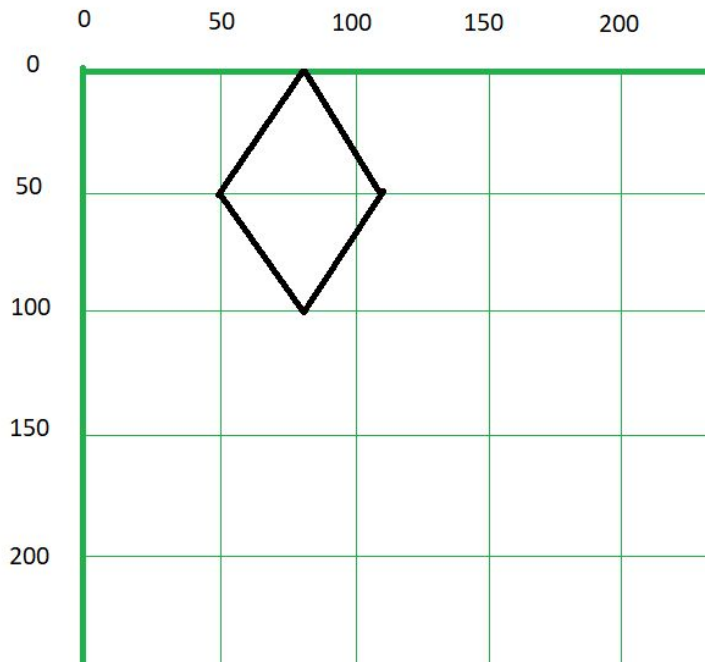
Основы программирования Лабораторная работа №9

WinAPI Графика – Относительные координаты.
Функции с параметрами.

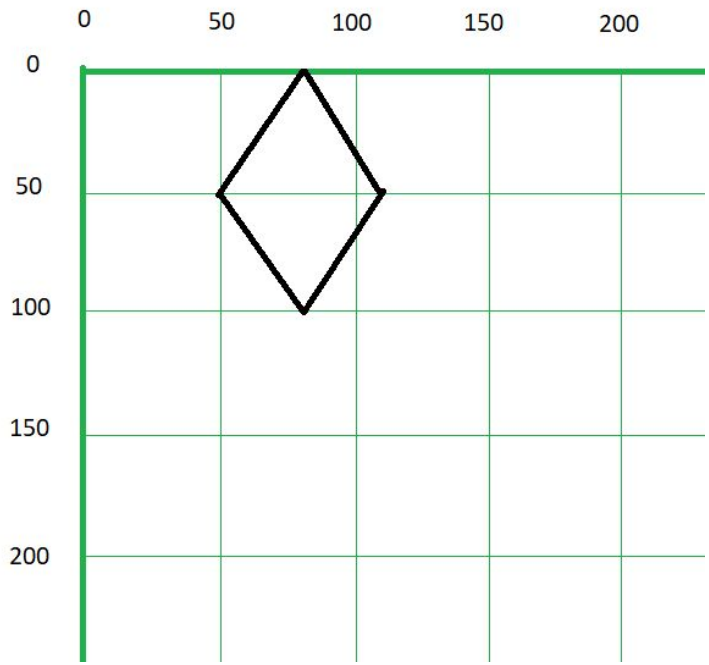
Власенко Олег Федосович

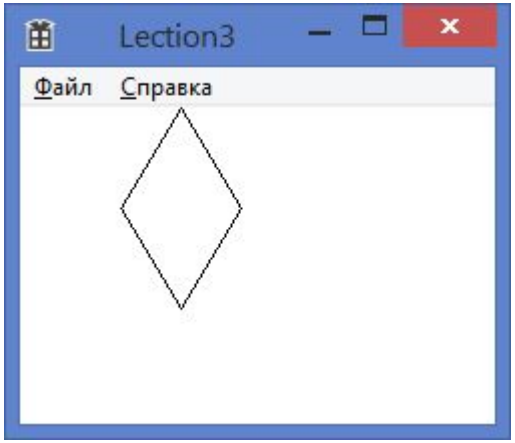
Нарисуем ромб




```
// Рисуем ромб
MoveToEx(hdc, 80, 0, NULL);
LineTo(hdc, 50, 50);
LineTo(hdc, 80, 100);
LineTo(hdc, 110, 50);
LineTo(hdc, 80, 0);
```

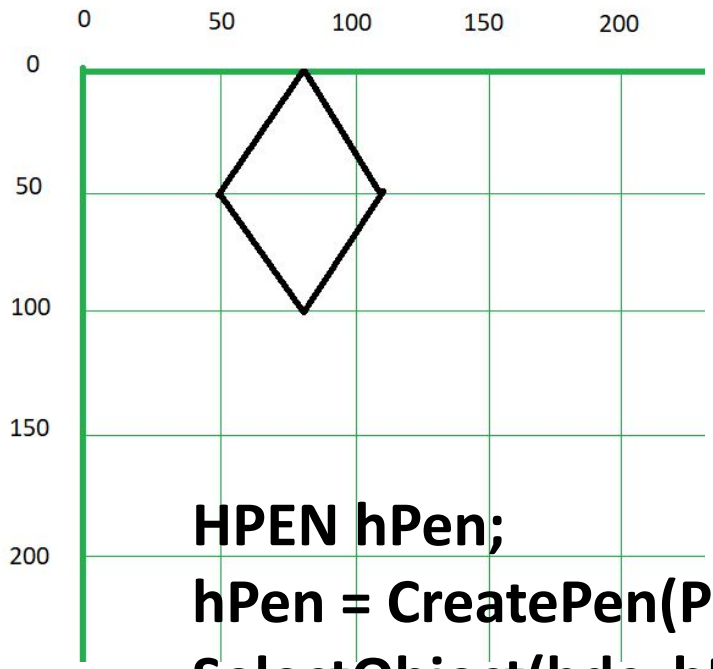
Ромб

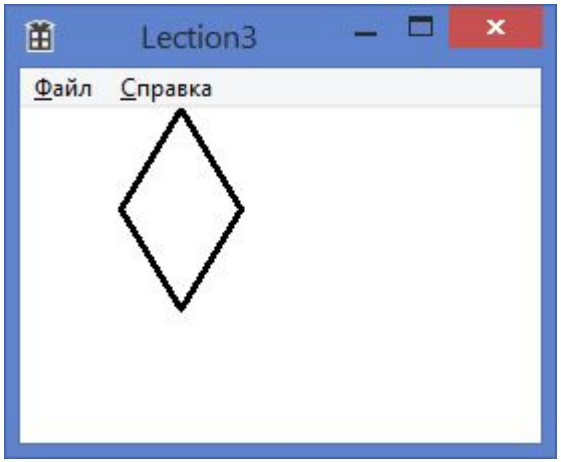





```
// Рисуем ромб
MoveToEx(hdc, 80, 0, NULL);
LineTo(hdc, 50, 50);
LineTo(hdc, 80, 100);
LineTo(hdc, 110, 50);
LineTo(hdc, 80, 0);
```

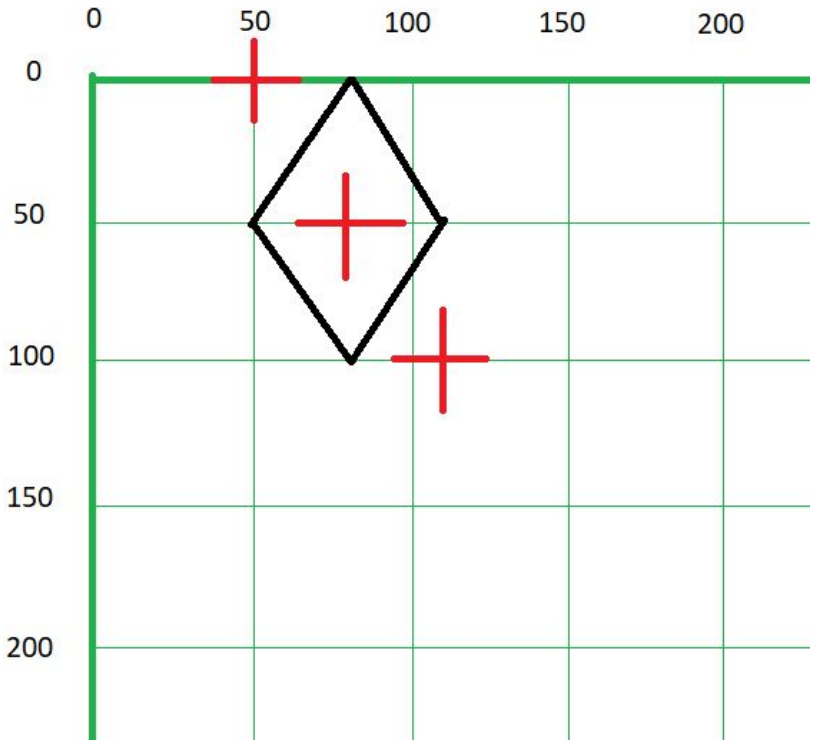
Ромб





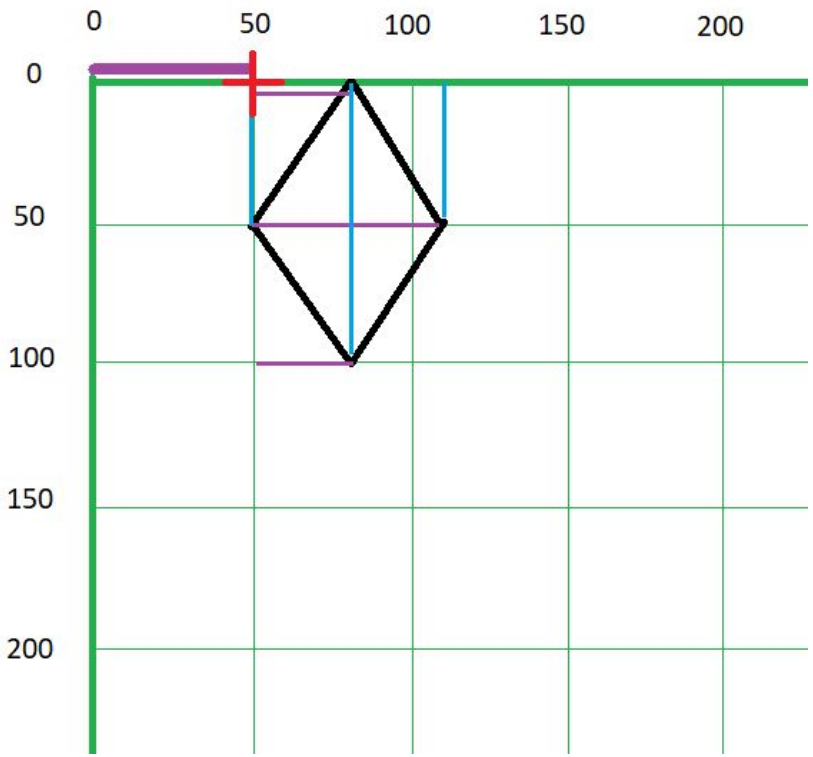

```
HPEN hPen;
hPen = CreatePen(PS_SOLID, 3, RGB(0, 0, 0));
SelectObject(hdc, hPen);
// Рисуем ромб
MoveToEx(hdc, 80, 0, NULL);
LineTo(hdc, 50, 50);
LineTo(hdc, 80, 100);
LineTo(hdc, 110, 50);
LineTo(hdc, 80, 0);
```

Относительные координаты



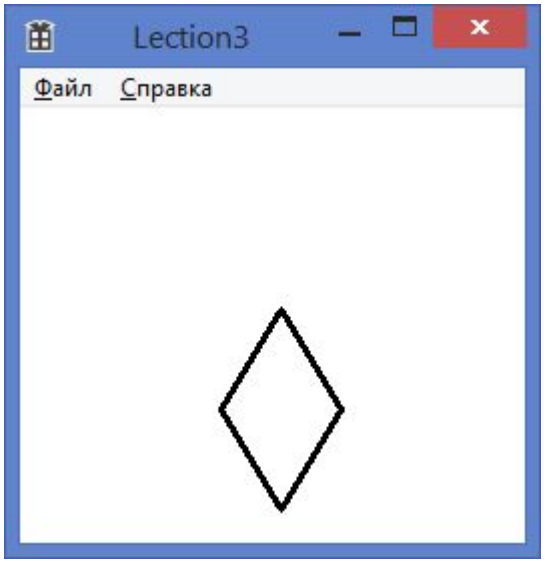

```
int x = 50;
int y = 0;
MoveToEx(hdc, x + 30, y, NULL);
LineTo(hdc, x, y + 50);
LineTo(hdc, x + 30, y + 100);
LineTo(hdc, x + 60, y + 50);
LineTo(hdc, x + 30, y);
```

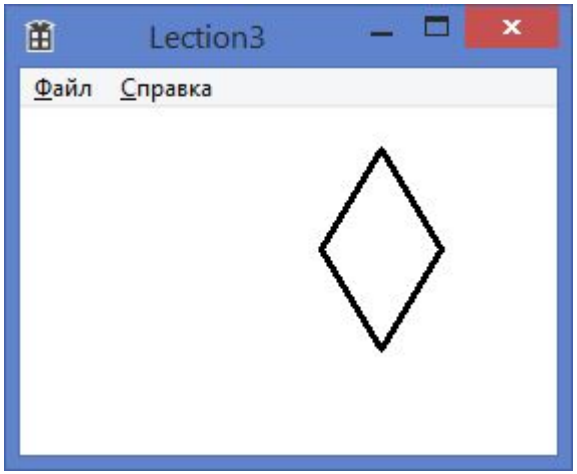
Относительные координаты




```
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);
```

