

# Основы программирования

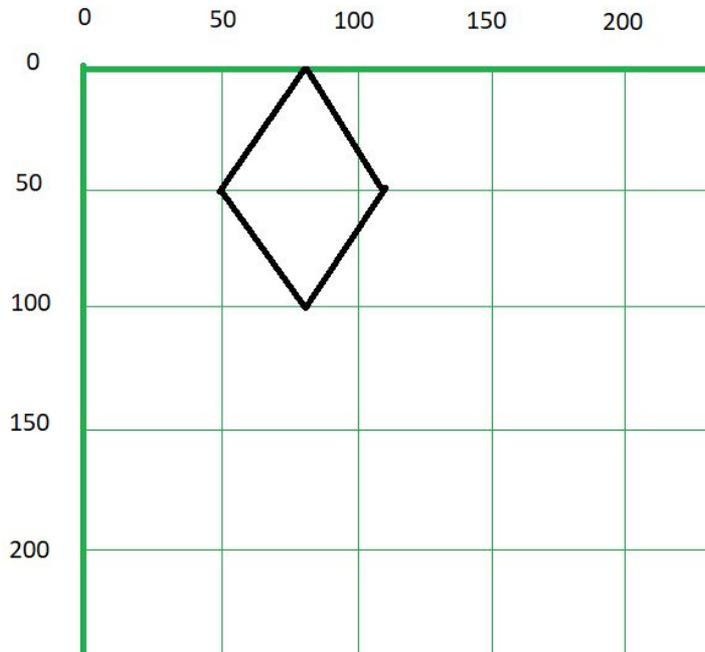
## Лабораторная работа №9

WinAPI Графика – Относительные координаты.

Функции с параметрами.

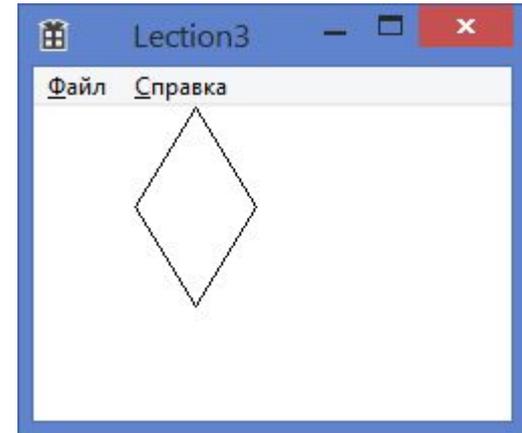
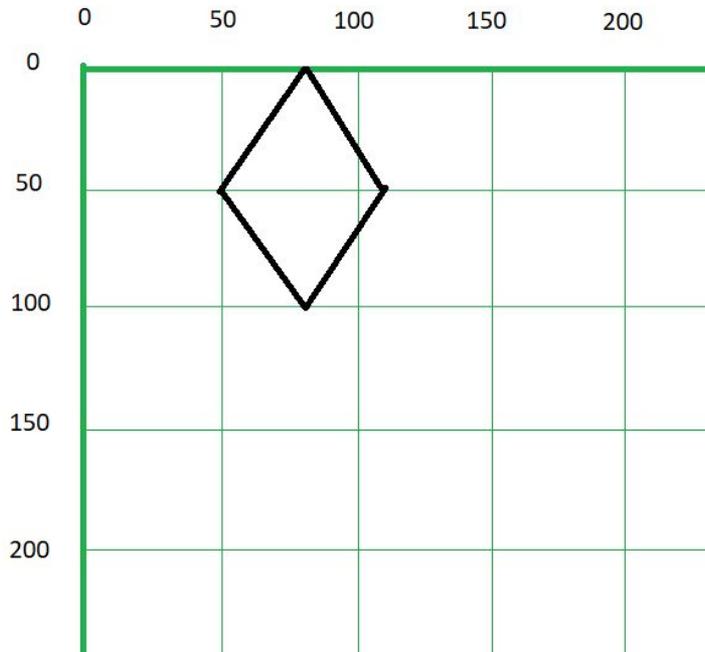
Власенко Олег Федосович

