

Основы программирования

Лабораторная работа №10

WinAPI Графика – Относительные координаты.

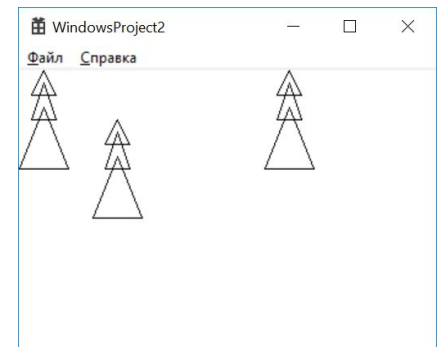
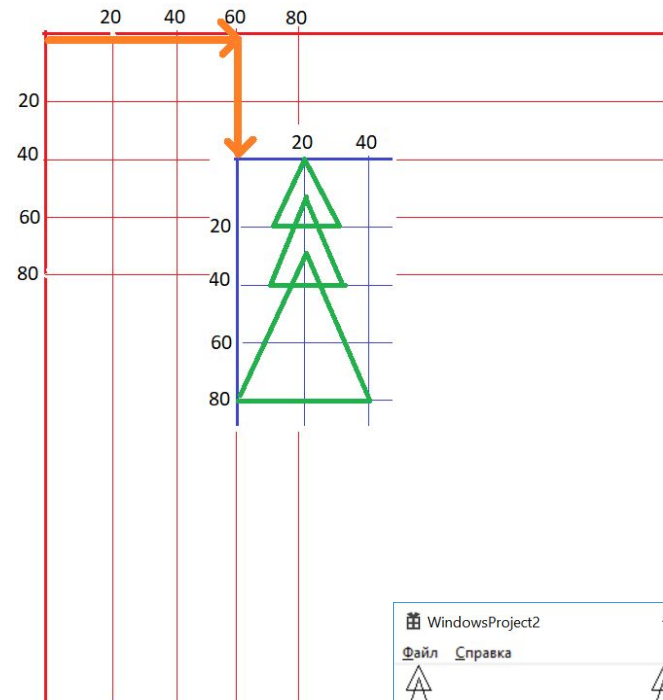
Управление размером фигур.

Функции с параметрами.

Власенко Олег Федосович

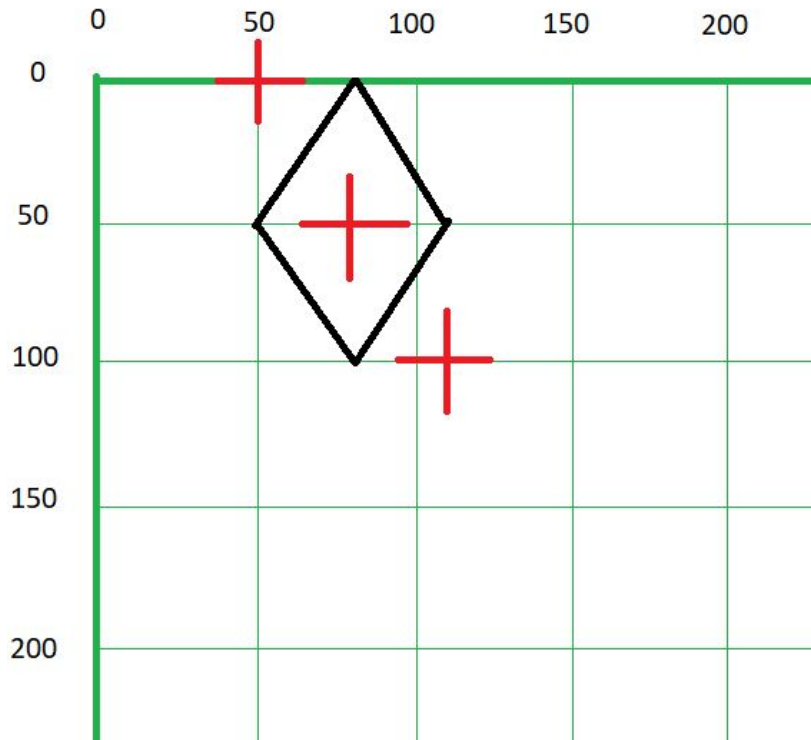
Относительные координаты

```
void StClausAuto(HDC hdc, int x, int y) {  
    // верхний треугольник  
    MoveToEx(hdc, 20 + x, 0 + y, NULL);  
    LineTo(hdc, 30 + x, 20 + y);  
    LineTo(hdc, 10 + x, 20 + y);  
    LineTo(hdc, 20 + x, 0 + y);  
  
    // средний треугольник  
    MoveToEx(hdc, 20 + x, 10 + y, NULL);  
    LineTo(hdc, 30 + x, 40 + y);  
    LineTo(hdc, 10 + x, 40 + y);  
    LineTo(hdc, 20 + x, 10 + y);  
  
    // нижний треугольник  
    MoveToEx(hdc, 20 + x, 30 + y, NULL);  
    LineTo(hdc, 40 + x, 80 + y);  
    LineTo(hdc, 0 + x, 80 + y);  
    LineTo(hdc, 20 + x, 30 + y);  
}  
...
```



```
case WM_PAINT:  
{  
    PAINTSTRUCT ps;  
    HDC hdc = BeginPaint(hWnd, &ps);  
    // TODO: Добавьте сюда любой код прорисовки, использующий HDC...  
  
    StClausAuto(hdc, 60, 40);  
    StClausAuto(hdc, 0, 0);  
    StClausAuto(hdc, 200, 0);  
  
    EndPaint(hWnd, &ps);  
}  
break;
```

Относительные координаты



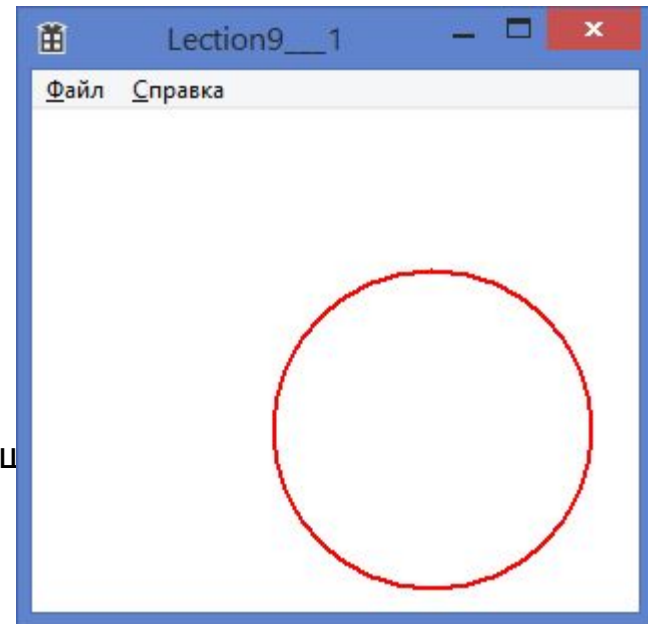
```
int x = 50;  
int y = 0;  
MoveToEx(hdc, x + 30, y, NULL);  
LineTo(hdc, x, y + 50);  
LineTo(hdc, x + 30, y + 100);  
LineTo(hdc, x + 60, y + 50);  
LineTo(hdc, x + 30, y);
```

Рисование окружности

```
void Circle(HDC hdc, int cx, int cy, int size) {  
    HPEN hPen;  
    hPen = CreatePen(PS_SOLID, 2, RGB(255, 0, 0));  
    SelectObject(hdc, hPen);  
    Ellipse(hdc, cx - size, cy - size, cx + size, cy + size);  
  
    DeleteObject(hPen);  
}
```

...

