

Графика в С

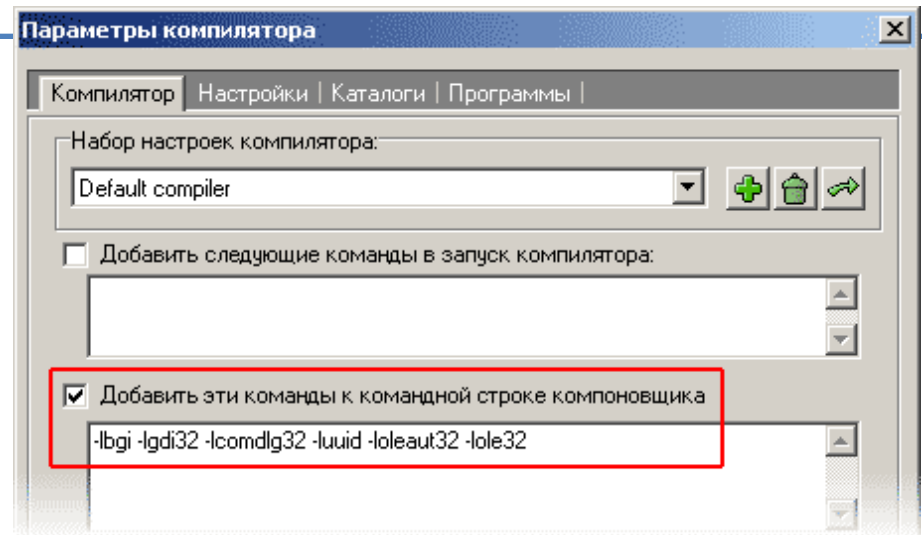
Семестр 1

Семинар 7

Windows, Dev C++

1. Скопируйте файл `graphics.h` в папку `C:\Dev-Cpp\include`.
2. Скопируйте файл `libbgi.a` в папку `C:\Dev-Cpp\lib`.
3. Запустите оболочку *Dev-C++* и войдите в меню *Сервис-Параметры компилятора*. Перейдите на вкладку *Компилятор*, включите флажок *Добавить эти команды к командной строке компоновщика* и добавьте в окно под этим флажком строчку

`-lbgi -lgdi32 -lcomdlg32 -luuid -loleaut32 -ole32`



Пример

```
#include <graphics.h>
```

```
int main()
```

```
{
```

```
    initwindow (400,300);
```

```
    // открыть окно для графики
```

```
    // размером 400 на 300 пикселей
```

```
    moveto (0,0);
```

```
    // курсор в точку (0,0)
```

```
    lineto (50,50);
```

```
    // отрезок в точку (50,50)
```

```
    getch();
```

```
    // ждать нажатия на любую клавишу
```

```
    closegraph ();
```

```
    // закрыть окно с графикой
```