# Нарисуем ромб



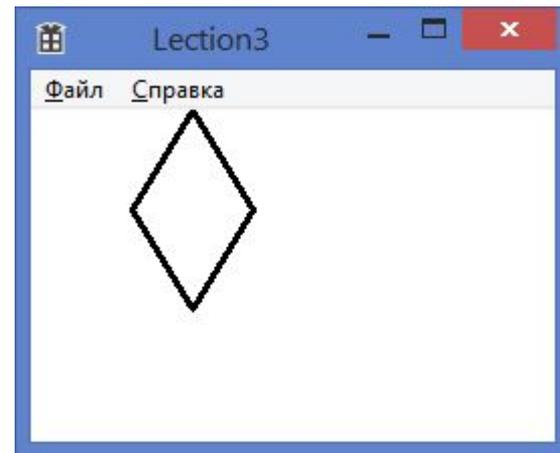
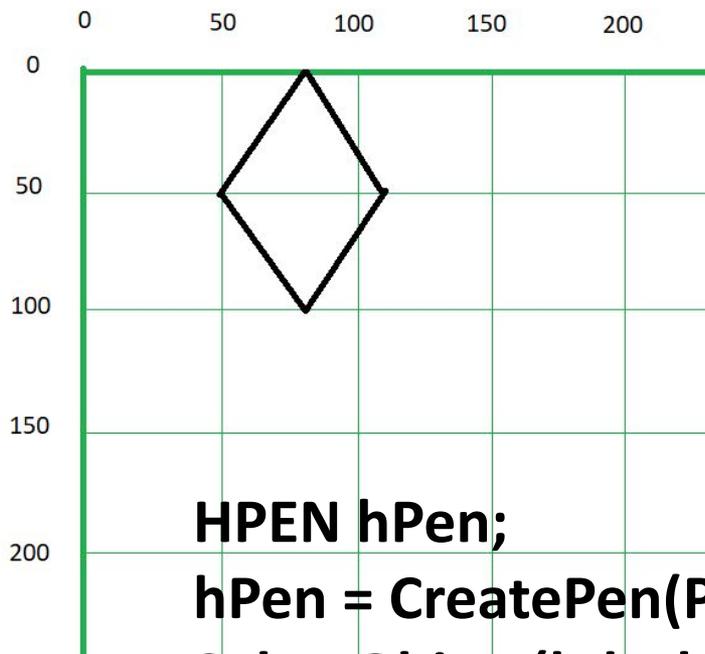
```
// Рисуем ромб  
MoveToEx(hdc, 80, 0, NULL);  
LineTo(hdc, 50, 50);  
LineTo(hdc, 80, 100);  
LineTo(hdc, 110, 50);  
LineTo(hdc, 80, 0);
```

# 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);
```

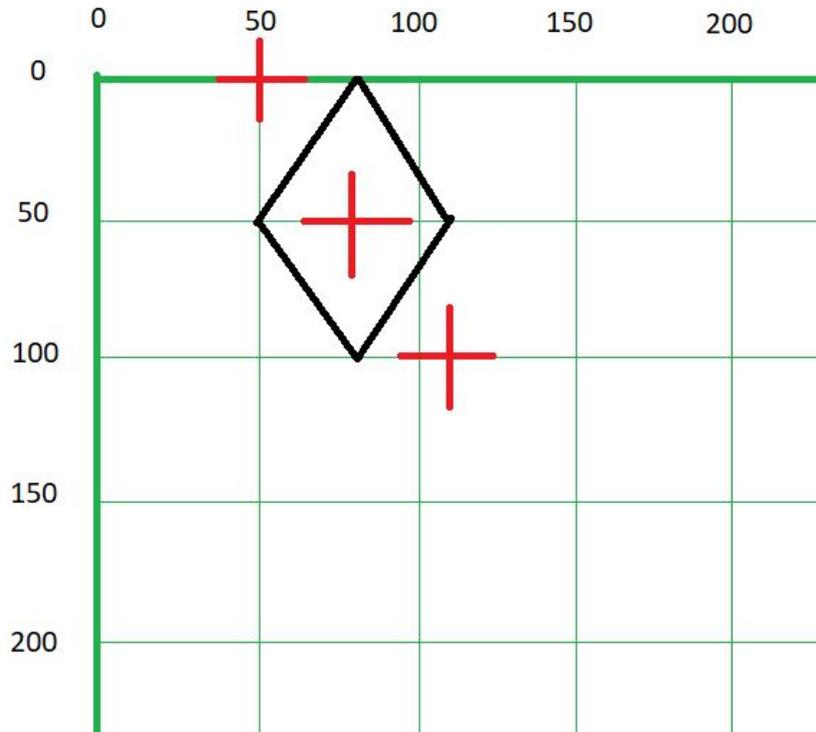
```
LineTo(hdc, 50, 50);
```

```
LineTo(hdc, 80, 100);
```

```
LineTo(hdc, 110, 50);
```

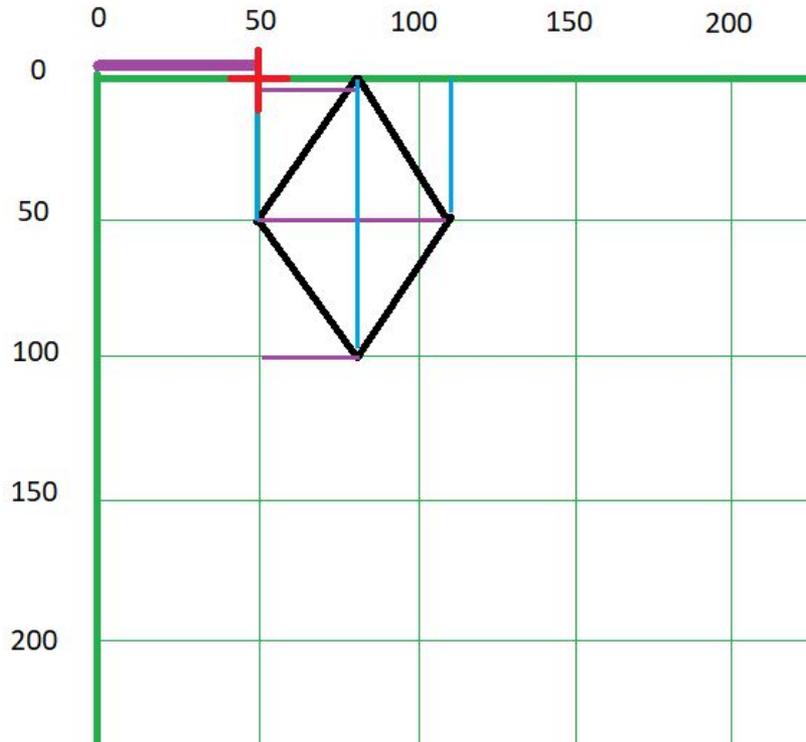
```
LineTo(hdc, 80, 0);
```

# 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);
```