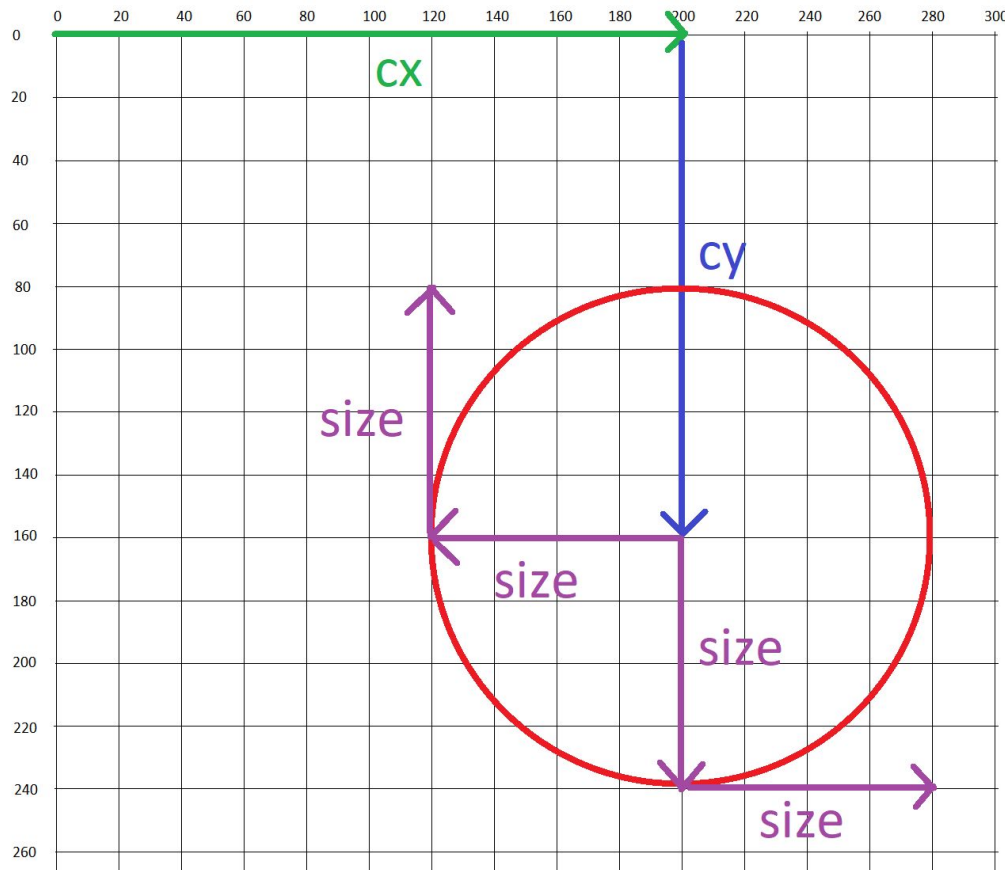
```
case WM_PAINT:  
    {  
        PAINTSTRUCT ps;  
        HDC hdc = BeginPaint(hWnd, &ps);  
        // TODO: Добавьте сюда любой код прорисовки, используя  
        Circle(hdc, 200, 160, 80);  
        EndPaint(hWnd, &ps);  
    }  
    break;
```



Рисование окружности

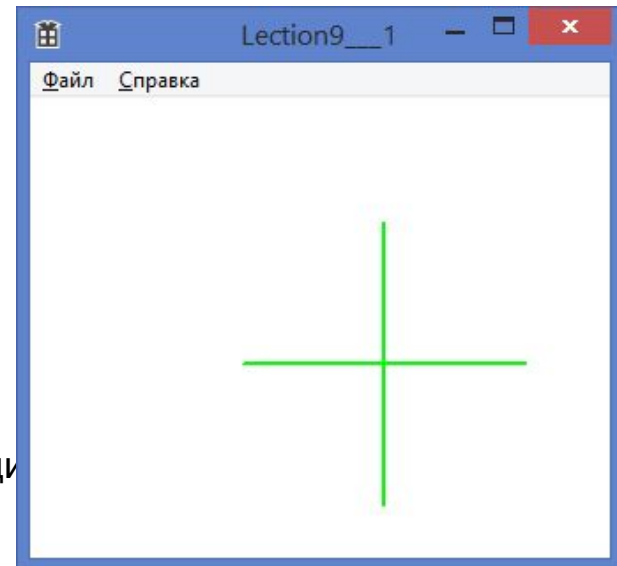
```
void Circle(HDC hdc, int cx, int cy, int size) {  
    HPEN hPen;  
    hPen = CreatePen(PS_SOLID, 2, RGB(255, 0, 0));  
    SelectObject(hdc, hPen);  
    Ellipse(hdc, cx - size, cy - size, cx + size, cy + size);  
  
    DeleteObject(hPen);  
}  
...
```

```
case WM_PAINT:  
    {  
        PAINTSTRUCT ps;  
        HDC hdc = BeginPaint(hWnd, &ps);  
        // TODO: Добавьте сюда любой код  
        Circle(hdc, 200, 160, 80);  
        EndPaint(hWnd, &ps);  
    }  
    break;
```



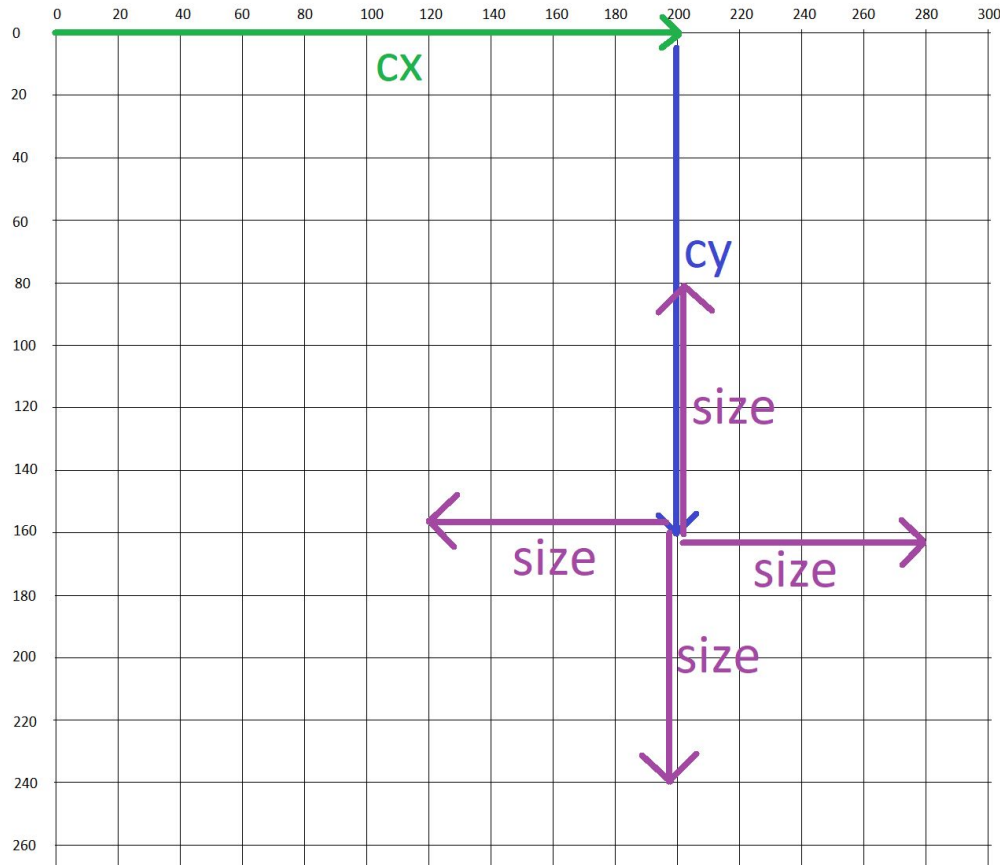
Рисование креста

```
void Cross(HDC hdc, int cx, int cy, int size) {
    HPEN hPen;
    hPen = CreatePen(PS_SOLID, 2, RGB(0, 255, 0));
    SelectObject(hdc, hPen);
    MoveToEx(hdc, cx - size, cy, NULL);
    LineTo(hdc, cx + size, cy);
    MoveToEx(hdc, cx, cy - size, NULL);
    LineTo(hdc, cx, cy + size);
    DeleteObject(hPen);
}
...
case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: Добавьте сюда любой код прорисовки, использующий
    Cross(hdc, 200, 160, 80);
    EndPaint(hWnd, &ps);
}
break;
```