Относительные координаты

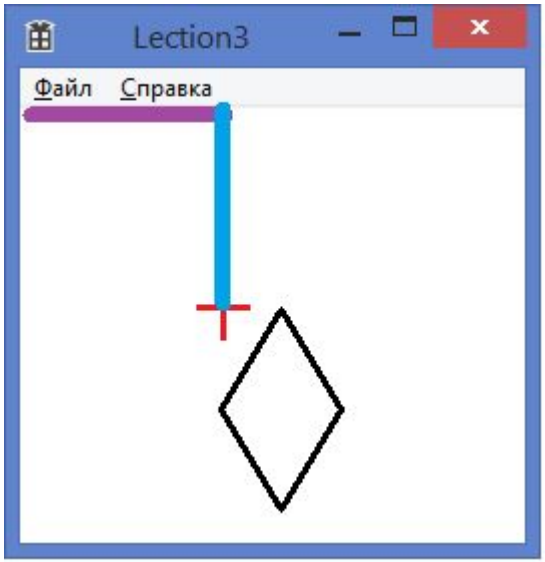


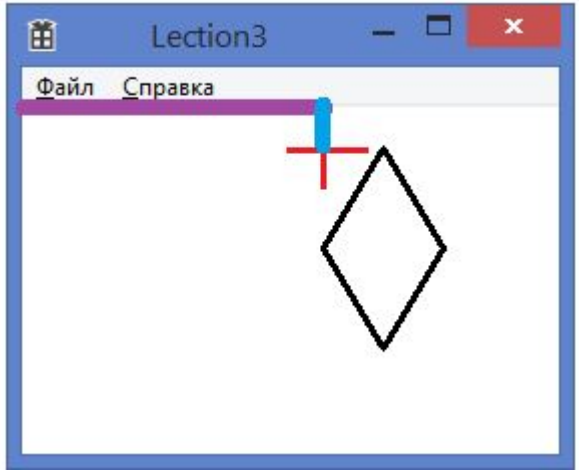



```
int x = 100;
int y = 100;
MoveToEx(hdc, x + 30, y, NULL);
LineTo(hdc, x, y + 50);
LineTo(hdc, x + 30, y + 100);
LineTo(hdc, x + 60, y + 50);
LineTo(hdc, x + 30, y);
```

```
int x = 150;
int y = 20;
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);
```

Относительные координаты

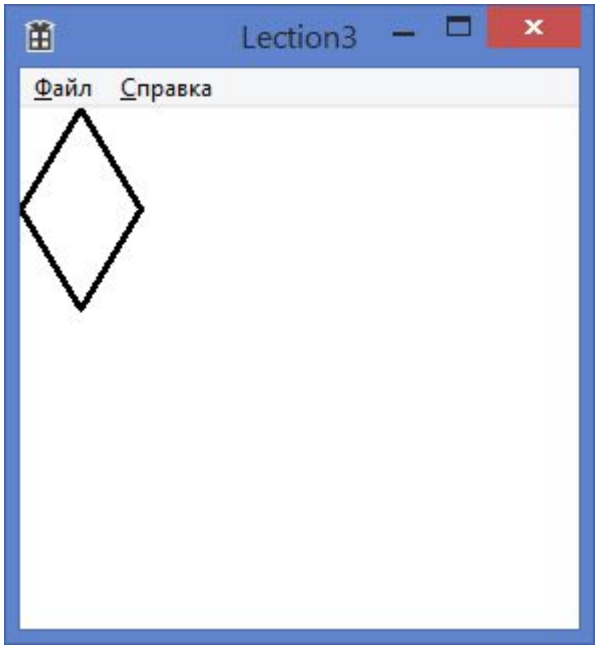


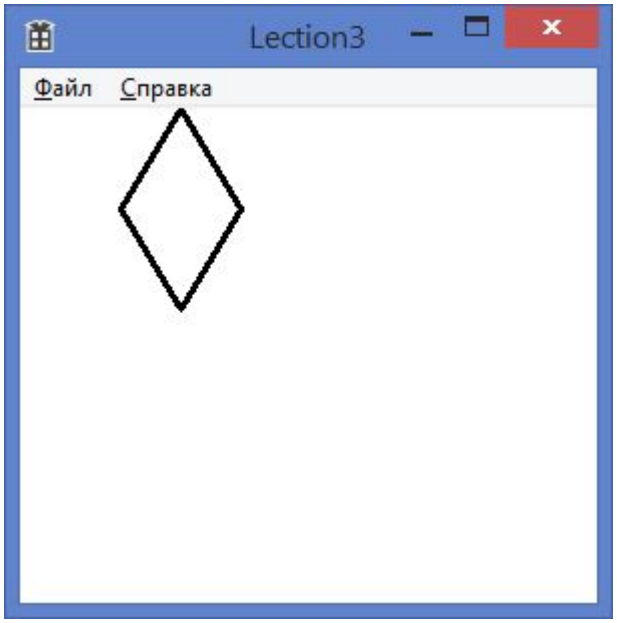


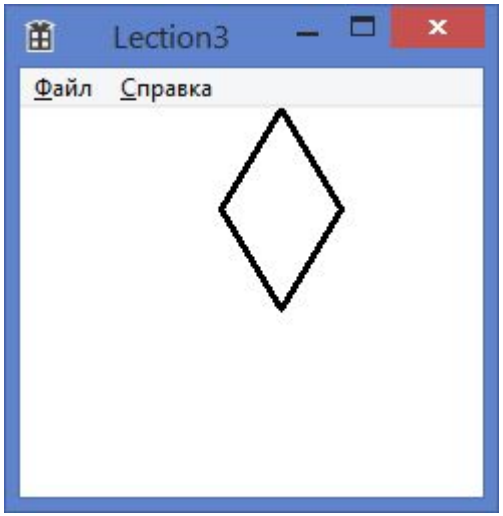

```
int x = 100;
int y = 100;
MoveToEx(hdc, x + 30, y, NULL);
LineTo(hdc, x, y + 50);
LineTo(hdc, x + 30, y + 100);
LineTo(hdc, x + 60, y + 50);
LineTo(hdc, x + 30, y);
```

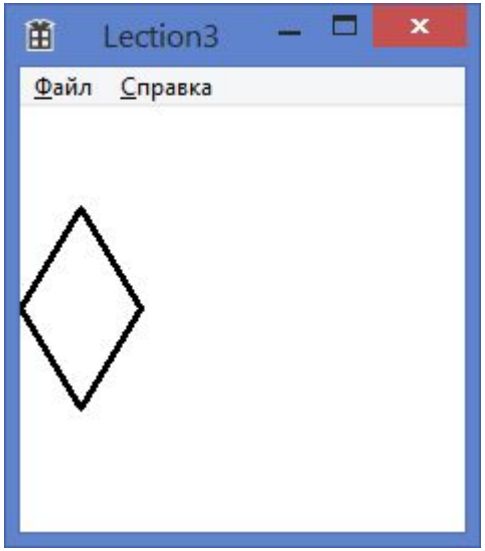
```
int x = 150;
int y = 20;
MoveToEx(hdc, x + 30, y, NULL);
LineTo(hdc, x, y + 50);
LineTo(hdc, x + 30, y + 100);
LineTo(hdc, x + 60, y + 50);
LineTo(hdc, x + 30, y);
```

Относительные координаты



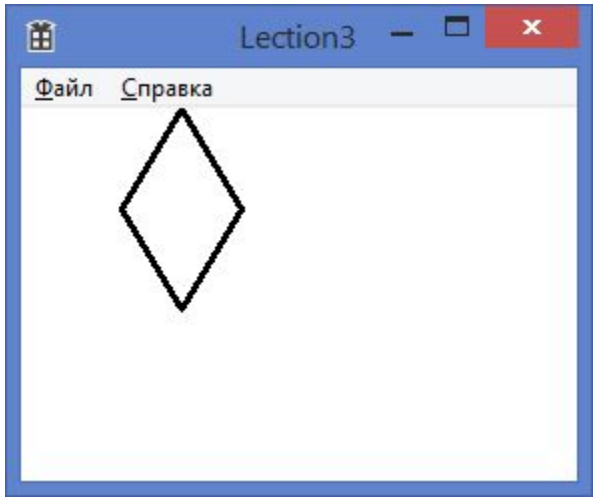






Отдельная функция для отрисовки ромба с заданным положением

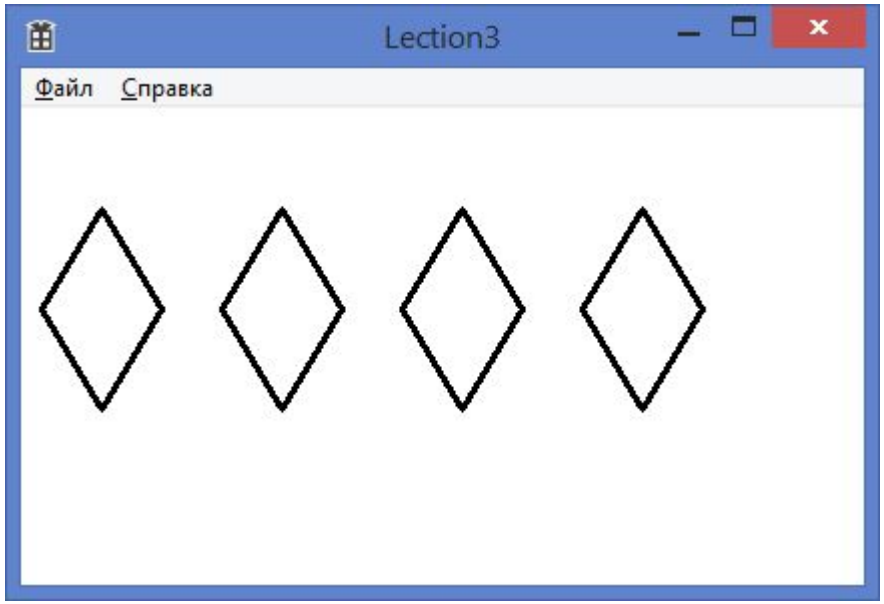
```
void Romb(HDC hdc, int x, int y) {
    MoveToEx(hdc, x + 30, y, NULL);
    LineTo(hdc, x, y + 50);
    LineTo(hdc, x + 30, y + 100);
    LineTo(hdc, x + 60, y + 50);
    LineTo(hdc, x + 30, y);
}
```




```
…
    HDC hdc = BeginPaint(hWnd, &ps);
    HPEN hPen;
    hPen = CreatePen(PS_SOLID, 3, RGB(0, 0, 0));
    SelectObject(hdc, hPen);
    Romb(hdc, 50, 0);
…
```

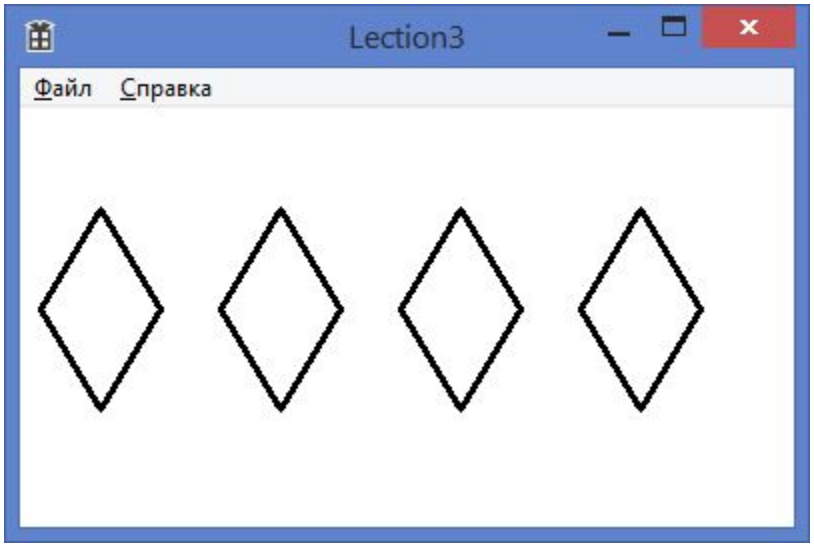
Рисуем при помощи нашей функции несколько ромбов в ряд

```
…
    HDC hdc = BeginPaint(hWnd, &ps);
    Romb(hdc, 10, 50);
    Romb(hdc, 100, 50);
    Romb(hdc, 190, 50);
    Romb(hdc, 280, 50);
…
```



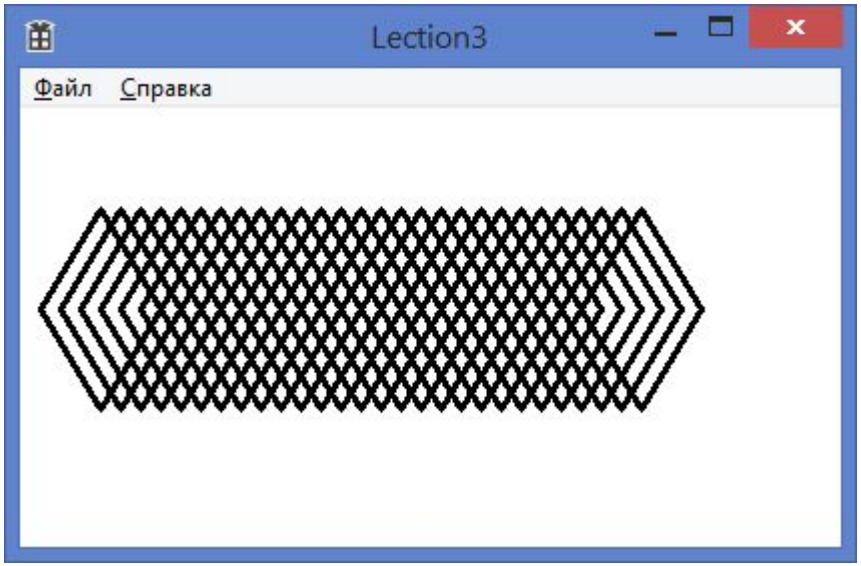
Рисуем при помощи нашей функции несколько ромбов – используем цикл

```
…
HPEN hPen;
hPen = CreatePen(PS_SOLID, 3, RGB(0, 0, 0));
SelectObject(hdc, hPen);

int x = 10;
int y = 50;
do {
    Romb(hdc, x, y);
    x += 90;
} while (x <= 280);
…
```



