, x2, y2); // отрезок (x1,y1)-(x2,y2)**

Есть и другой способ: сначала перевести курсор (указатель текущего положения) в точку **(x1,y1)** командой **moveto**, а затем нарисовать отрезок в точку **(x2,y2)** командой **lineto**:

▶ **moveto (x1, y1); // курсор в точку (x1,y1)**

▶ **lineto (x2, y2); // отрезок в точку (x2,y2)**

После выполнения команды **lineto** курсор смещается в новую точку **(x2,y2)**.

УСТАНОВИТЬ СТИЛЬ И ТОЛЩИНУ ЛИНИЙ

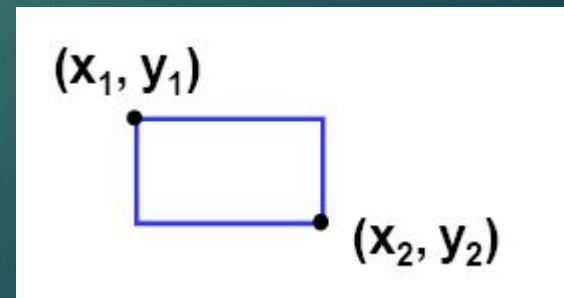
`setlinestyle(<тип линии>,1,<толщина линии>)`

Название	Знач.	Описание
SOLID_LINE	0	сплошная
DOTTED_LINE	1	пунктир
CENTER_LINE	2	штрихпунктир
DASHED_LINE	3	штриховая

```
setlinestyle(0,1,4);
```

Прямоугольник

- ▶ Для рисования прямоугольника необходимо задать координаты двух противоположных углов (обычно выбирают левый верхний и правый нижний углы). Цвет контура устанавливается с помощью функции **setcolor**, а сам прямоугольник рисуется командой **rectangle**:
- ▶ **setcolor (9);**
- ▶ **rectangle (x1, y1, x2, y2);**



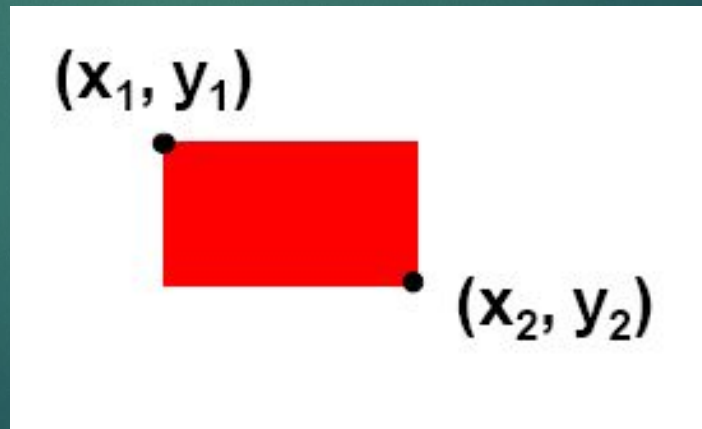
Закрашенный

прямоугольник

Закрашенный прямоугольник рисует команда **bar**.

Цвет и стиль заливки нужно заранее установить, вызвав функцию **setfillstyle**:

- ▶ `setfillstyle (1, 12);` // стиль заливки 1, цвет 12
- ▶ `bar (x1, y1, x2, y2);`



УСТАНОВИТЬ ЦВЕТ ДЛЯ ЗАЛИВКИ ФИГУР

- ▶ `setfillstyle(<стиль закрашки>, <цвет>)` Установить цвет и стиль закрашки фигур

Первое число в команде **setfillstyle** задает стиль заливки:

0 – отключить заливку

1 – сплошная заливка

3,4,5,6 – наклонные линии

7,8 – сетка

9,10,11 – точечные узоры

а второе – цвет.

Стили заливок



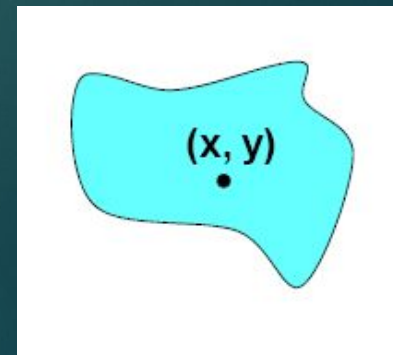
имя	значение	описание
EMPTY_FILL	0	заполнение цветом фона
SOLID_FILL	1	сплошное заполнение цветом заполнения
LINE_FILL	2	заполнение _____
LSTLASH_FILL	3	заполнение //
SLASH_FILL	4	заполнение /// толстой линией
BKSLASH_FILL	5	заполнение \\ толстой линией
LTBKSLASH_FILL	6	заполнение \\\
HATCH_FILL	7	светлая штриховка сеткой
XHATCH_FILL	8	крестообразная штриховка
INTERLEAVE_FILL	9	тканеобразная штриховка
WIDE_DOT_FILL	10	заполнение редкими точками
CLOSE_DOT_FILL	11	заполнение частыми точками

Заливка произвольной области

Иногда бывает нужно залить каким-то цветом произвольную область, ограниченную контуром одного цвета. Это можно сделать с помощью функции **floodfill**:

- ▶ **setfillstyle (1, 11); // стиль 1, цвет 11**
- ▶ **floodfill (x, y, 0); // до границы цвета 0**

Для заливки нужно знать координаты **(x,y)** одной (любой!) точки внутри этой области. Кроме того, нужно, чтобы граница области была одного цвета, без разрывов. Цвет границы указывается последним в списке данных, которые передаются функции **floodfill**



Окружность

Чтобы нарисовать окружность, используют функцию **circle**:

- ▶ **setcolor (COLOR(0,255,0)); // зеленый цвет**
- ▶ **circle (x, y, R);**

При вызове функции **circle** в скобках указывают координаты центра и радиус окружности в пикселях. Это могут быть числа, имена переменных или арифметические выражения, например:

- ▶ **circle (200, y0+20, R);**

Точка

putpixel(x,y,color); -ВЫВОДИТ ТОЧКУ ПО КООРДИНАТАМ
ЦВЕТА color.

3D рамка

bar3d(x1,y1,x2,y2, width, topflag);

width – боковая ширина,

число topflag задаёт наложение верха на рамку. Его обычно ставят 1.

bar3d(10,10,100,100,10,1);