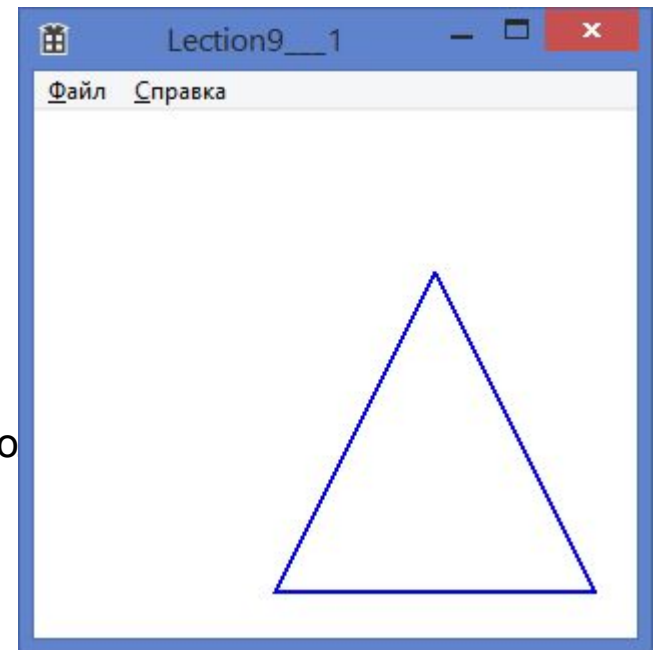
Рисование креста

```
void Cross(HDC hdc, int cx, int cy, int size) {  
    HPEN hPen;  
    hPen = CreatePen(PS_SOLID, 2, RGB(0, 255, 0));  
    SelectObject(hdc, hPen);  
    MoveToEx(hdc, cx - size, cy, NULL);  
    LineTo(hdc, cx + size, cy);  
    MoveToEx(hdc, cx, cy - size, NULL);  
    LineTo(hdc, cx, cy + size);  
    DeleteObject(hPen);  
} ...  
case WM_PAINT:  
    {  
        PAINTSTRUCT ps;  
        HDC hdc = BeginPaint(hWnd, &ps);  
        // TODO: Добавьте сюда любой код  
        Cross(hdc, 200, 160, 80);  
        EndPaint(hWnd, &ps);  
    }  
break;
```



Рисование треугольника

```
void Triangle(HDC hdc, int cx, int cy, int size) {
    HPEN hPen;
    hPen = CreatePen(PS_SOLID, 2, RGB(0, 0, 255));
    SelectObject(hdc, hPen);
    MoveToEx(hdc, cx, cy - size, NULL);
    LineTo(hdc, cx + size, cy + size);
    LineTo(hdc, cx - size, cy + size);
    LineTo(hdc, cx, cy - size);
    DeleteObject(hPen);
}
...
case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: Добавьте сюда любой код прорисовки, испо
    Triangle(hdc, 200, 160, 80);
    EndPaint(hWnd, &ps);
}
break;
```



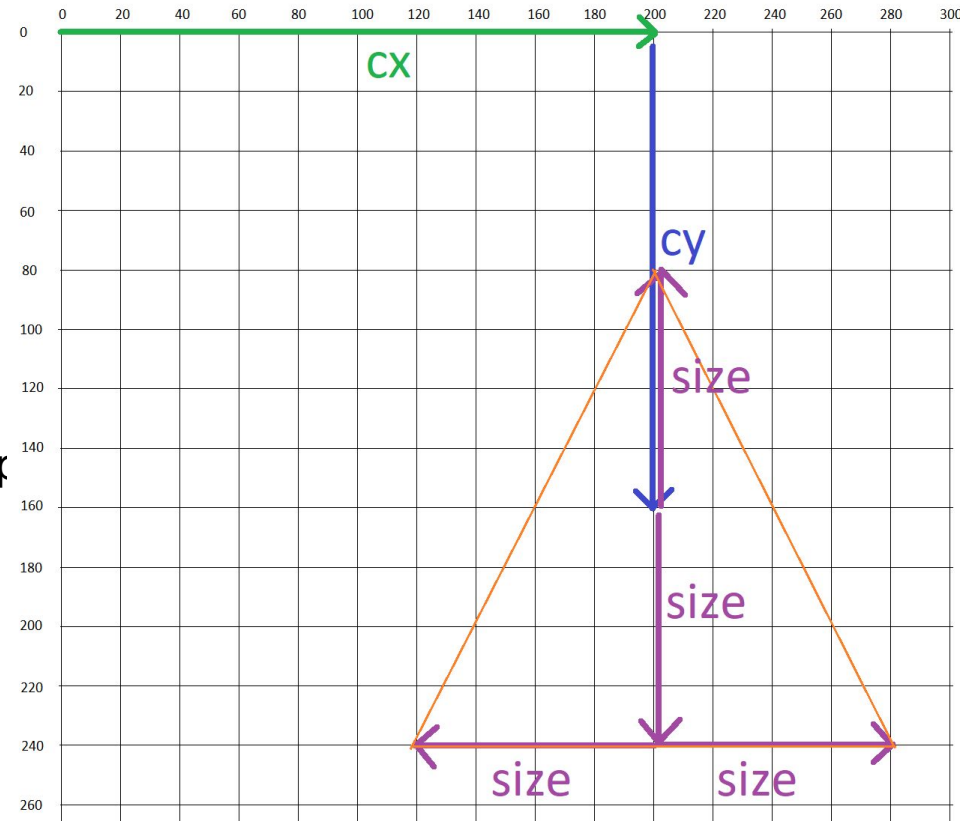
Рисование треугольника

```
void Triangle(HDC hdc, int cx, int cy, int size) {
    HPEN hPen;
    hPen = CreatePen(PS_SOLID, 2, RGB(0, 0, 255));
    SelectObject(hdc, hPen);

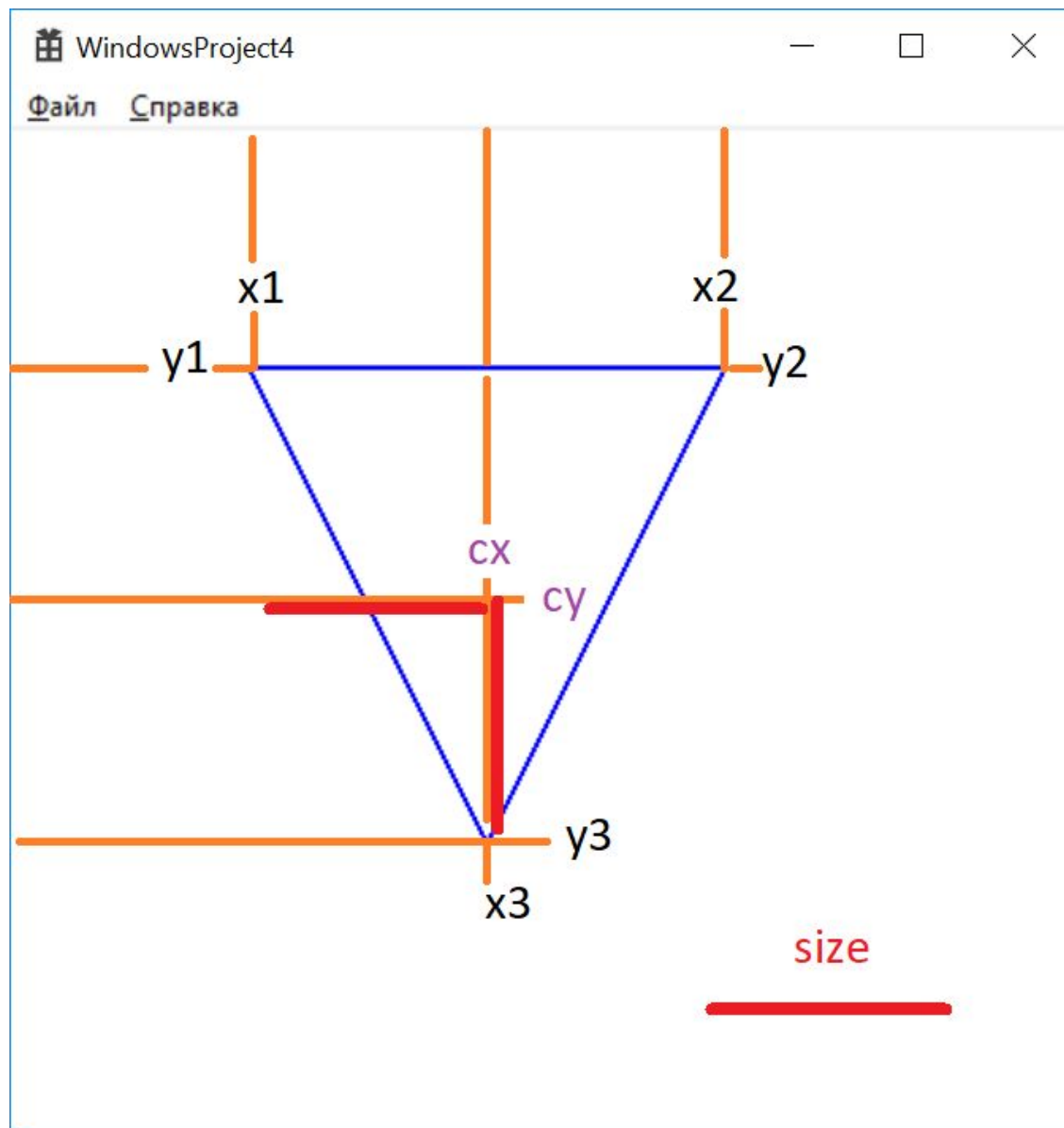
    MoveToEx(hdc, cx, cy - size, NULL);
    LineTo(hdc, cx + size, cy + size);
    LineTo(hdc, cx - size, cy + size);
    LineTo(hdc, cx, cy - size);

    DeleteObject(hPen);
}

...
case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: Добавьте сюда любой код пр
    Triangle(hdc, 200, 160, 80);
    EndPaint(hWnd, &ps);
}
break;
```