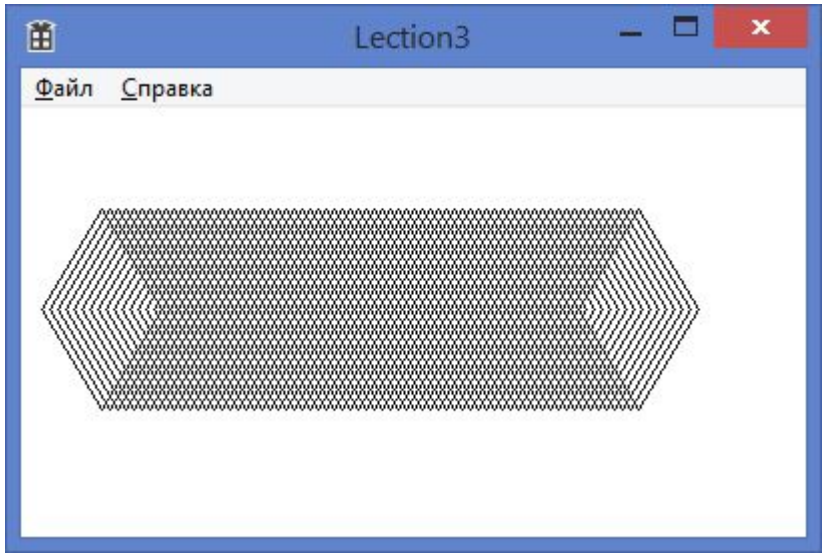
Рисуем при помощи нашей функции несколько ромбов – при помощи цикла

```
…
HPEN hPen;
hPen = CreatePen(PS_SOLID, 3, RGB(0, 0, 0));
SelectObject(hdc, hPen);

int x = 10;
int y = 50;
do {
    Romb(hdc, x, y);
    x += 10;
} while (x <= 280);
…
```



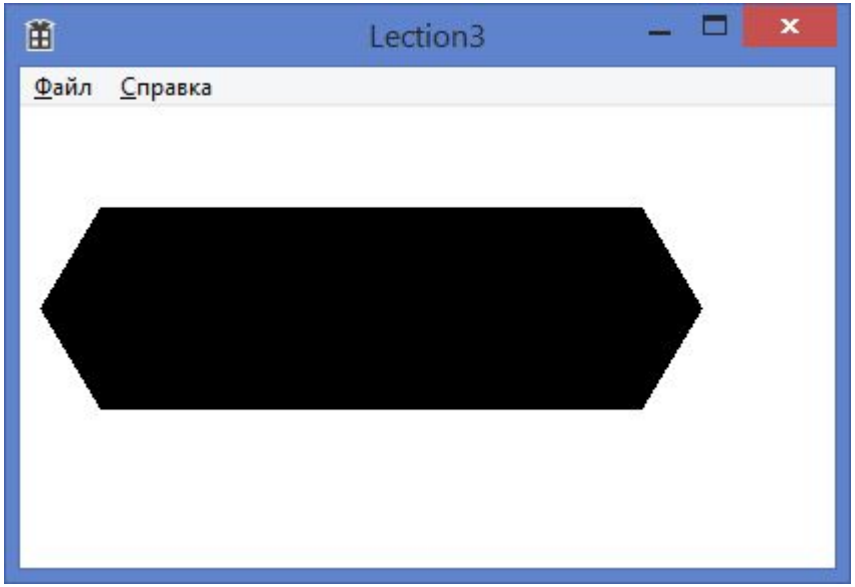
Рисуем при помощи нашей функции несколько ромбов – при помощи цикла

```
…
HPEN hPen;
hPen = CreatePen(PS_SOLID, 1, RGB(0, 0, 0));
SelectObject(hdc, hPen);

int x = 10;
int y = 50;
do {
    Romb(hdc, x, y);
    x += 4;
} while (x <= 280);
…
```



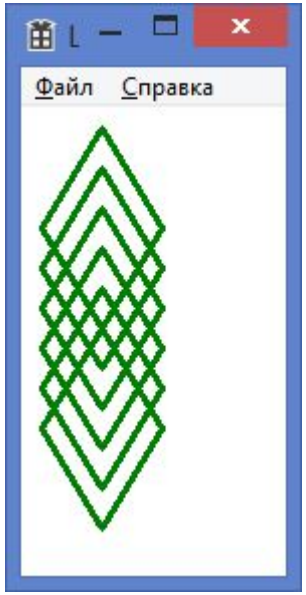
Рисуем при помощи нашей функции несколько ромбов – при помощи цикла

```
…
HPEN hPen;
hPen = CreatePen(PS_SOLID, 1, RGB(0, 0, 0));
SelectObject(hdc, hPen);

int x = 10;
int y = 50;
do {
    Romb(hdc, x, y);
    x += 1;
} while (x <= 280);…
```



Трассировка циклического алгоритма

```
…
HPEN hPen;
hPen = CreatePen(PS_SOLID, 3, RGB(0, 128, 0));
SelectObject(hdc, hPen);

int x = 10;
int y = 10;
do {
    Romb(hdc, x, y);
    y += 20;
} while (y <= 120);
```



Трассировка циклического алгоритма (2)
Установка точки останова

Lections2016 - Microsoft Visual Studio Expres...

Файл Изменить Просмотр Проект Сборка Отладка Команда Сервис Тест Олег Власенко
Окно Справка

Debug x86 Локальный отладчик Windows

Lection3.cpp Win32Project1.cpp Source1.cpp

Lection3 (Глобальная область) WndProc(HWND hWnd, UINT message,

```
            // TODO: Добавьте сюда любой код прорисовки, использующий
            HPEN hPen;
            hPen = CreatePen(PS_SOLID, 3, RGB(0, 128, 0));
            SelectObject(hdc, hPen);

            int x = 10;
            int y = 10;
            do {
                Romb(hdc, x, y);
                y += 20;
            } while (y <= 120);


            EndPaint(hWnd, &ps);
        }
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
```

132 %

Вывод

Готово Строка 168 Столбец 24 Знак 15 ВСТ

Трассировка циклического алгоритма (3)

Lections2016 - Microsoft Visual Studio Expres...

Быстрый запуск (Ctrl+Q)

Файл Изменить Просмотр Проект Сборка Отладка Команда Сервис Тест Олег Власенко

Окно Справка

Debug x86 Локальный отладчик Windows

Lection3.cpp Win32Project1.cpp Source1.cpp

Lection3 (Глобальная область) WndProc(HWND hWnd, UINT message,

```
// TODO: Добавьте сюда любой код прорисовки, использующий
HPEN hPen;
hPen = CreatePen(PS_SOLID, 3, RGB(0, 128, 0));
SelectObject(hdc, hPen);

int x = 10;
int y = 10;
do

} w

End
}
break;
case WM_DES
PostQui
```

Быстрые действия...	Ctrl+ю
Вставить фрагмент...	Ctrl+K, X
Разместить во фрагменте...	Ctrl+K, S
Перейти к определению	F12
Перейти к объявлению	Ctrl+Alt+F12
Найти все ссылки	Ctrl+K, R
Просмотреть иерархию вызовов	Ctrl+K, Ctrl+T
Переключение между файлами заголовков и кода	Ctrl+K, Ctrl+O
Точка останова	
Выполнить до текущей позиции	Ctrl+F10
Выполнить помеченные потоки до курсора	

132 %

Трассировка циклического алгоритма (4)

Lection3.cpp | Win32Project1.cpp | Source1.cpp

Lection3 | (Глобальная область) | WndProc(HWND hWnd, UINT message,

```
// TODO: Добавьте сюда любой код прорисовки, использующий
HPEN hPen;
hPen = CreatePen(PS_SOLID, 3, RGB(0, 128, 0));
SelectObject(hdc, hPen);

int x = 10;
```

Быстрые действия...	Ctrl+ю
Вставить фрагмент...	Ctrl+K, X
Разместить во фрагменте...	Ctrl+K, S
Перейти к определению	F12
Перейти к объявлению	Ctrl+Alt+F12
Найти все ссылки	Ctrl+K, R
Просмотреть иерархию вызовов	Ctrl+K, Ctrl+T
Переключение между файлами заголовков и кода	Ctrl+K, Ctrl+O
Точка останова	
Выполнить до текущей позиции	Ctrl+F10
Выполнить помеченные потоки до курсора	
Вырезать	Ctrl+X
Копировать	Ctrl+C
Вставить	Ctrl+V
Структура	

- Вставить точку останова
- Вставить точку отслеживания

132 %

Вывод

Готово

ВСТ

Трассировка циклического алгоритма (5)

Lections2016 - Microsoft Visual Studio Expres...

Файл Изменить Просмотр Проект Сборка Отладка Команда Сервис Тест Окно Справка

Быстрый запуск (Ctrl+Q) Олег Власенко

Debug x86 Локальный отладчик Windows

Lection3.cpp Win32Project1.cpp Source1.cpp

Lection3 (Глобальная область) WndProc(HWND hWnd, UINT message,

```
            // TODO: Добавьте сюда любой код прорисовки, использующий
            HPEN hPen;
            hPen = CreatePen(PS_SOLID, 3, RGB(0, 128, 0));
            SelectObject(hdc, hPen);

            int x = 10;
            int y = 10;
            do {
                Romb(hdc, x, y);
                y += 20;
            } while (y <= 120);


            EndPaint(hWnd, &ps);
        }
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
```

132 %

Вывод

Готово Строка 168 Столбец 17 Знак 8 ВСТ

Трассировка циклического алгоритма (6)
Запуск трассировки

Lections2016 - Microsoft Visual Studio Expres...

Быстрый запуск (Ctrl+Q)

Файл Изменить Просмотр Проект Сборка Отладка Команда Сервис Тест Олег Власенко

Окно Справка

Debug x86 Локальный отладчик Windows

Lection3.cpp Win32Project1.cpp Source1.cpp

Lection3 (Глобальная область) WndProc(HWND hWnd, UINT message,

```
            // TODO: Добавьте сюда любой код прорисовки, использующий
            HPEN hPen;
            hPen = CreatePen(PS_SOLID, 3, RGB(0, 128, 0));
            SelectObject(hdc, hPen);

            int x = 10;
            int y = 10;
            do {
                Romb(hdc, x, y);
                y += 20;
            } while (y <= 120);


            EndPaint(hWnd, &ps);
        }
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
```

132 %

Вывод

Готово Строка 168 Столбец 17 Знак 8 ВСТ

Трассировка циклического алгоритма (7) (при)остановка в точке остановки

Lections2016 (Отладка) - Microsoft Visual St...

Файл Изменить Просмотр Проект Сборка Отладка Команда Сервис Тест Окно Справка

Быстрый запуск (Ctrl+Q) Олег Власенко

Debug x86 Продолжить

Процесс: [7264] Lection3.exe События жизненного цикла Поток: [9992] Основной поток

Lection3.cpp Win32Project1.cpp Source1.cpp

Lection3 (Глобальная область) WndProc(HWND hWnd, UINT message,

```
// TODO: Добавьте сюда любой код прорисовки, использующий
HPEN hPen;
hPen = CreatePen(PS_SOLID, 3, RGB(0, 128, 0));
SelectObject(hdc, hPen);

int x = 10;
int y = 10;
do {
    Romb(hdc, x, y);
```

132 %

Локальные

Имя	Значение	Тип
y	-858993460	int
hPen	0xa930090a {unused=??? }	HPEN_
ps	{hdc=0x4b0127ff {unused=??? } fEra	tagPAIN
hdc	0x4b0127ff {unused=??? }	HDC_*
hWnd	0x00150a46 {unused=??? }	HWND_
message	15	unsigne
wParam	0	unsigne
lParam	0	long