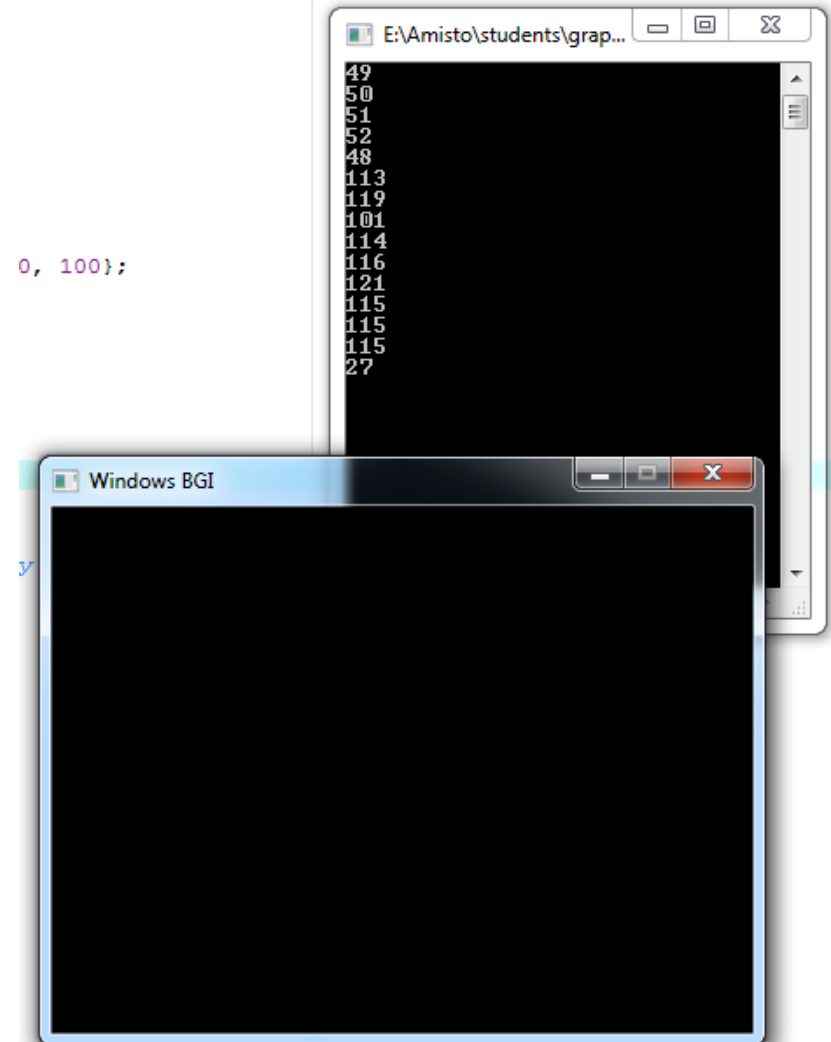
```
    return 0;
```

```
}
```

Консоль

```
#include <graphics.h>
int main()
{
    initwindow (400,300);
    char c = 0;
    while (c != 80)
    {
        c = getch();
        printf("%d\n", c);
    }
    closegraph ();
    return 0;
}
```

0, 100};



Прямые линии

```
#include <graphics.h>

int main()
{
    initwindow (400,300);
    moveto (0,0);           // курсор в точку (0,0)
    lineto (50,50);         // отрезок в точку (50,50)
    setlinestyle (DOTTED_LINE, 1, 1); //линия из точек, толщина 1
    setcolor (CYAN);        //бирюзовый цвет
    lineto (100,50);
    setlinestyle(SOLID_LINE, 1, 3); //сплошная линия, толщина 3
    setcolor(RED);
    lineto(100,100);
    getch();                // ждать нажатия на любую клавишу
    closegraph ();          // закрыть окно с графикой
    return 0;
}
```

Круги и дуги

```
#include <graphics.h>
int main()
{
    initwindow (400,300);
    setcolor(RED);
    arc(100, 100, 45, 135, 50);
    //дуга: x центра, y центра, начальный угол, конечный угол, радиус
    setcolor(CYAN);
    circle(150, 150, 30); //окружность: x центра, y центра, радиус
    setcolor(MAGENTA);
    ellipse(200, 200, 0, 360, 150, 40); //эллиптическая дуга:
    //x центра, y центра, начальный угол, конечный угол, радиус x, радиус y
    getch();
    closegraph ();
    return 0;
}
```

Заливка

```
#include <graphics.h>
int main()
{
    initwindow (400,300);
    setcolor(GREEN);
    sector(50,250,45,135,20,50);           //сектор эллипса:
    //x центра, y центра, начальный угол, конечный угол, радиус x, радиус y
    setcolor(MAGENTA);
    setfillstyle(4,BLUE);                  //тип заливки: стиль (0 – 12), цвет
    fillellipse(200,200,150,40);          //закрашенный эллипс:
    //x левого верхнего угла, y левого верхнего угла, радиус x, радиус y
    setcolor(CYAN);
    circle(150,150,30);
    setfillstyle(1,RED);
    floodfill(150,160,CYAN);              //заливка: x центра, y центра, цвет границы
    getch();
    closegraph ();
    return 0;
}
```

Многоугольники

```
#include <graphics.h>
int main()
{
    initwindow (400,300);
    setcolor(GREEN);
    int points[10] = {50, 50, 100, 50, 100, 100, 75, 125, 50, 100};
    drawpoly(5, points);           //нарисовать ломаную по точкам
    int pointsf[10] = {150, 150, 200, 150, 200, 200, 175, 225, 150, 200};
    fillpoly(5, pointsf);          //нарисовать многоугольник по точкам
    setcolor(GREEN);
    rectangle(50, 50, 250, 150);   //нарисовать прямоугольник:
    //левый край, верхний, правый, нижний
    setfillstyle(4,RED);
    bar(100, 100, 350, 250);       //нарисовать сплошной прямоугольник:
    //левый край, верхний, правый, нижний
    getch();
    closegraph ();
    return 0;
}
```


Клики мыши

```
#include <graphics.h>
int main() {
    initwindow (400,300);
    setcolor(GREEN);
    rectangle(50,50,250,150);           //зеленый прямоугольник
    int x = 0, y = 0;
    while(x < 200)                       //пока кликаем в левую половину окна
        while (!ismouseclick(WM_LBUTTONDOWN)) //пока не сделан клик мыши
            delay(500);                  //ждем 500 мс
        getmouseclick(WM_LBUTTONDOWN, x, y);
        printf("%d %d\n",x,y);          //получаем и печатаем координаты клика
    }
    setfillstyle(4,RED);
    bar(100,100,350,250);               //красный прямоугольник
    getch();
    closegraph ();
    return 0;
}
```

Случайность

```
#include <graphics.h>
int main() {
    initwindow (400,300);
    setcolor(GREEN);
    srand(time(NULL));           // «зерно» рандома – текущее время
    int x = 0, y = 0;
    while(x < 200)               //пока кликаем в левую половину окна
    {
        while (!ismouseclick(WM_LBUTTONDOWN))
            delay(500);
        getmouseclick(WM_LBUTTONDOWN, x, y);
        x_circle = rand()%400;   //произвольное целое число 0 – 399
        y_circle = rand()%300;   //произвольное целое число 0 – 299
        circle(x_circle, y_circle, 5);
    }
    getch();
    closegraph ();
    return 0;
}
```