Рисование треугольника 2



Рисование треугольника 2

```
void Triangle2(HDC hdc, int cx, int cy, int size) {  
    int x1 = cx - size;  
    int y1 = cy - size;  
  
    int x2 = cx + size;  
    int y2 = cy - size;  
  
    int x3 = cx;  
    int y3 = cy + size;  
  
    HPEN hPen;  
    hPen = CreatePen(PS_SOLID, 2, RGB(0, 0, 255));  
    SelectObject(hdc, hPen);  
    MoveToEx(hdc, x1, y1, NULL);  
    LineTo(hdc, x2, y2);  
    LineTo(hdc, x3, y3);  
    LineTo(hdc, x1, y1);  
    DeleteObject(hPen);  
}  
...  
Triangle2 (hdc, 200, 160, 80);
```

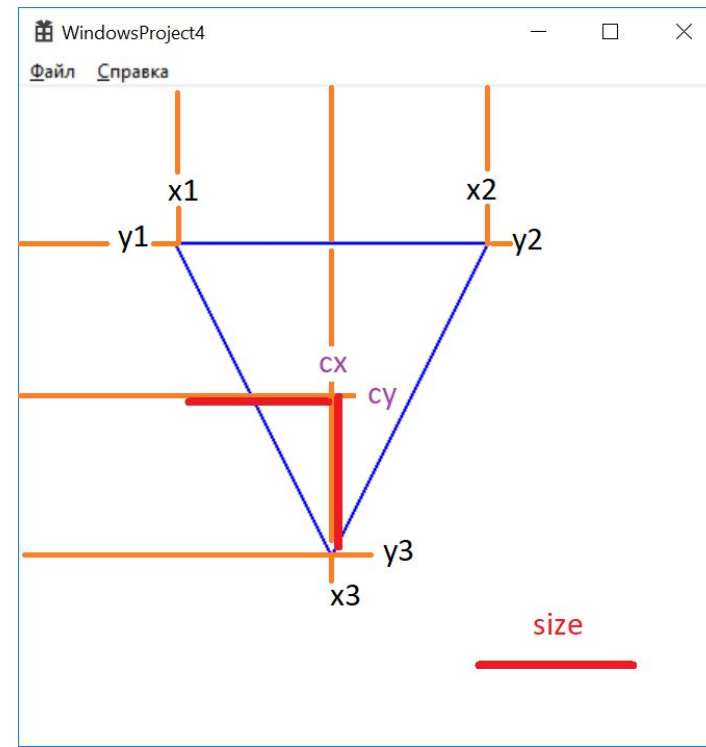


Image1

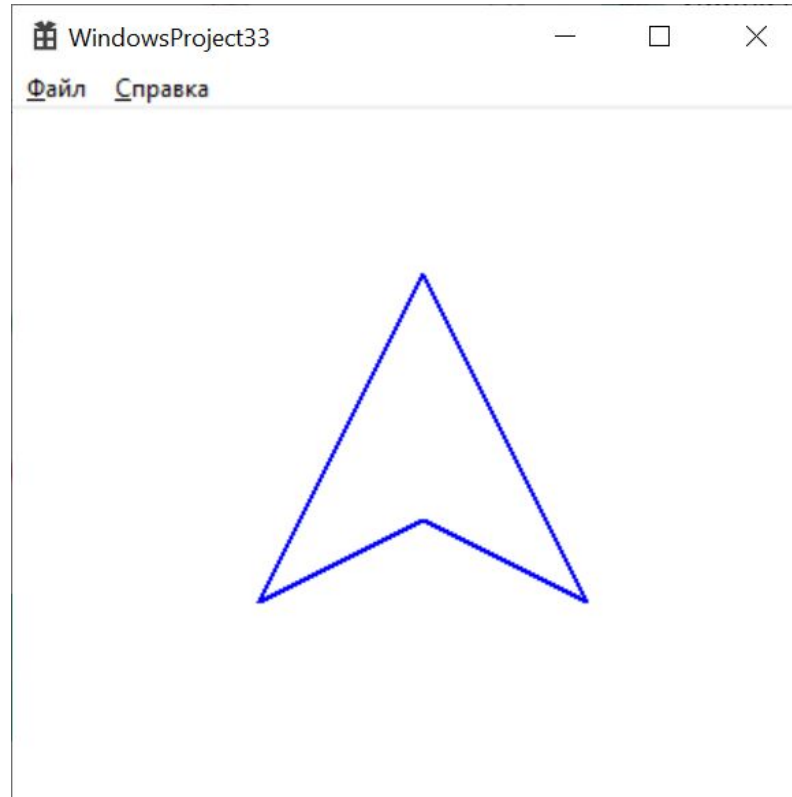


Image1

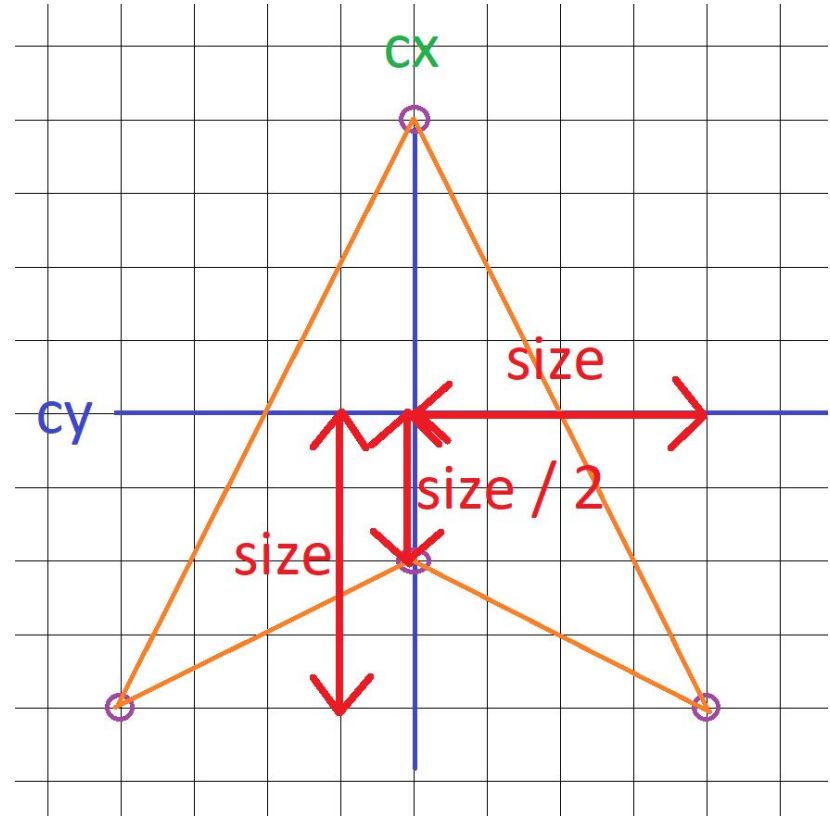
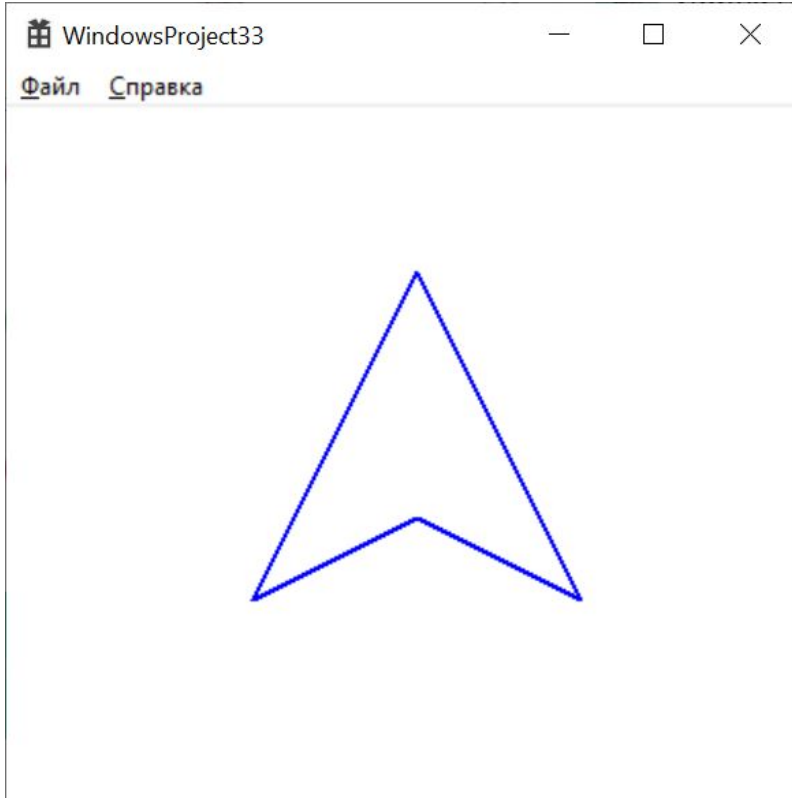