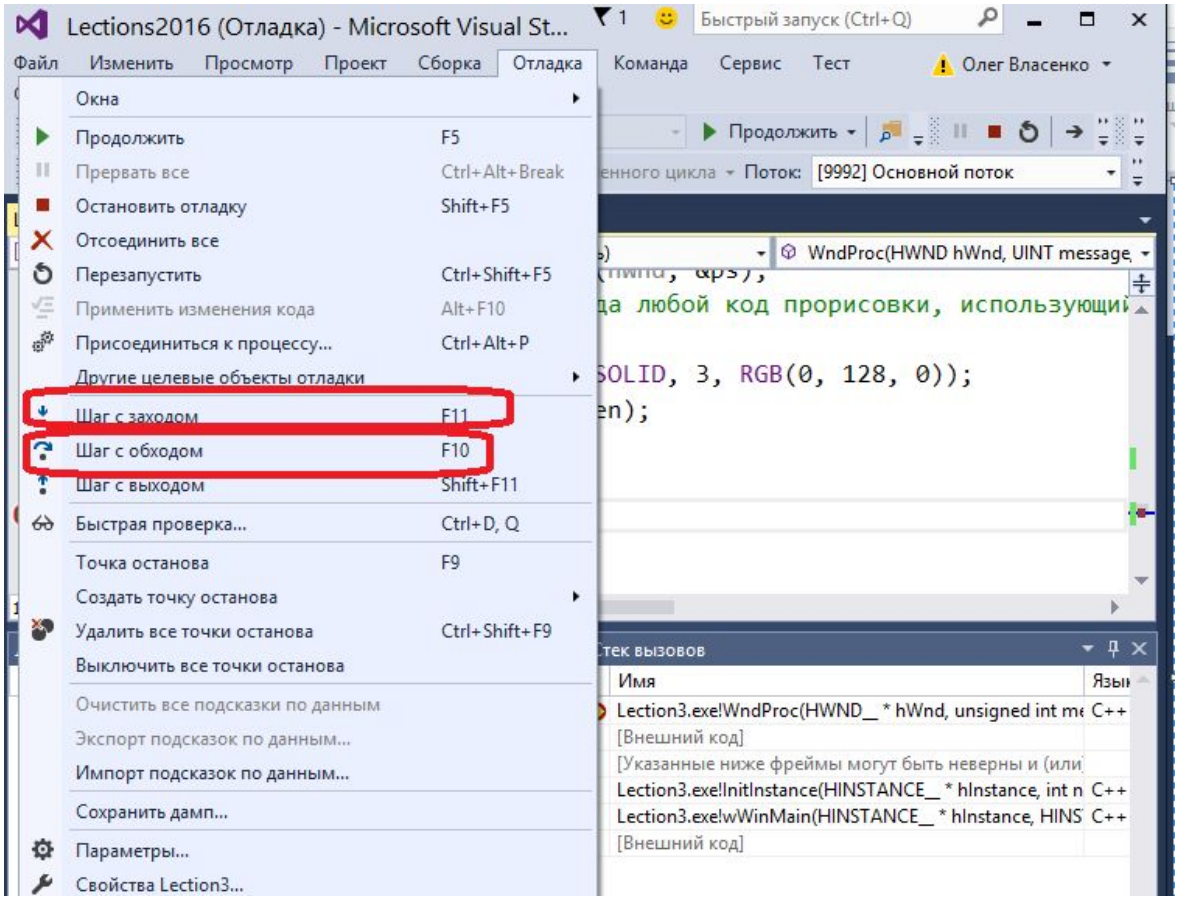
Локальные Контрольные значения 1

Стек вызовов

Имя	Язы
Lection3.exe!WndProc(HWND_* hWnd, unsigned int me	C++
[Внешний код]	
[Указанные ниже фреймы могут быть неверны и (или	
Lection3.exe!InitInstance(HINSTANCE_* hInstance, int n	C++
Lection3.exe!wWinMain(HINSTANCE_* hInstance, HINS	C++
[Внешний код]	

Загрузка символов для oleacc.dll Строка 168 Столбец 1 Знак 1 ВСТ

Трассировка циклического алгоритма (8)
Пошаговая трассировка



Трассировка циклического алгоритма (9)
Шаг сделан (F10)

Lections2016 (Отладка) - Microsoft Visual St...

Быстрый запуск (Ctrl+Q)

Файл Изменить Просмотр Проект Сборка Отладка Команда Сервис Тест Олег Власенко

Окно Справка

Debug x86 Продолжить

Процесс: [7264] Lection3.exe События жизненного цикла Поток: [9992] Основной поток

Lection3.cpp Win32Project1.cpp Source1.cpp

Lection3 (Глобальная область) WndProc(HWND hWnd, UINT message,

```
int x = 10;
int y = 10;
do {
    Romb(hdc, x, y);  ≤ 1 мс прошло
    y += 20;
} while (y <= 120);


EndPaint(hWnd, &ps);
```

132 %

Локальные

Имя	Значение	Тип
y	10	int
hPen	0xa930090a {unused=??? }	HPEN_
ps	{hdc=0x4b0127ff {unused=??? } fEra	tagPAIN
hdc	0x4b0127ff {unused=??? }	HDC_*
hWnd	0x00150a46 {unused=??? }	HWND_
message	15	unsigne
wParam	0	unsigne
lParam	0	long

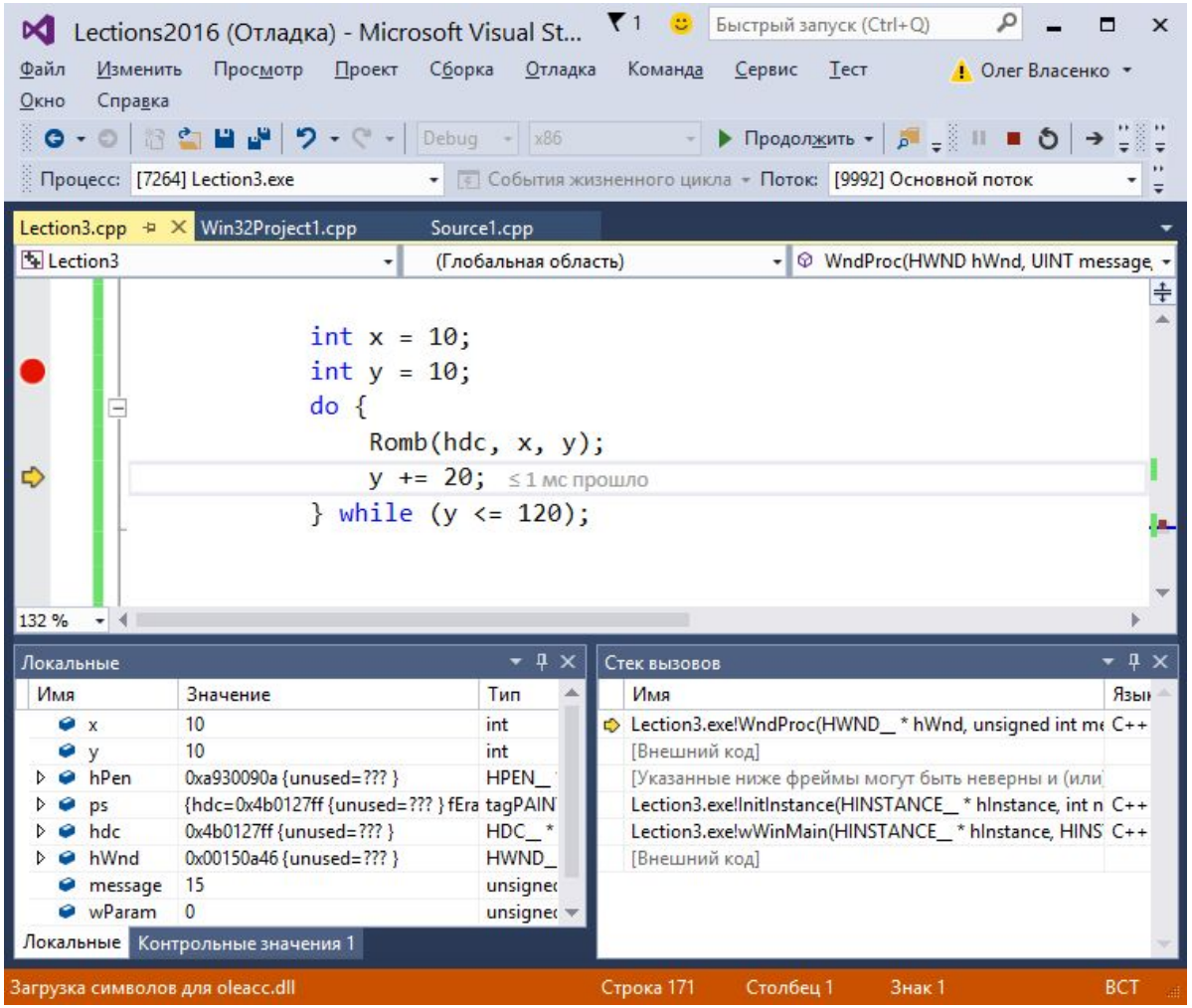
Локальные Контрольные значения 1

Стек вызовов

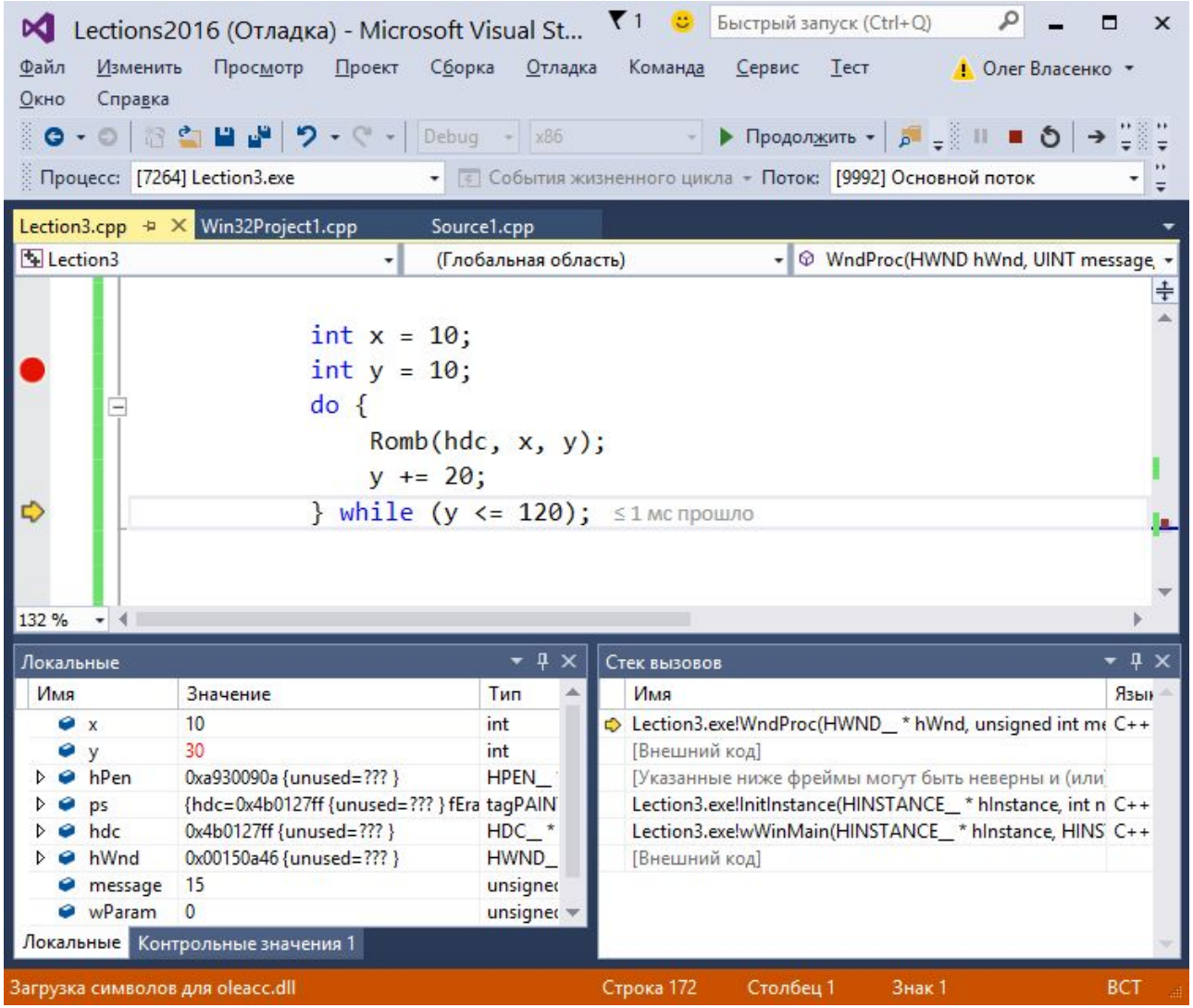
Имя	Язы
Lection3.exe!WndProc(HWND_* hWnd, unsigned int me	C++
[Внешний код]	
[Указанные ниже фреймы могут быть неверны и (или	
Lection3.exe!InitInstance(HINSTANCE_* hInstance, int n	C++
Lection3.exe!wWinMain(HINSTANCE_* hInstance, HINS	C++
[Внешний код]	

Загрузка символов для oleacc.dll Строка 170 Столбец 1 Знак 1 ВСТ

Трассировка циклического алгоритма (10)
Еще шаг сделан (F10)



Трассировка циклического алгоритма (11)
Еще шаг сделан (F10)



Трассировка циклического алгоритма (12)
Еще шаг сделан

Lections2016 (Отладка) - Microsoft Visual St...