Анимация

```
#include <graphics.h>
int main() {
    initwindow (400,300);
    int x=0, y=0, x_circle = 50, y_circle = 150;
    setcolor(BLACK);
    setfillstyle(1,BLUE);
    fillellipse(x_circle,y_circle,10,10);
    while (!ismouseclick(WM_LBUTTONDOWN)) {
        delay(500);
        setfillstyle(1,BLACK);
        fillellipse(x_circle,y_circle,10,10);
        setfillstyle(1,BLUE);
        x_circle += 5;
        fillellipse(x_circle,y_circle,10,10);
    }
    setfillstyle(4,RED);
    bar(50,50,350,250);
    getch();
    closegraph ();
    return 0;
}
```

//начальные координаты
//черная граница
//синяя заливка
//первый кадр
//рисует до клика мышью
//время между кадрами
//закрашиваем старый кадр черным

//рисует новый кадр синим
//и со смещением вправо

//в конце рисуем красный прямоугольник

Спецэффекты

```
#include <graphics.h>
#include <time.h>
int main()
{
    initwindow(400,300);
    int color = 0;
    srand(time(NULL));
    while(1)
    {
        delay(500);
        for (int x=0; x < 400; x++)
            for (int y=0; y < 300; y++)
                putpixel(x, y, color);
        color++;
        if (color > 15) color = 0;
        printf("%d ", color);
    }
    getch();
    closegraph();
    return 0;
}
```

```
#include <graphics.h>
#include <time.h>
#include <math.h>
int main() {
    int R = 300;
    initwindow(2*R,2*R);
    int color = 0;
    srand(time(NULL));
    while(1) {
        for (int r=0; r<R; r++)
            for (int p=0; p < 360; p++)
                if (getpixel (R + r*sin(p), R + r*cos(p)) == color-1)
                    putpixel (R + r*sin(p), R + r*cos(p), rand()%2*color);
        color++;
        if (color > 15) color = 0;
        printf("%d ", color);
    }
    getch();
    closegraph();
    return 0;
}
```

Текст

```
#include <graphics.h>

int main()
{
    initwindow (400,300);
    moveto (0,0);
    lineto (50,50);
    outtext ("from 50:50");
    settextstyle (SCRIPT_FONT, 1, 2);
    // шрифт, направление, размер
    outtextxy (100, 100, "from 100:100");
    getch();
    closegraph ();
    return 0;
}
```

Шрифты:

0. DEFAULT_FONT
1. TRIPLEX_FONT
2. SMALL_FONT
3. SANS_SERIF_FONT
4. GOTHIC_FONT
5. SCRIPT_FONT
6. SIMPLEX_FONT
7. TRIPLEX_SCR_FONT
8. COMPLEX_FONT
9. EUROPEAN_FONT
10. BOLD_FONT

Направление:

0. Нормальный вывод
1. Строка повёрнута на 90 градусов по часовой стрелке
2. Буквы в строке повёрнуты на 90 градусов против часовой стрелки

Константы

Клавиатура:

- вверх – 72
- вниз – 80
- влево – 75
- вправо – 77
- esc – 27
- enter – 13
- space – 32

Цвета:

0. BLACK
1. BLUE
2. GREEN
3. CYAN
4. RED
5. MAGENTA
6. BROWN
7. LIGHTGRAY
8. DARKGRAY
9. LIGHTBLUE
10. LIGHTGREEN
11. LIGHTCYAN
12. LIGHTRED
13. LIGHTMAGENTA
14. YELLOW
15. WHITE

Линии (стиль):

0. SOLID_LINE
1. DOTTED_LINE
2. CENTER_LINE
3. DASHED_LINE
4. USERBIT_LINE

Линии (толщина):

0. NORM_WIDTH
1. THICK_WIDTH

Заливка:

0. EMPTY_FILL
1. SOLID_FILL
2. LINE_FILL
3. LTSLASH_FILL
4. SLASH_FILL
5. BKSLASH_FILL
6. LTBKSLASH_FILL
7. HATCH_FILL
8. XHATCH_FILL
9. INTERLEAVE_FILL
10. WIDE_DOT_FILL
11. CLOSE_DOT_FILL
12. USER_FILL

Задачи

0. Нарисовать треугольник.
1. Нарисовать график параболы.
2. «Убегающий кружок». При клике на кружок он перескакивает на произвольный участок экрана и меняет цвет.
3. «Два нейтрона». Два шарика одинакового радиуса и разного цвета расположены в разных углах экрана. С клавиатуры задаются четыре числа: скорости и углы вылета шариков. Шарик упряго отскакивают от стен и друг от друга. Рисовать до клика мышью.
4. «Лабиринт». Нарисовать лабиринт, состоящий из черных стен и белых коридоров. Красный – игрок, желтый – выход. С помощью стрелок клавиатуры игрок перемещается по коридорам, попадание на желтый – победа, на черный – поражение.