Image1

```
void Image1(HDC hdc, int cx, int cy, int size) {  
    int x1 = cx;  
    int y1 = cy - size;  
  
    int x2 = cx + size;  
    int y2 = cy + size;  
  
    int x3 = cx;  
    int y3 = cy + size / 2;  
  
    int x4 = cx - size;  
    int y4 = cy + size;  
  
    HPEN hPen;  
    hPen = CreatePen(PS_SOLID, 2, RGB(0, 0, 255));  
    SelectObject(hdc, hPen);  
  
    MoveToEx(hdc, x1, y1, NULL);  
    LineTo(hdc, x2, y2);  
    LineTo(hdc, x3, y3);  
    LineTo(hdc, x4, y4);  
    LineTo(hdc, x1, y1);  
  
    DeleteObject(hPen);  
}
```

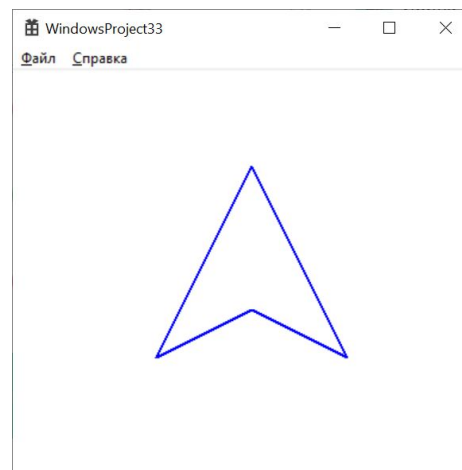
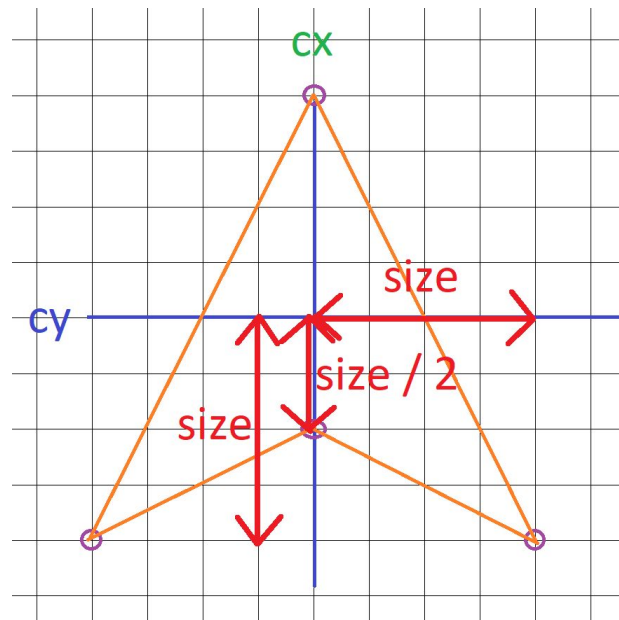


Image2

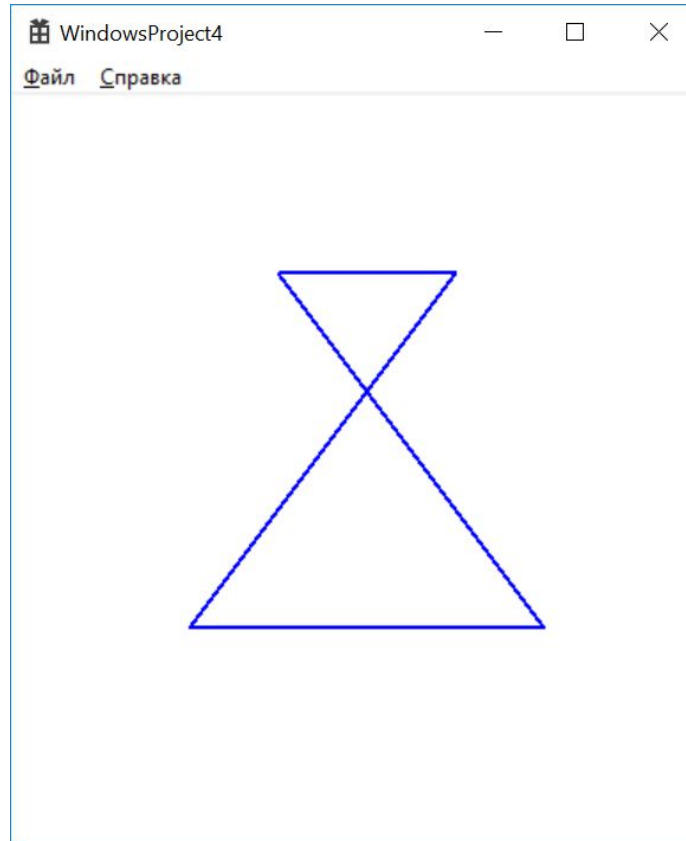


Image2

```
void Image2(HDC hdc, int cx, int cy, int size) {  
    int x1 = cx - size / 2;  
    int y1 = cy - size;  
    int x2 = cx + size / 2;  
    int y2 = cy - size;  
    int x3 = cx - size;  
    int y3 = cy + size;  
    int x4 = cx + size;  
    int y4 = cy + size;  
  
    HPEN hPen;  
    hPen = CreatePen(PS_SOLID, 2, RGB(0, 0, 255));  
    SelectObject(hdc, hPen);  
  
    MoveToEx(hdc, x1, y1, NULL);  
    LineTo(hdc, x2, y2);  
    LineTo(hdc, x3, y3);  
    LineTo(hdc, x4, y4);  
    LineTo(hdc, x1, y1);  
  
    DeleteObject(hPen);  
}
```

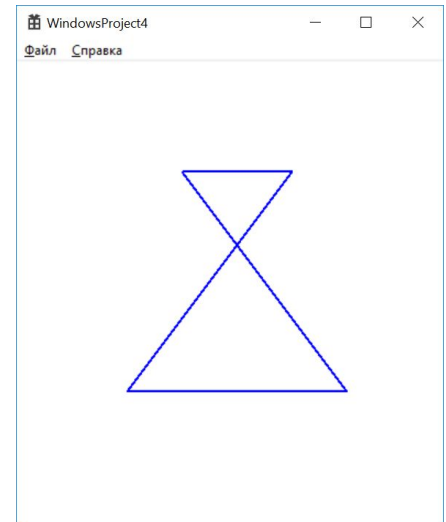


Image2

```
void Image2(HDC hdc, int cx, int cy, int size) {  
    int x1 = cx - size / 2;  
    int y1 = cy - size;  
    int x2 = cx + size / 2;  
    int y2 = cy - size;  
    int x3 = cx - size;  
    int y3 = cy + size;  
    int x4 = cx + size;  
    int y4 = cy + size;  
  
    HPEN hPen;  
    hPen = CreatePen(PS_SOLID, 2, RGB(0, 0, 255));  
    SelectObject(hdc, hPen);  
  
    MoveToEx(hdc, x1, y1, NULL);  
    LineTo(hdc, x2, y2);  
    LineTo(hdc, x3, y3);  
    LineTo(hdc, x4, y4);  
    LineTo(hdc, x1, y1);  
  
    DeleteObject(hPen);  
}
```