Файл Изменить Просмотр Проект Сборка Отладка Команда Сервис Тест Окно Справка

Быстрый запуск (Ctrl+Q) Олег Власенко

Debug x86 Продолжить

Процесс: [7264] Lection3.exe События жизненного цикла Поток: [9992] Основной поток

Lection3.cpp Win32Project1.cpp Source1.cpp

Lection3 (Глобальная область) WndProc(HWND hWnd, UINT message,

```
int x = 10;
int y = 10;
do {
    Romb(hdc, x, y);  ≤ 1 мс прошло
    y += 20;
} while (y <= 120);
```

132 %

Локальные

Имя	Значение	Тип
x	10	int
y	30	int
hPen	0xa930090a {unused=??? }	HPEN__
ps	{hdc=0x4b0127ff {unused=??? } fEra	tagPAIN
hdc	0x4b0127ff {unused=??? }	HDC__ *
hWnd	0x00150a46 {unused=??? }	HWND_
message	15	unsignec
wParam	0	unsignec

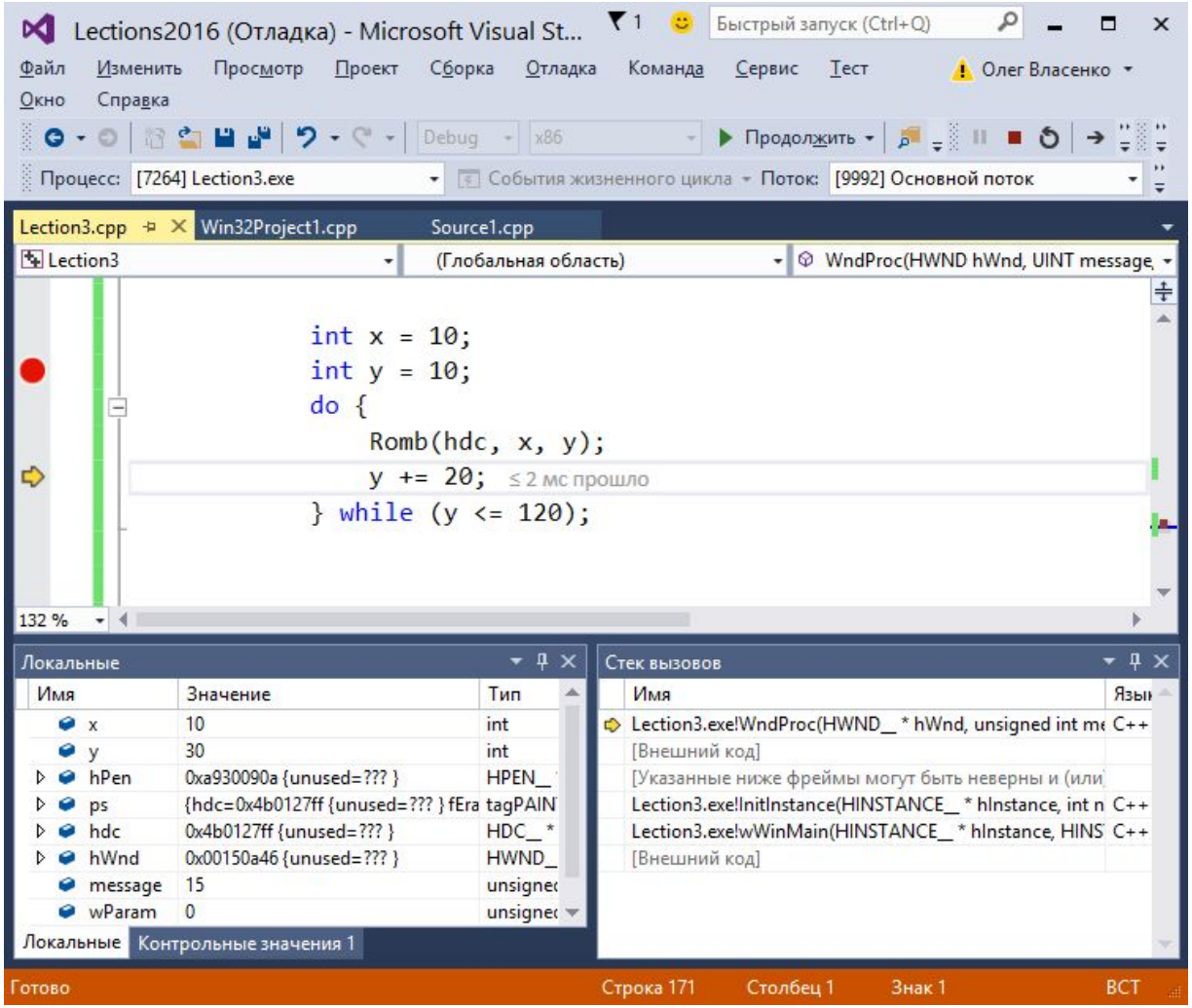
Локальные Контрольные значения 1

Стек вызовов

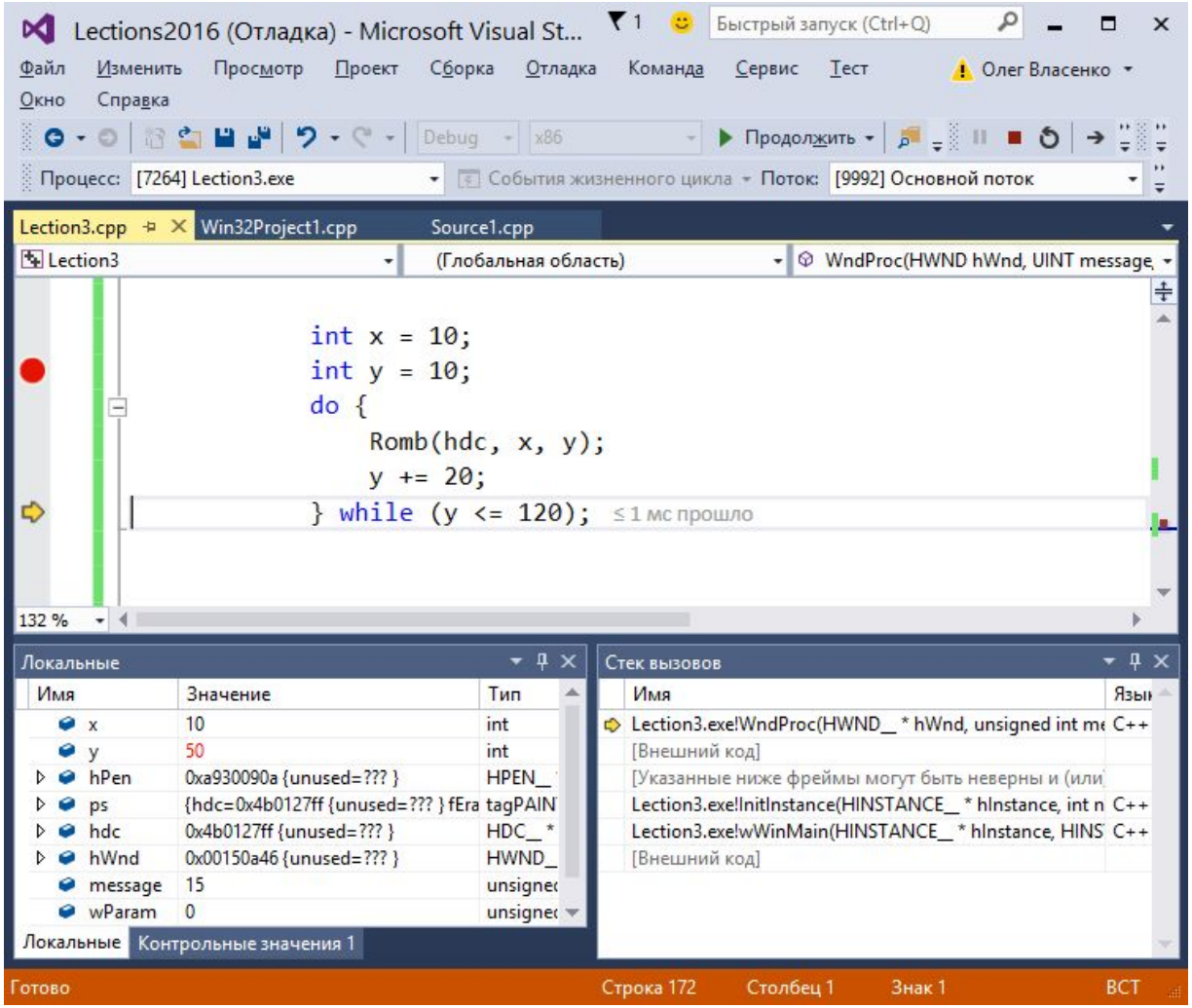
Имя	Язы
Lection3.exe!WndProc(HWND__ * hWnd, unsigned int me	C++
[Внешний код]	
[Указанные ниже фреймы могут быть неверны и (или	
Lection3.exe!InitInstance(HINSTANCE__ * hInstance, int n	C++
Lection3.exe!wWinMain(HINSTANCE__ * hInstance, HINS	C++
[Внешний код]	

Готово Строка 170 Столбец 1 Знак 1 ВСТ

Трассировка циклического алгоритма (13)
Еще шаг сделан



Трассировка циклического алгоритма (14)
Еще шаг сделан



Трассировка циклического алгоритма (15)
Еще шаг сделан

Lections2016 (Отладка) - Microsoft Visual St...

Быстрый запуск (Ctrl+Q)

Файл Изменить Просмотр Проект Сборка Отладка Команда Сервис Тест Олег Власенко

Окно Справка

Debug x86 Продолжить

Процесс: [7264] Lection3.exe События жизненного цикла Поток: [9992] Основной поток

Lection3.cpp Win32Project1.cpp Source1.cpp

Lection3 (Глобальная область) WndProc(HWND hWnd, UINT message,

```
int x = 10;
int y = 10;
do {
    Romb(hdc, x, y);   ≤ 1 мс прошло
    y += 20;
} while (y <= 120);
```

132 %

Локальные

Имя	Значение	Тип
x	10	int
y	50	int
hPen	0xa930090a {unused=??? }	HPEN_
ps	{hdc=0x4b0127ff {unused=??? } fEra	tagPAIN
hdc	0x4b0127ff {unused=??? }	HDC_*
hWnd	0x00150a46 {unused=??? }	HWND_
message	15	unsigne
wParam	0	unsigne