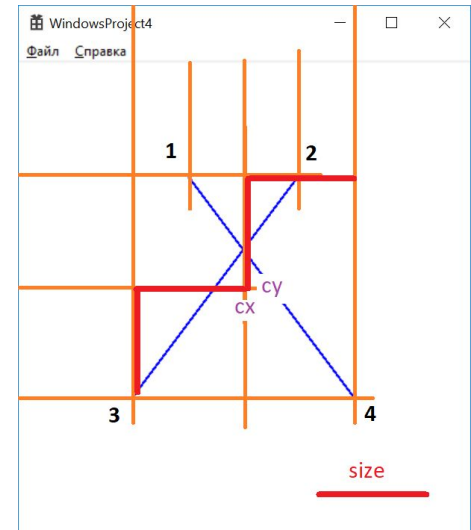
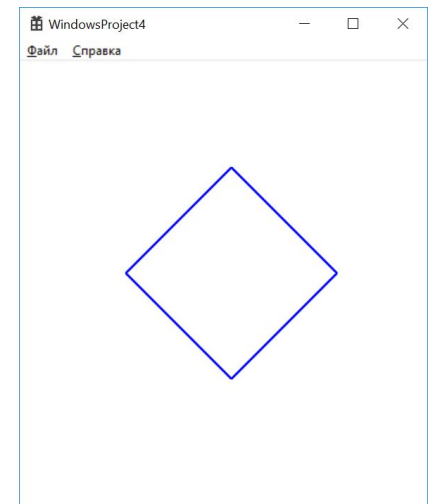


Image3

```
void Image3(HDC hdc, int cx, int cy, int size) {  
    int x1 = cx;  
    int y1 = cy - size;  
  
    int x2 = cx + size;  
    int y2 = cy;  
  
    int x3 = cx;  
    int y3 = cy + size;  
  
    int x4 = cx - size;  
    int y4 = cy;  
  
    HPEN hPen;  
    hPen = CreatePen(PS_SOLID, 2, RGB(0, 0, 255));  
    SelectObject(hdc, hPen);  
  
    MoveToEx(hdc, x1, y1, NULL);  
    LineTo(hdc, x2, y2);  
    LineTo(hdc, x3, y3);  
    LineTo(hdc, x4, y4);  
    LineTo(hdc, x1, y1);  
  
    DeleteObject(hPen);  
}
```



```
case WM_PAINT:  
{  
    PAINTSTRUCT ps;  
    HDC hdc = BeginPaint(hWnd, &ps);  
    // TODO: Добавьте сюда любой код прорисовки, использующий HDC...  
  
    Image3(hdc, 200, 200, 100);  
  
    EndPaint(hWnd, &ps);  
}  
break;
```

Image4

```
void Image4(HDC hdc, int cx, int cy, int size) {
```

```
    int x1 = cx;  
    int y1 = cy - size;  
  
    int x2 = cx + size / 4;  
    int y2 = cy - size / 4;  
  
    int x3 = cx + size;  
    int y3 = cy;  
  
    int x4 = cx + size / 4;  
    int y4 = cy + size / 4;  
  
    int x5 = cx;  
    int y5 = cy + size;
```

```
    HPEN hPen;  
    hPen = CreatePen(PS_SOLID, 2, RGB(r_g, g_g, b_g));  
    SelectObject(hdc, hPen);
```

```
    MoveToEx(hdc, x1, y1, NULL);  
    LineTo(hdc, x2, y2);  
    LineTo(hdc, x3, y3);  
    LineTo(hdc, x4, y4);  
    LineTo(hdc, x5, y5);  
    LineTo(hdc, x6, y6);  
    LineTo(hdc, x7, y7);  
    LineTo(hdc, x8, y8);  
    LineTo(hdc, x1, y1);
```

```
    DeleteObject(hPen);
```

```
}
```

```
case WM_PAINT:
```

```
{
```

```
    PAINTSTRUCT ps;  
    HDC hdc = BeginPaint(hWnd, &ps);  
    // TODO: Добавьте сюда любой код прорисовки, использующий HDC...
```

```
    Image4(hdc, 200, 200, 100);
```

```
    EndPaint(hWnd, &ps);
```

```
}
```

```
break;
```

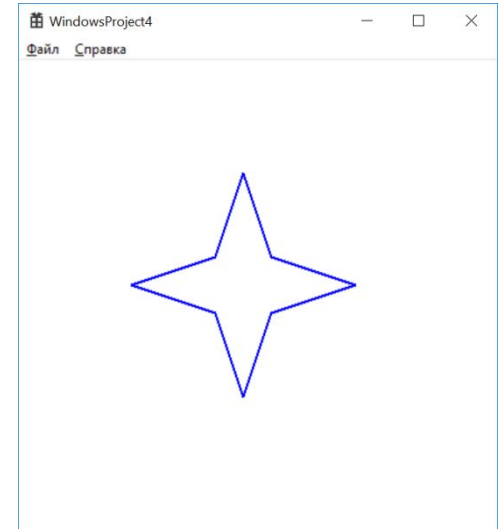
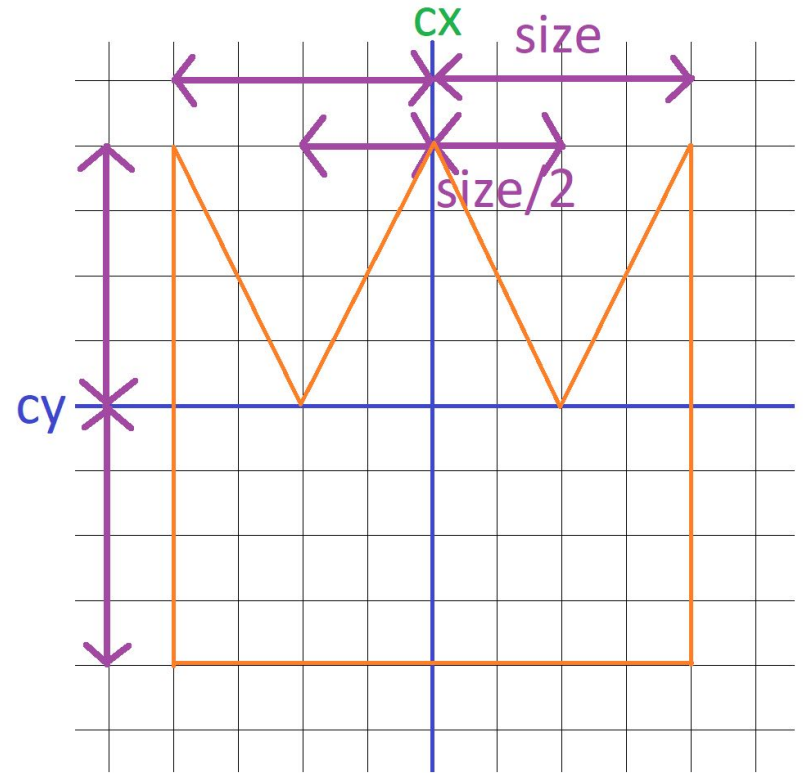
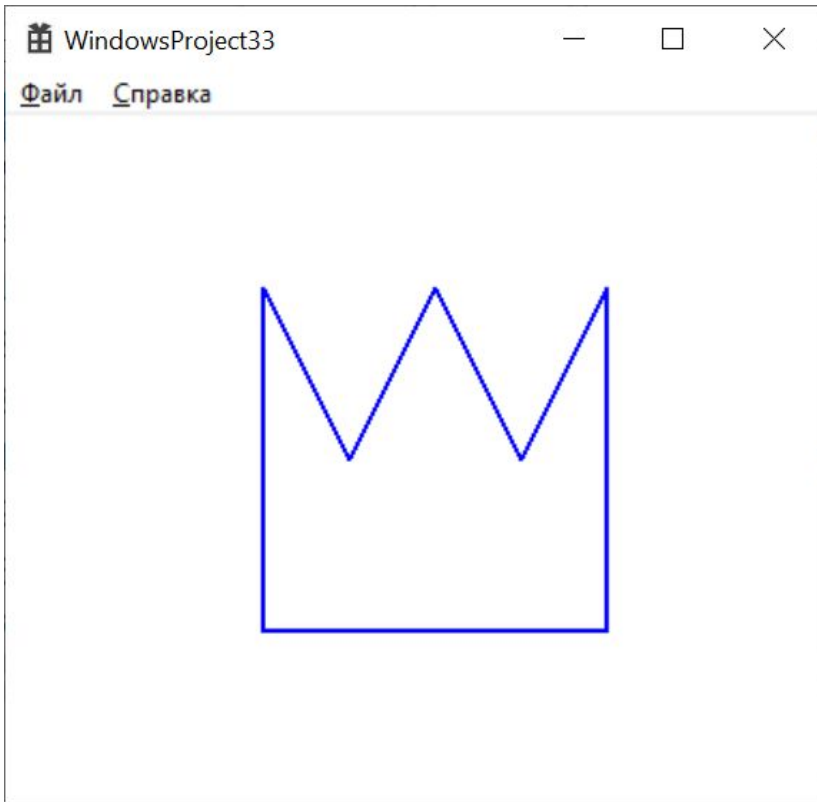
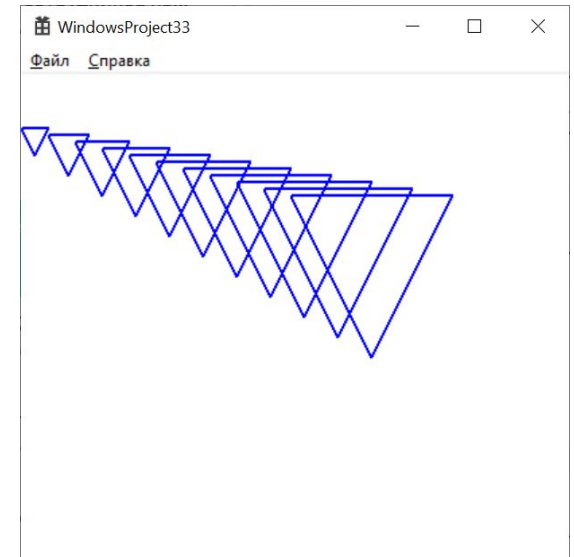