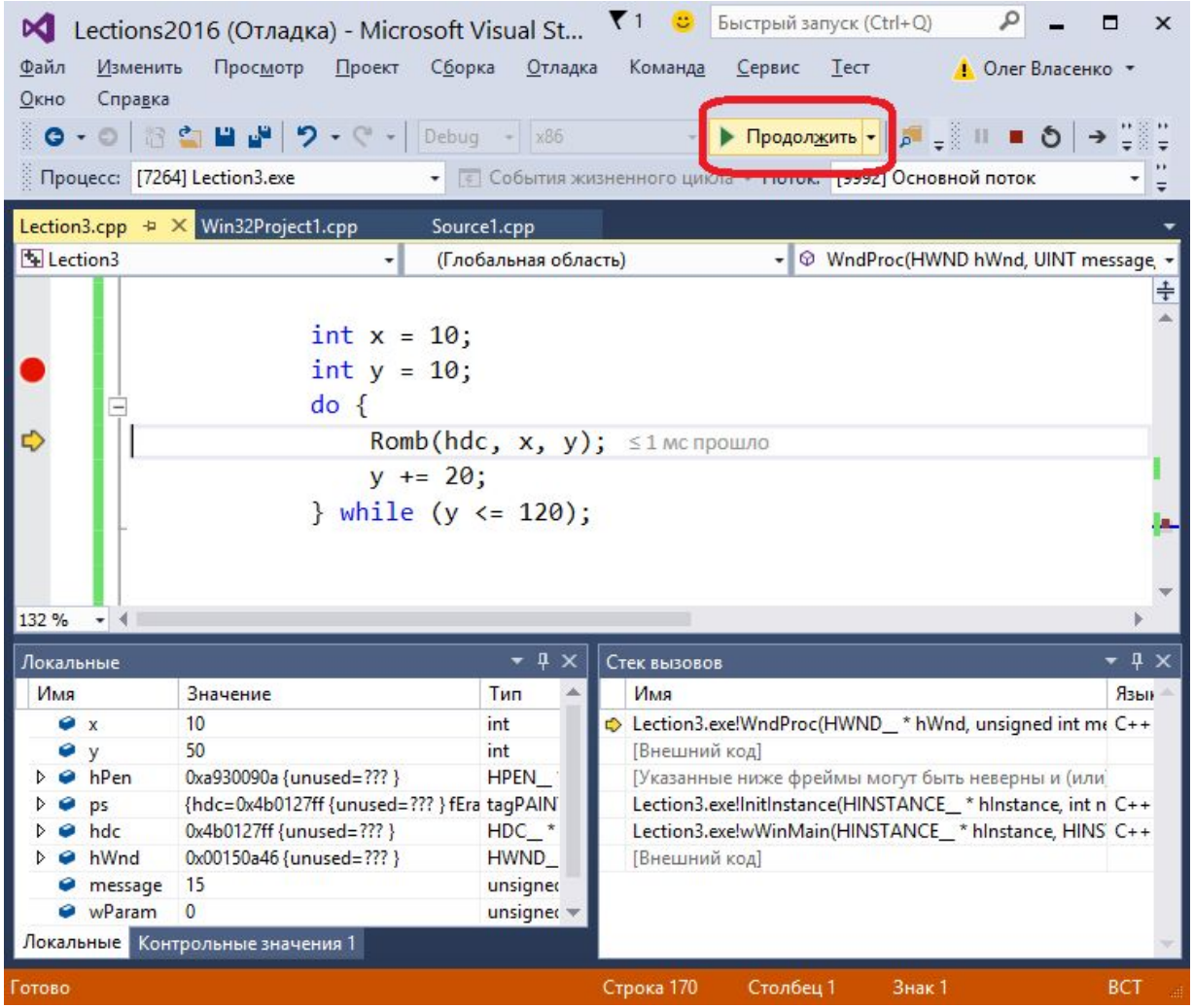
Локальные Контрольные значения 1

Стек вызовов

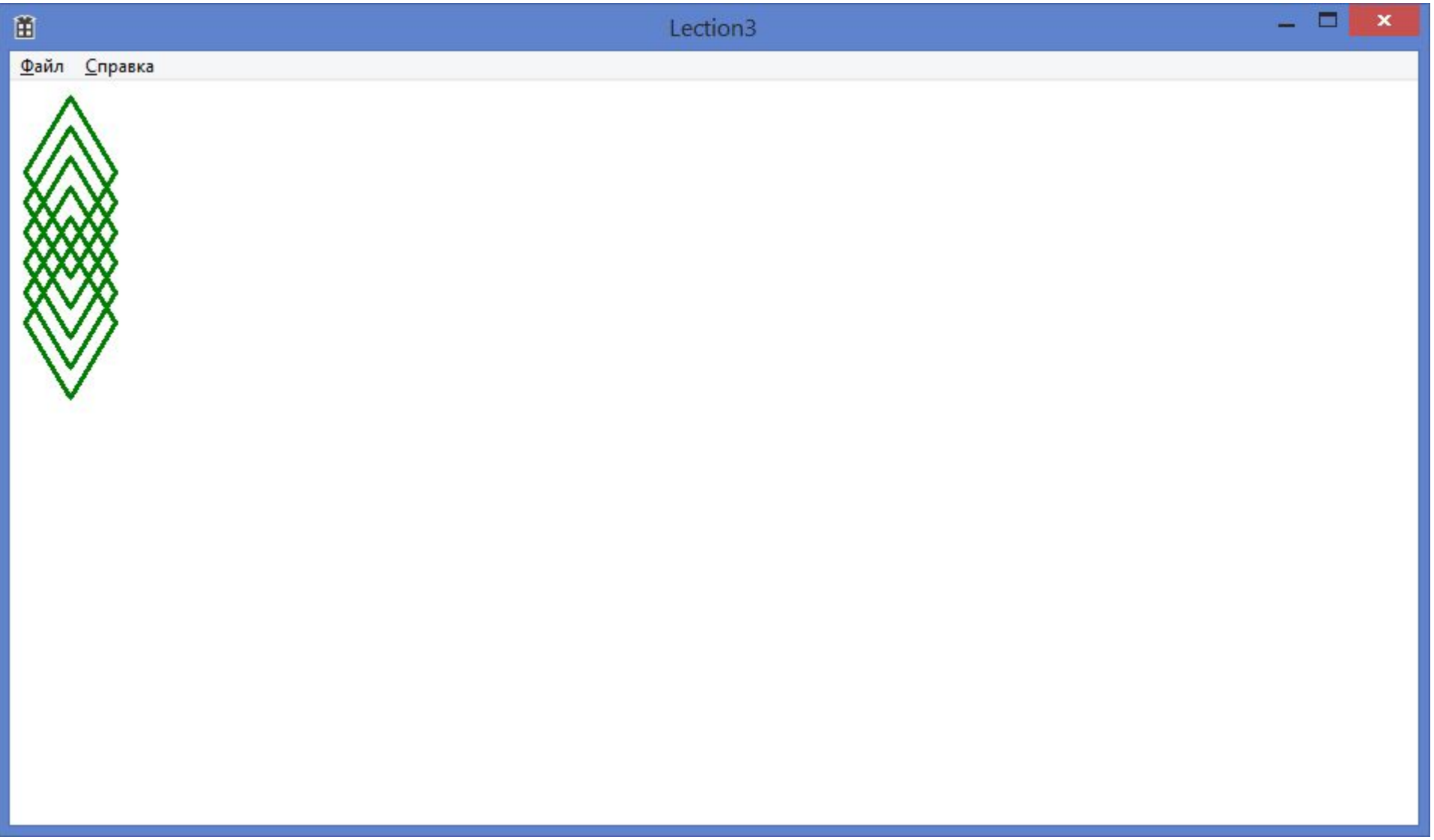
Имя	Язы
Lection3.exe!WndProc(HWND_* hWnd, unsigned int me	C++
[Внешний код]	
[Указанные ниже фреймы могут быть неверны и (или	
Lection3.exe!InitInstance(HINSTANCE_* hInstance, int n	C++
Lection3.exe!wWinMain(HINSTANCE_* hInstance, HINS	C++
[Внешний код]	

Готово Строка 170 Столбец 1 Знак 1 ВСТ

Трассировка циклического алгоритма (16)
Возобновить выполнение

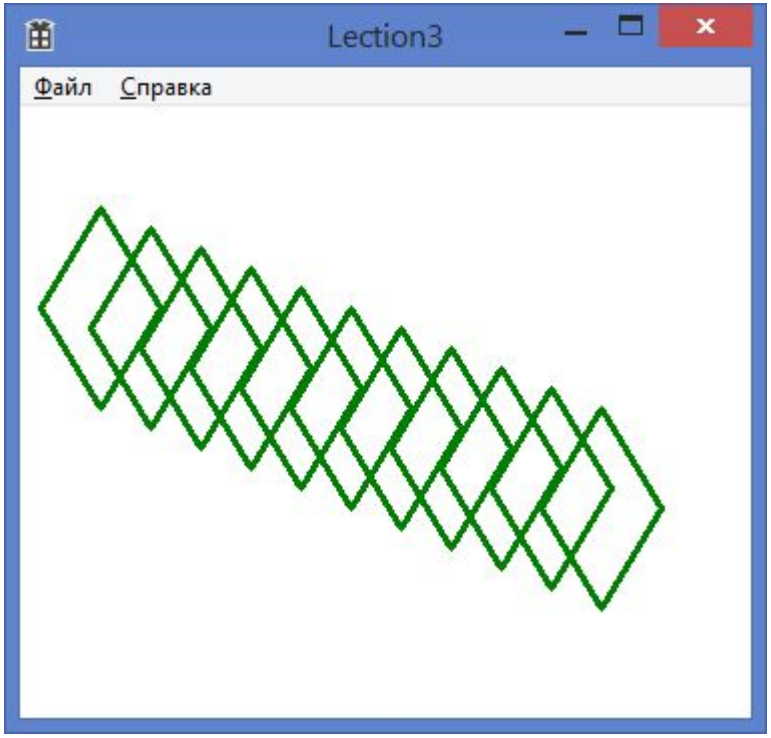


Трассировка циклического алгоритма (17)
Приложение работает!

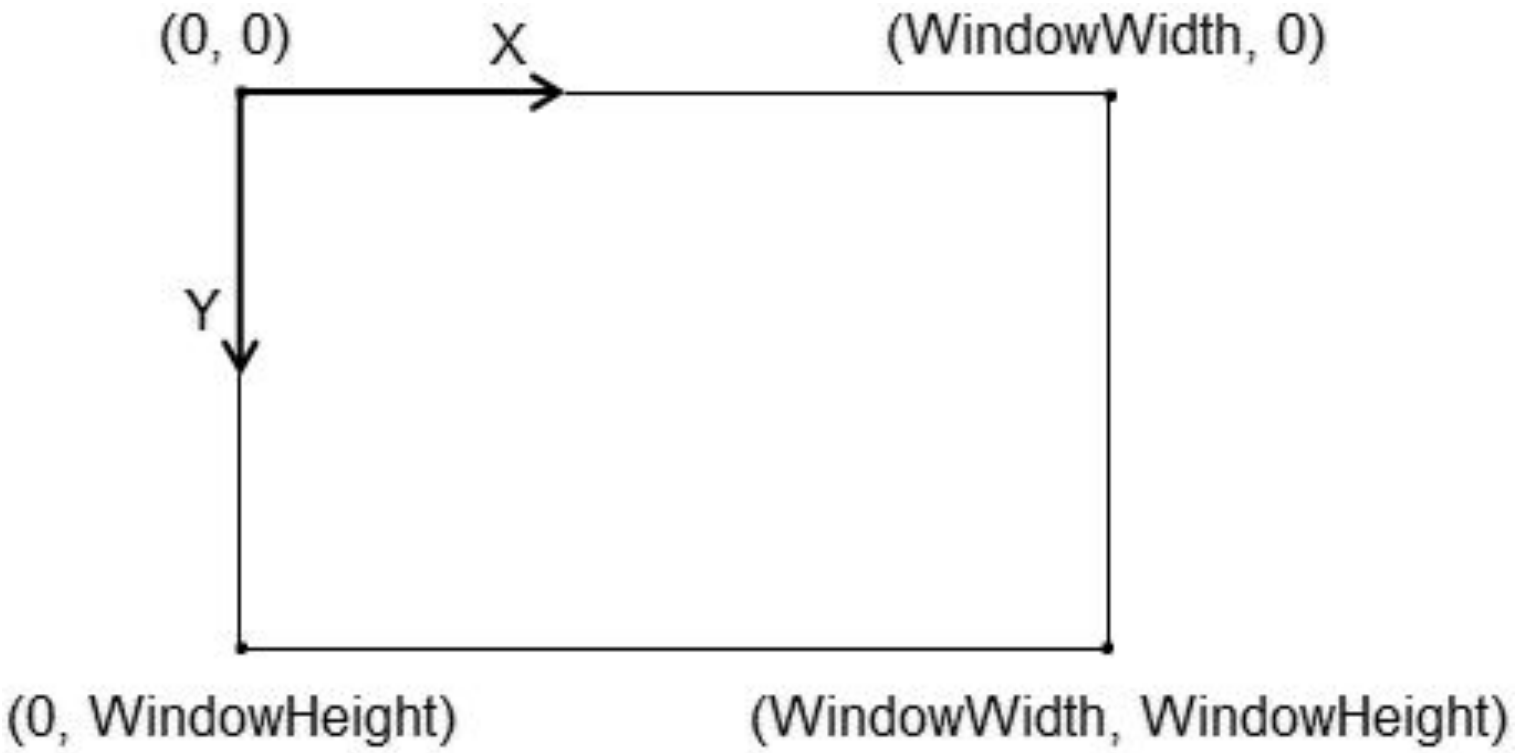


Изменяем сразу и X и Y

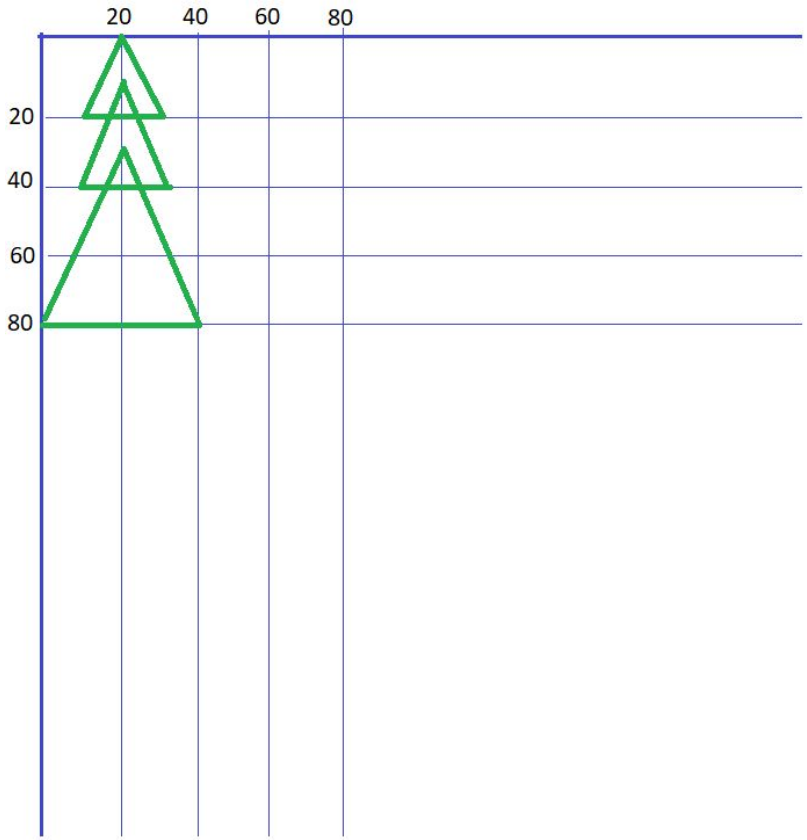
```
…
int x = 10;
int y = 50;

do {
    Romb(hdc, x, y);
    x += 25;
    y += 10;

} while (x <= 280);
…
```