Image5



Функция вызывает функцию

```
244 void example1(HDC hdc)
245 {
246     int x = 10;
247     int y = 50;
248     int size = 10;
249
250     do {
251         Triangle2(hdc, x, y, size);
252         x += 25;
253         y += 10;
254         size += 5;
255
256     } while (x <= 280);
257 }
```

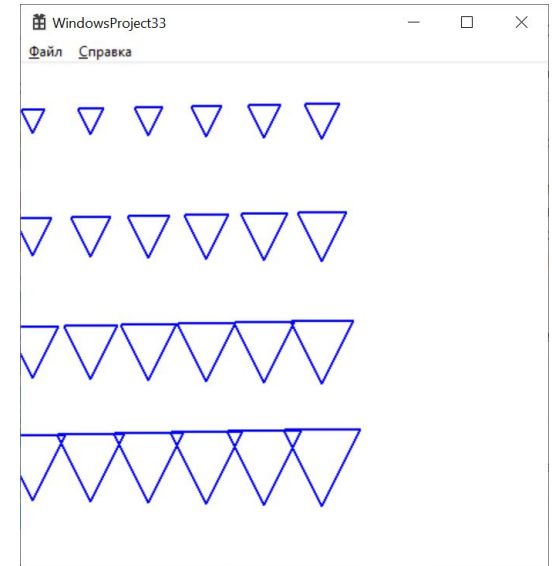


```
395 case WM_PAINT:
396     {
397         PAINTSTRUCT ps;
398         HDC hdc = BeginPaint(hWnd, &ps);
399         // TODO: Добавьте сюда любой код прорисовки, использующий HDC...
400         example1(hdc);
401         //example2(hdc);
402         //example3(hdc);
403         //example4(hdc);
404         //task1(hdc);
405         //task2(hdc);
406         //task2_2(hdc);
407         EndPaint(hWnd, &ps);
408     }
409     break;
```

Функция вызывает функцию

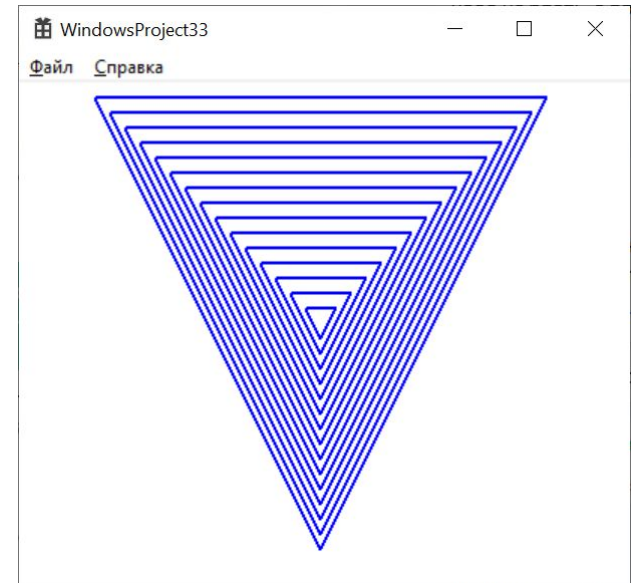
```
261 void example2(HDC hdc)
262 {
263     int y = 50;
264     int size = 10;
265     do {
266
267         int x = 10;
268         do {
269             Triangle2(hdc, x, y, size);
270             x += 50;
271             size += 1;
272         } while (x <= 280);
273         y += 100;
274     } while (y <= 400);
276 }
```

```
388 case WM_PAINT:
389     {
390         PAINTSTRUCT ps;
391         HDC hdc = BeginPaint(hWnd, &ps);
392         // TODO: Добавьте сюда любой код прорисовки, использующий HDC...
393         //example1(hdc);
394         example2(hdc);
395         //example3(hdc);
396         //example4(hdc);
397         //task1(hdc);
398         //task2(hdc);
399         //task2_2(hdc);
400         EndPaint(hWnd, &ps);
401     }
402     break;
```



ФУНКЦИЯ ВЫЗЫВАЕТ ФУНКЦИЮ

```
279 void example3(HDC hdc)
280 {
281     int x = 200;
282     int y = 160;
283     int size = 10;
284
285     do {
286         Triangle2(hdc, x, y, size);
287         size += 10;
288     } while (size <= 150);
289 }
```

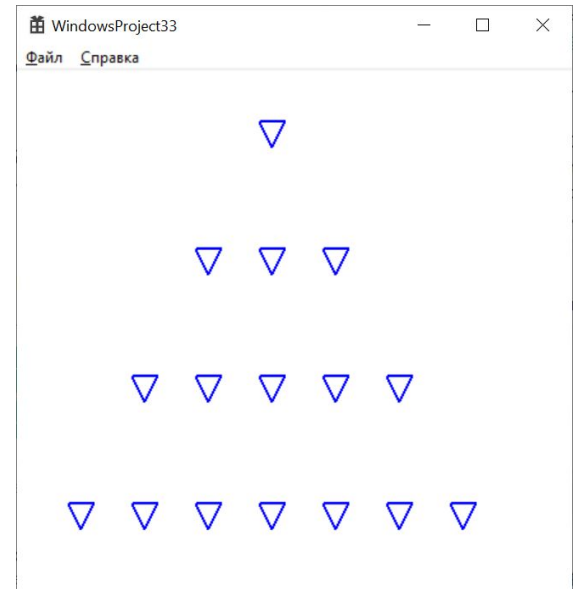


```
376 case WM_PAINT:
377 {
378     PAINTSTRUCT ps;
379     HDC hdc = BeginPaint(hWnd, &ps);
380     // TODO: Добавьте сюда любой код прорисовки, использующий HDC...
381     //example1(hdc);
382     //example2(hdc);
383     example3(hdc);
384     //example4(hdc);
385     //task1(hdc);
386     //task2(hdc);
387     //task2_2(hdc);
388     EndPaint(hWnd, &ps);
389 }
390 break;
```

Функция вызывает функцию

```
303 void example4(HDC hdc)
304 {
305     int xStart = 200;
306     int xFinish = 200;
307
308     int y = 50;
309     int size = 10;
310     do {
311         int x = xStart;
312
313         do {
314             Triangle2(hdc, x, y, size);
315             x += 50;
316         } while (x <= xFinish);
317
318         y += 100;
319         xStart -= 50;
320         xFinish += 50;
321     } while (y <= 400);
322 }
```

```
388 case WM_PAINT:
389     {
390         PAINTSTRUCT ps;
391         HDC hdc = BeginPaint(hWnd, &ps);
392         // TODO: Добавьте сюда любой код прорисовки, использующий HDC...
393         //example1(hdc);
394         //example2(hdc);
395         //example3(hdc);
396         example4(hdc);
397         //task1(hdc);
398         //task2(hdc);
399         //task2_2(hdc);
400         EndPaint(hWnd, &ps);
401     }
402     break;
```