Автомобиль деда Мороза - логотип

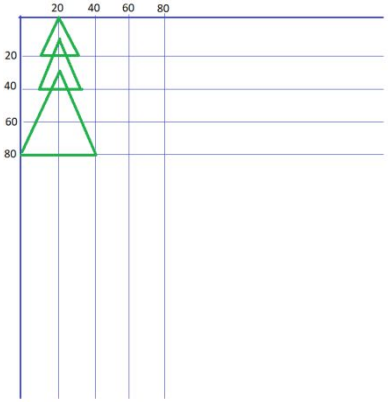


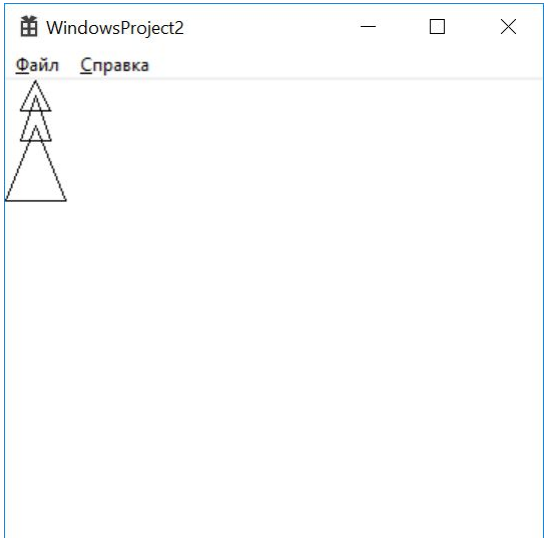
Автомобиль деда Мороза - логотип



Автомобиль деда Мороза - логотип

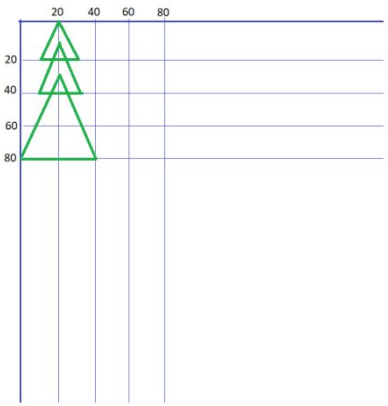
```
case WM_PAINT:
    {
        PAINTSTRUCT ps;
        HDC hdc = BeginPaint(hWnd, &ps);
        // TODO: Добавьте сюда любой код прорисовки, использующий HDC...

        // верхний треугольник
        MoveToEx(hdc, 20, 0, NULL);
        LineTo(hdc, 30, 20);
        LineTo(hdc, 10, 20);
        LineTo(hdc, 20, 0);

        // средний треугольник
        MoveToEx(hdc, 20, 10, NULL);
        LineTo(hdc, 30, 40);
        LineTo(hdc, 10, 40);
        LineTo(hdc, 20, 10);

        // нижний треугольник
        MoveToEx(hdc, 20, 30, NULL);
        LineTo(hdc, 40, 80);
        LineTo(hdc, 0, 80);
        LineTo(hdc, 20, 30);

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

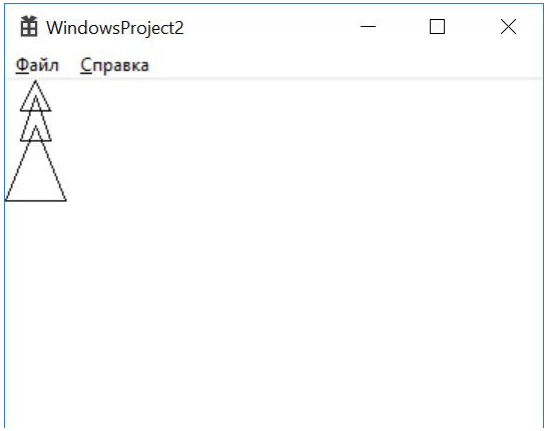




Вынесли код в отдельную функцию

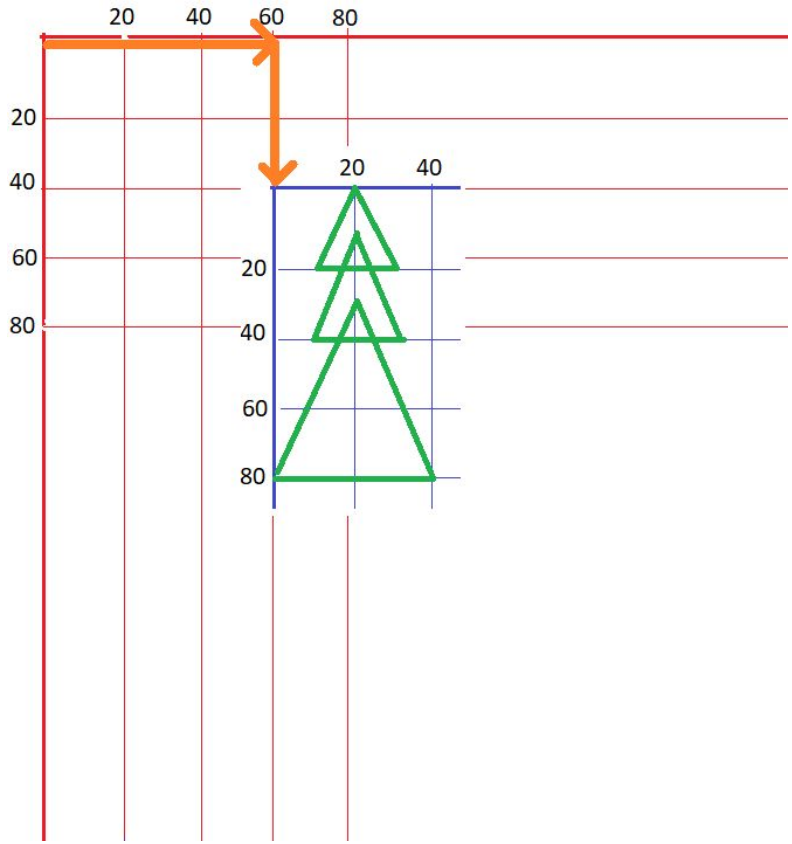
```
void StClausAuto(HDC hdc) {
        // верхний треугольник
        MoveToEx(hdc, 20, 0, NULL);
        LineTo(hdc, 30, 20);
        LineTo(hdc, 10, 20);
        LineTo(hdc, 20, 0);

        // средний треугольник
        MoveToEx(hdc, 20, 10, NULL);
        LineTo(hdc, 30, 40);
        LineTo(hdc, 10, 40);
        LineTo(hdc, 20, 10);

        // нижний треугольник
        MoveToEx(hdc, 20, 30, NULL);
        LineTo(hdc, 40, 80);
        LineTo(hdc, 0, 80);
        LineTo(hdc, 20, 30);
}
…
```

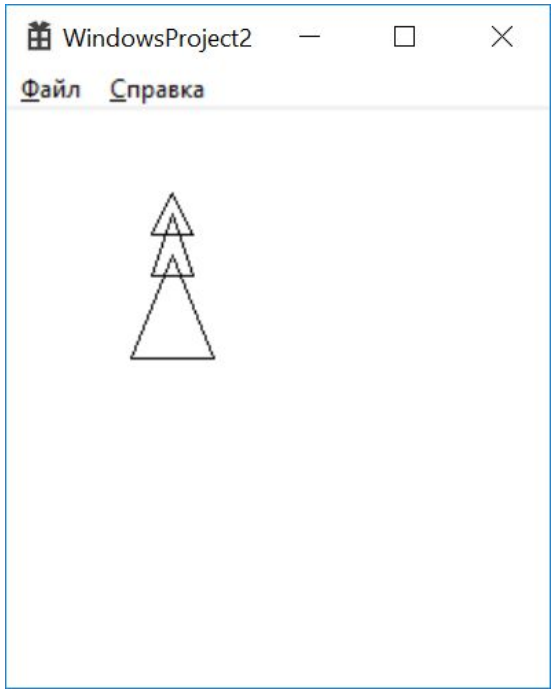





```
case WM_PAINT:
    {
        PAINTSTRUCT ps;
        HDC hdc = BeginPaint(hWnd, &ps);
        // TODO: Добавьте сюда любой код прорисовки, использующий HDC...

        StClausAuto(hdc);

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

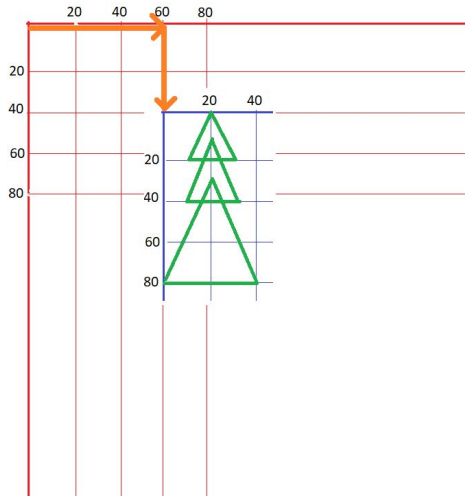
Относительные координаты

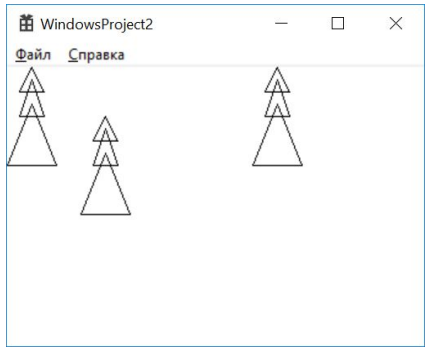




Относительные координаты

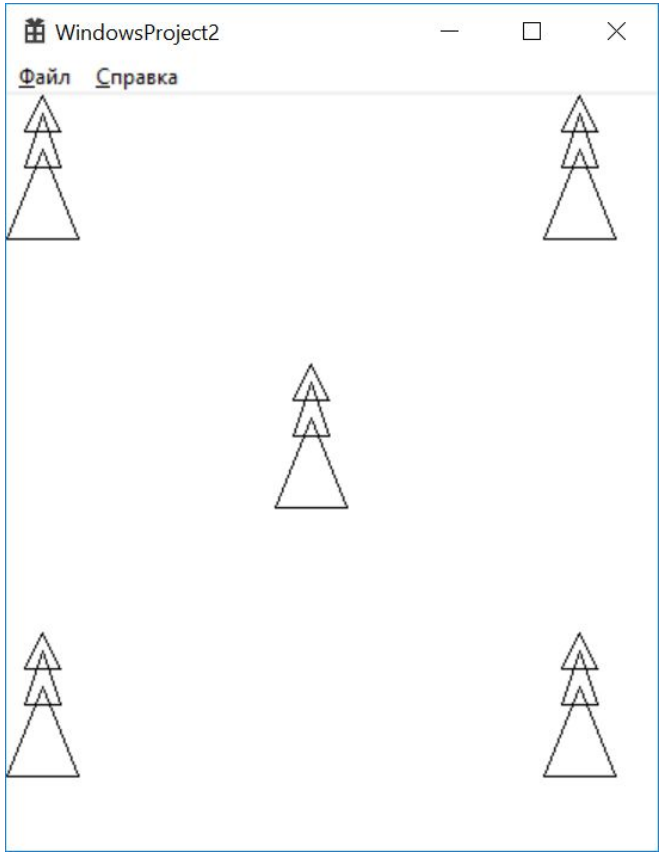
```
void StClausAuto(HDC hdc, int x, int y) {
        // верхний треугольник
        MoveToEx(hdc, 20 + x, 0 + y, NULL);
        LineTo(hdc, 30 + x, 20 + y);
        LineTo(hdc, 10 + x, 20 + y);
        LineTo(hdc, 20 + x, 0 + y);

        // 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);
}
...
```





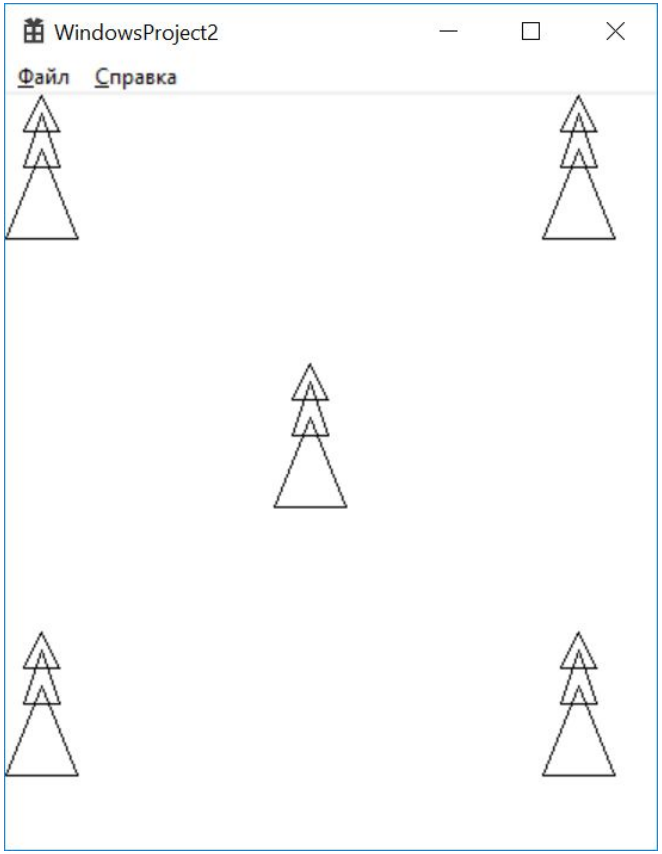

```
case WM_PAINT:
    {
        PAINTSTRUCT ps;
        HDC hdc = BeginPaint(hWnd, &ps);
        // TODO: Добавьте сюда любой код прорисовки, использующий HDC...

        StClausAuto(hdc, 60, 40);
        StClausAuto(hdc, 0, 0);
        StClausAuto(hdc, 200, 0);

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

Задача 1.1: 5 логотипов по углам и в центре

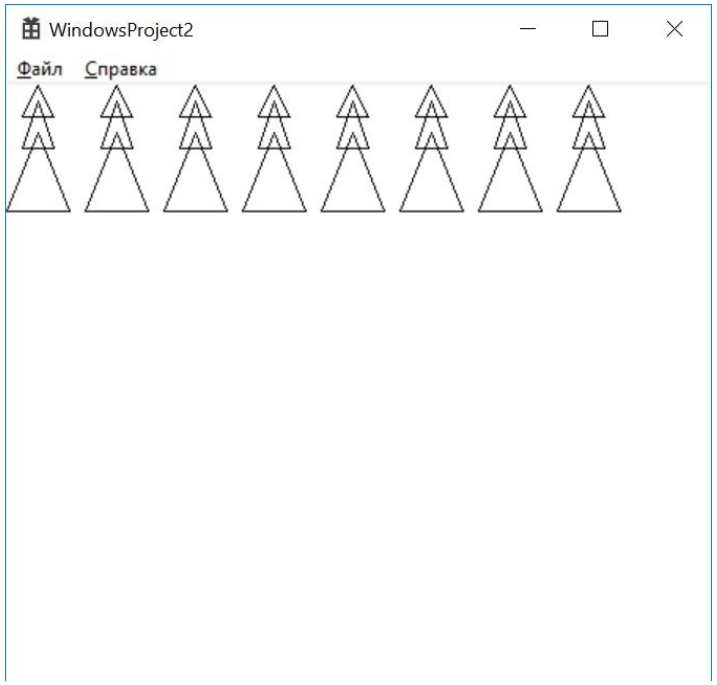


Задача 1.1: 5 логотипов по углам и в центре