Задача 1

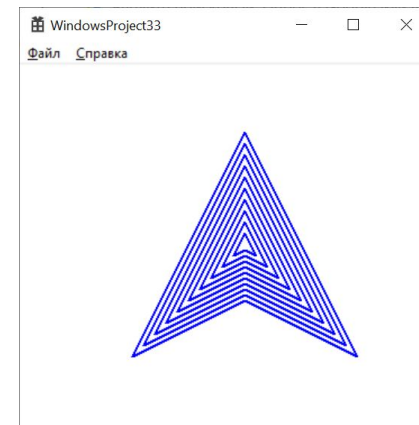
```
void task1(HDC hdc)
{
    int x = 200;
    int y = 160;
    int size = 10;

    do {
        Image1(hdc, 200, 160, size);
        size += 10;
    } while (size <= 100);
}

case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: Добавьте сюда любой код прорисовки, использующий HDC...
    //example1(hdc);

    task1(hdc);
    //task2(hdc);
    //task2_2(hdc);
    //task3(hdc);
    //task3_2(hdc);
    //task3_3(hdc);

    //hometask1(hdc);
    //hometask2(hdc);
    //hometask3(hdc);
    //hometask4__(hdc);
    //hometask5__(hdc);
    EndPaint(hWnd, &ps);
}
break;
```



Задача 2

```
void task2(HDC hdc)
{
    int x = 200;
    int y = 160;
    int size = 10;

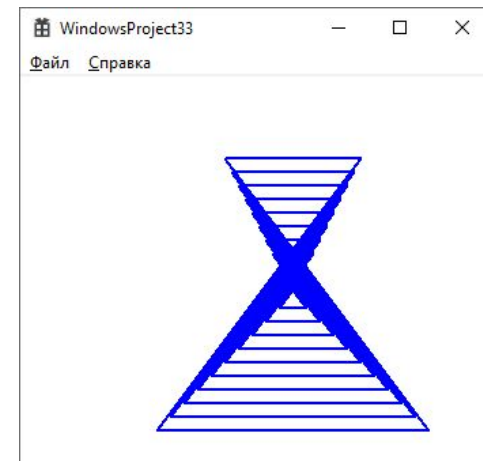
    do {
        Image2(hdc, 200, 160, size);
        size += 10;
    } while (size <= 100);
}
```

case WM_PAINT:

```
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: Добавьте сюда любой код прорисовки, использующий HDC...
    //example1(hdc);

    //task1(hdc);
    task2(hdc);
    //task2_2(hdc);
    //task3(hdc);
    //task3_2(hdc);
    //task3_3(hdc);

    //hometask1(hdc);
    //hometask2(hdc);
    //hometask3(hdc);
    //hometask4__(hdc);
    //hometask5__(hdc);
    EndPaint(hWnd, &ps);
}
break;
```



Задача 2.2

```
void task2_2(HDC hdc)
{
    int x = 10;
    int y = 10;
    int size = 10;

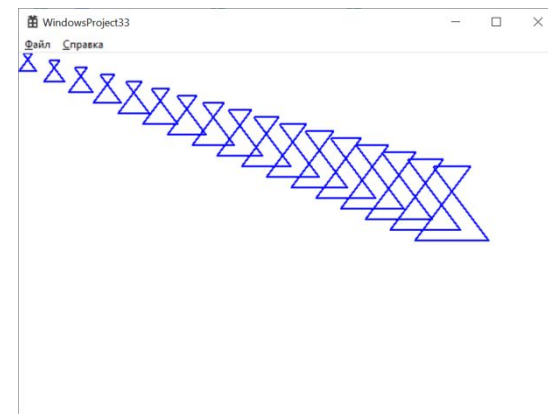
    do {
        Image2(hdc, x, y, size);

        x += 30;
        y += 10;
        size += 2;
    } while (x <= 500);
}
```

```
case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: Добавьте сюда любой код прорисовки, использующий HDC...
    //example1(hdc);

    //task1(hdc);
    //task2(hdc);
    task2_2(hdc);
    //task3(hdc);
    //task3_2(hdc);
    //task3_3(hdc);

    //hometask1(hdc);
    //hometask2(hdc);
    //hometask3(hdc);
    //hometask4__(hdc);
    //hometask5__(hdc);
    EndPaint(hWnd, &ps);
}
break;
```



Задача 3

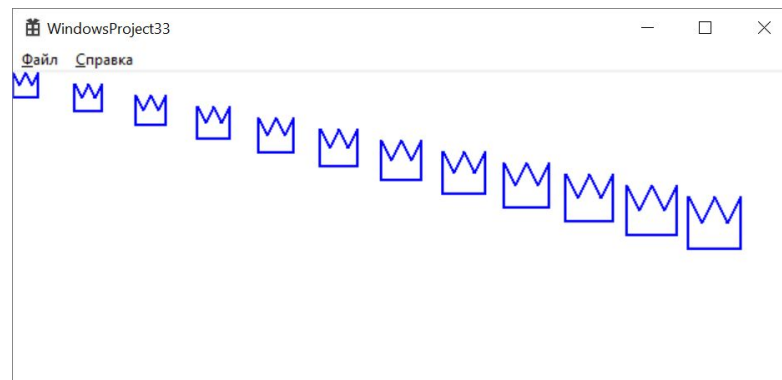
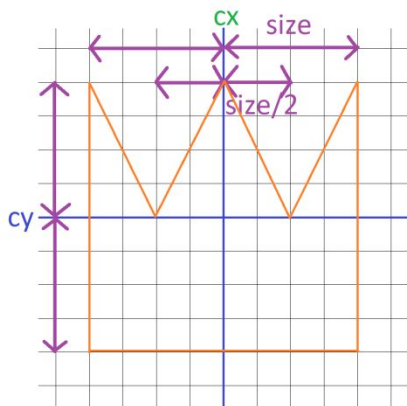
```
void Image5(HDC hdc, int cx, int cy, int size) {
    int x1 = cx;
    int y1 = cy - size;

    int x2 = cx + size / 2;
    int y2 = cy;

    int x3 = cx + size;
    int y3 = cy - size;

    int x4 = cx + size;
    int y4 = cy + size;

    int x5 = cx - size;
    int y5 = cy + size;
```



```
void task3(HDC hdc)
{
    int x = 10;
    int y = 10;
    int size = 10;

    do {
        Image5(hdc, x, y, size);

        x += 50;
        y += 10;
        size += 1;
    } while (x <= 600);
}
```

```
case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: Добавьте сюда любой код прорисовки, использующий HDC...
    //example1(hdc);

    //task1(hdc);
    //task2(hdc);
    //task2_2(hdc);
    task3(hdc);
    //task3_2(hdc);
    //task3_3(hdc);

    //hometask1(hdc);
    //hometask2(hdc);
    //hometask3(hdc);
    //hometask4__(hdc);
    //hometask5__(hdc);
    EndPaint(hWnd, &ps);
}
break;
```

Задача 3.2

Используя корону (рисунок создаваемый функцией `Image5()`) создайте любой красивый узор из множества корон.

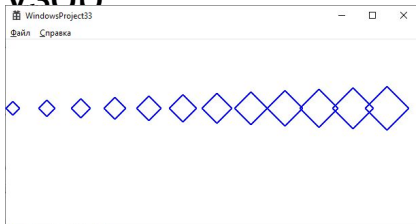
Задача 3.3

Используя корону (рисунок создаваемый функцией `Image5()`) создайте еще один любой красивый узор из множества корон.

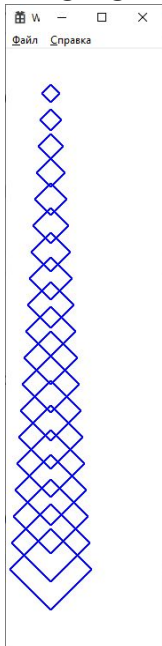
Домашнее задание

1. Используя ромб (рисунок создаваемый функцией `Image3()`) создайте

узор

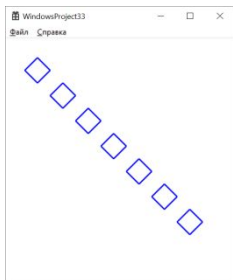


2. Используя ромб (рисунок создаваемый функцией `Image3()`) создайте

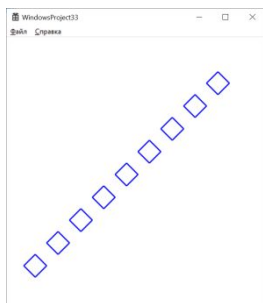


Домашнее задание

3. Используя ромб (рисунок создаваемый функцией `Image3()`) создайте узор



4. Используя ромб (рисунок создаваемый функцией `Image3()`) создайте узор



5*. Используя `Image4()` (четырёхконечную звезду) или свой собственный рисунок, создайте красивый узор