```
247     case WM_PAINT:
248     {
249         PAINTSTRUCT ps;
250         HDC hdc = BeginPaint(hWnd, &ps);
251         // TODO: Добавьте сюда любой код прорисовки, использующий HDC...
252
253         StClausAuto(hdc, 0, 0);
254         StClausAuto(hdc, 200, 200);
255         StClausAuto(hdc, 100, 100);
256         StClausAuto(hdc, 0, 200);
257         StClausAuto(hdc, 200, 0);
258
259         EndPaint(hWnd, &ps);
260     }
261     break;
```

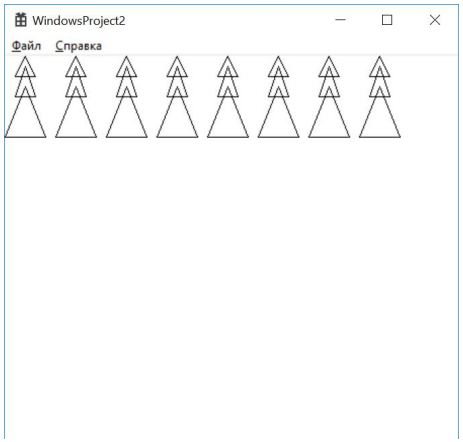


Задача 1.2: 8 логотипов в горизонтальную линию

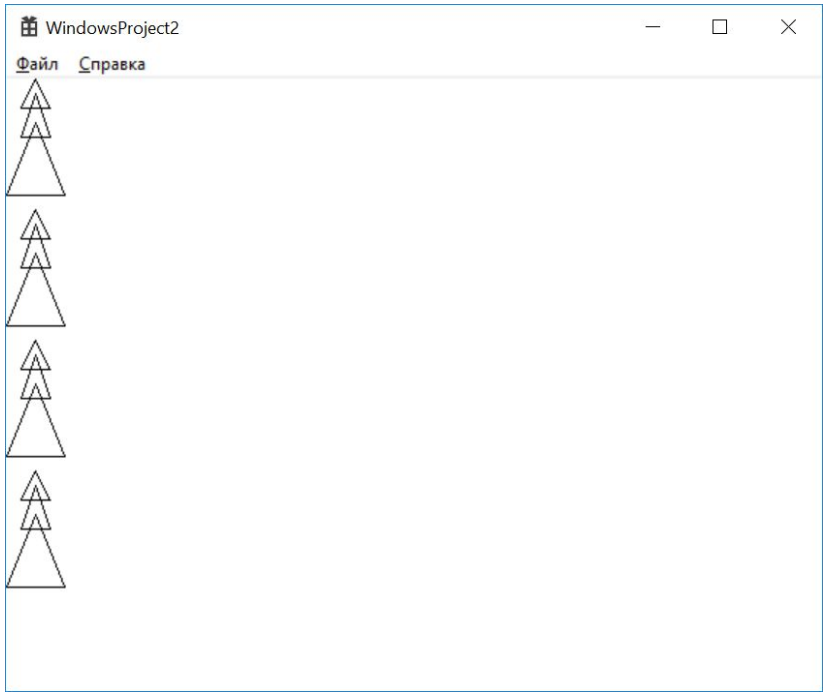


Задача 1.2: 8 логотипов в горизонтальную линию
Решение

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

        int x = 0;
        do {
            StClausAuto(hdc, x, 0);
            x += 50;
        } while (x < 400);

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

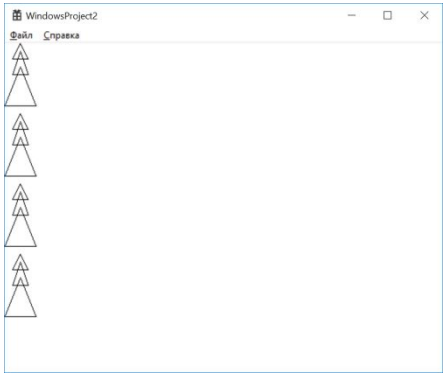


Задача 1.3: 4 логотипа в вертикальную линию

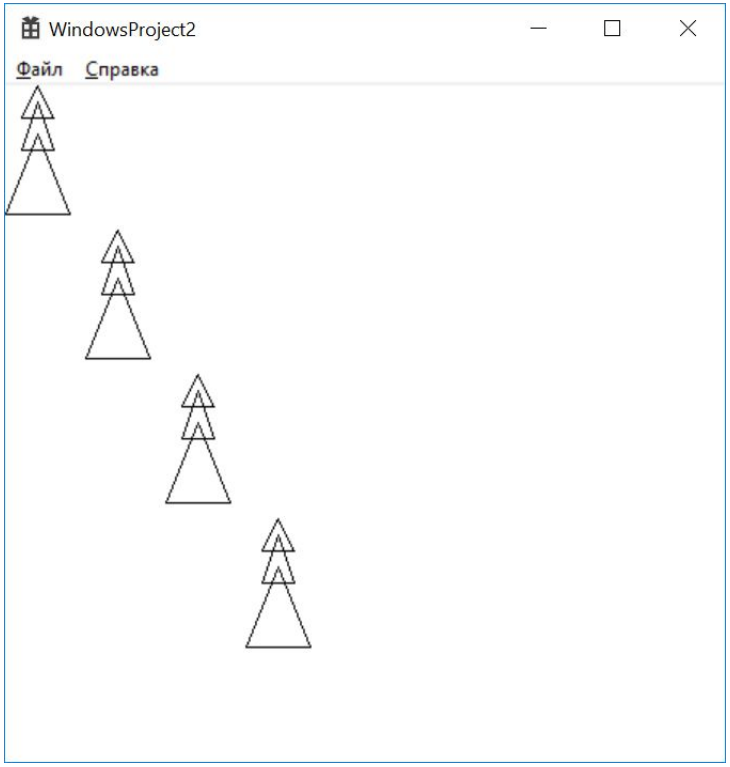


Задача 1.3 - РЕШЕНИЕ:
4 логотипа в вертикальную линию

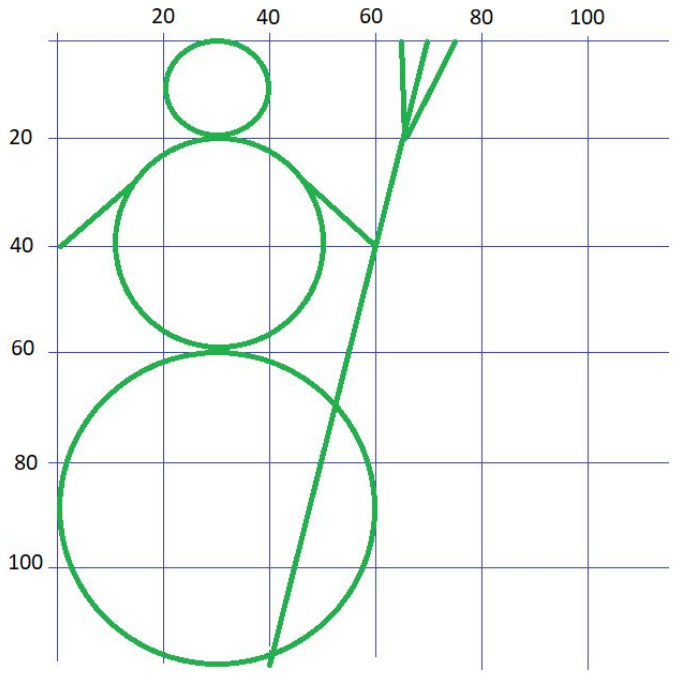
```
int y = 0;
do {
    StClausAuto(hdc, 0, y);
    y += 90;
} while (y < 300);
```

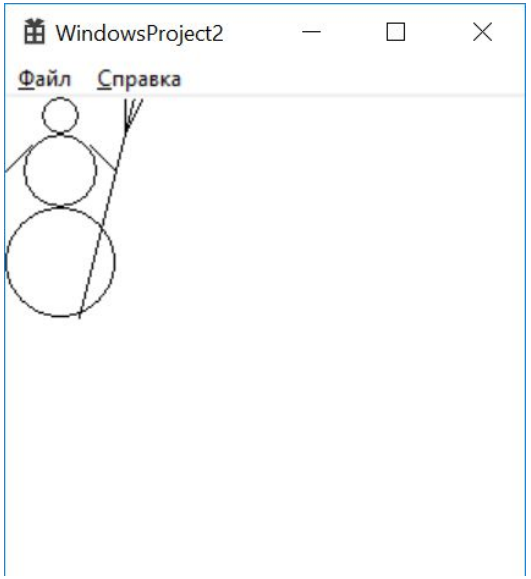


Задача 1.4: 4 логотипа в диагональную линию



Задача 2.1 – Создать картинку по образцу

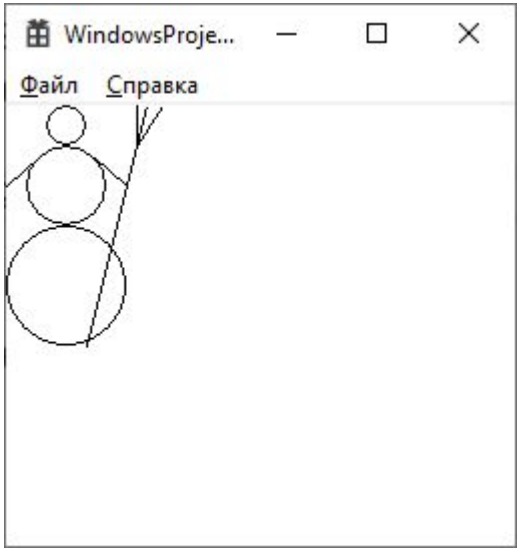




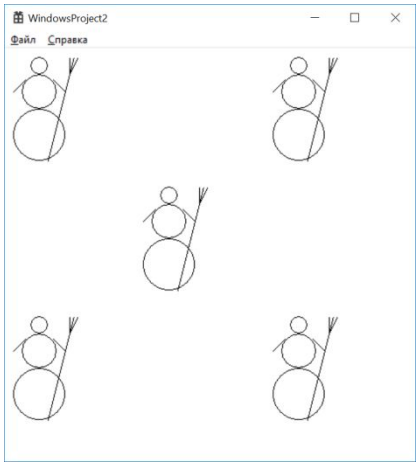
Задача 2.2 – Создать функцию SnowWoman

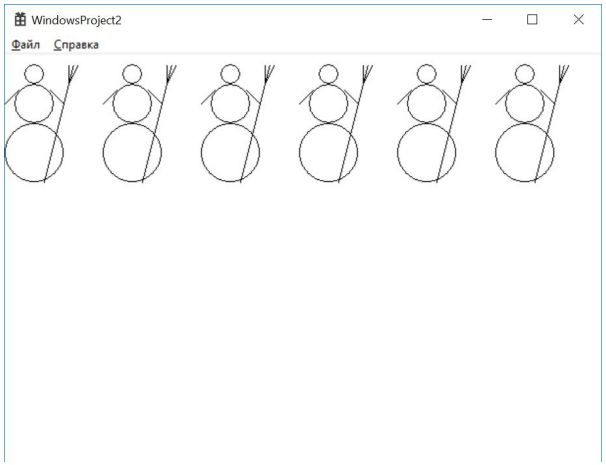
```
209 void SnowWoman(HDC hdc, int x, int y) {
210     Ellipse(hdc, 20 + x, 0 + y, 40 + x, 20 + y);
211     Ellipse(hdc, 10 + x, 20 + y, 50 + x, 60 + y);
```

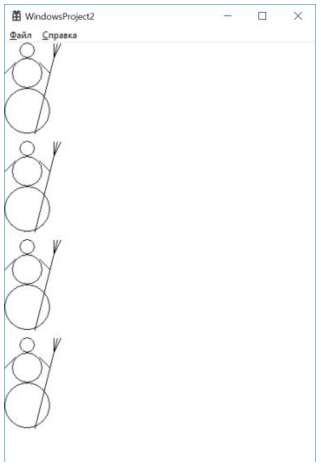
```
261     case WM_PAINT:
262     {
263         PAINTSTRUCT ps;
264         HDC hdc = BeginPaint(hWnd, &ps);
265         // TODO: Добавьте сюда любой код прорисовки, использующий HDC...
266
267         SnowWoman(hdc, 0, 0);
268
269         EndPaint(hWnd, &ps);
270     }
271     break;
```

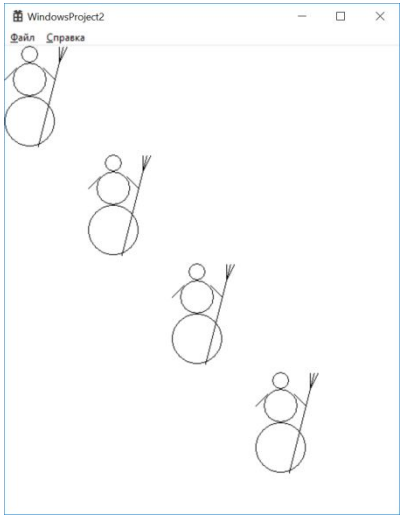


Задача 2.3 – 2.6 – Создать 4 рисунков из снежных баб



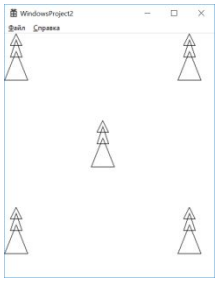


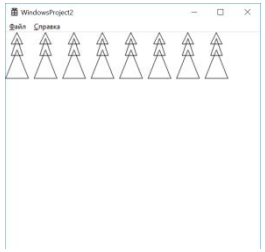


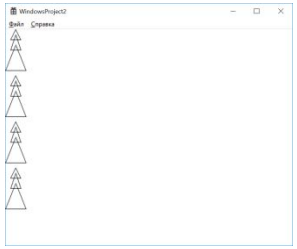


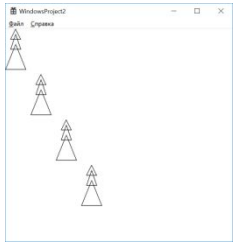
Домашнее задание

1) Доделать рисунки из Задач 1 и 2, которые не успели сделать на занятии в классе.
2) Нарисовать для всех циклических алгоритмов блоксхему.
3) Создать в виде отдельной функции логотип любого автомобиля. Созданная фукнция должна иметь вид Logo(HDC hdc, int x, int y) . При этом сделать логотип такого размера, чтобы он вмещался по высоте окна не меньше 4 раз, по ширине не меньше 6 раз
4) Используя эту функцию создать узоры из логотипов по следующим 4 схемам:









5) Для последнего узора нарисовать блоксхему алгоритма.

Домашнее задание – оформление

Для сдачи домашней работы нужно иметь:

1) код программы

2) Расчет картинки – на бумаге (на отдельном листе или в тетради) или в файле с видимыми признаками расчетов

3) Скрин шот выполнения программы

4) Блоксхемы для циклических алгоритмов

Срок выполнения – до следующей встречи на лабораторной работе

Если болел/не мог – это становится «долгом»