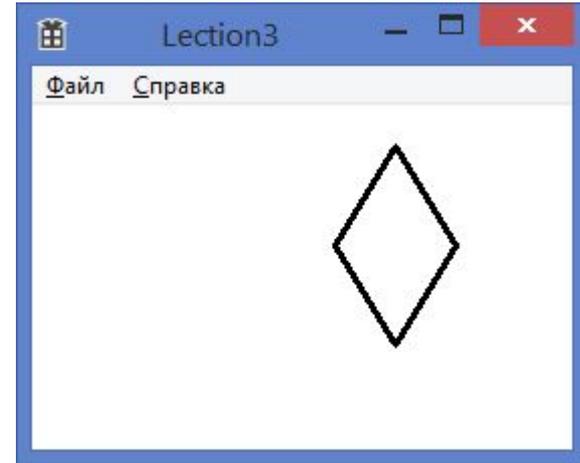
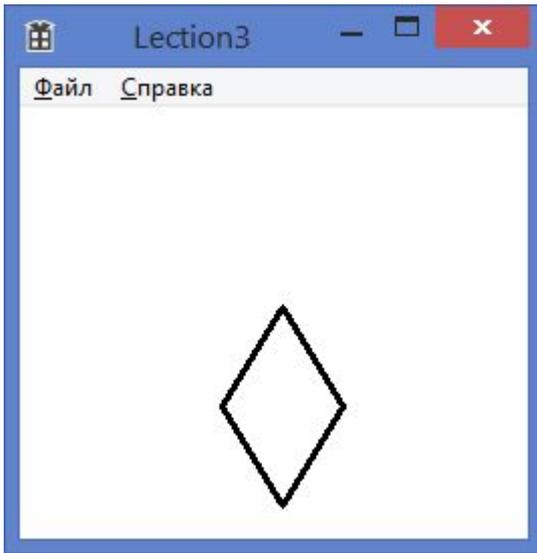
```
LineTo(hdc, x, y + 50);
```

```
LineTo(hdc, x + 30, y + 100);
```

```
LineTo(hdc, x + 60, y + 50);
```

```
LineTo(hdc, x + 30, y);
```

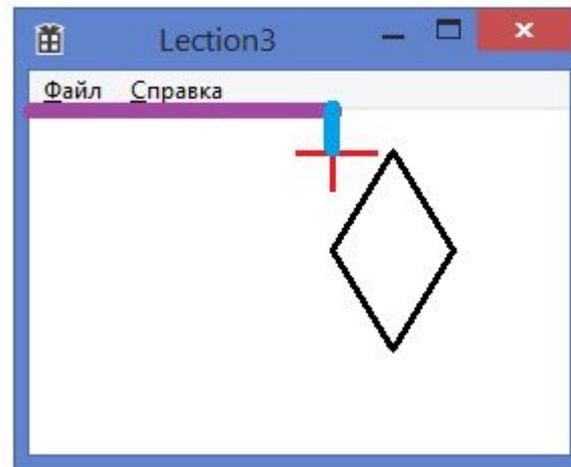
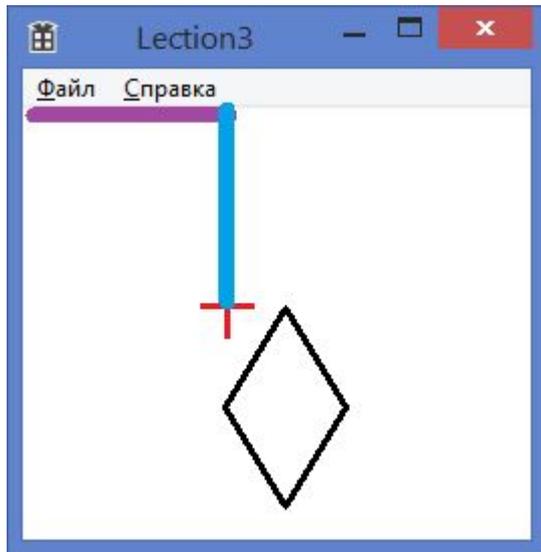
# Относительные координаты



```
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);
```

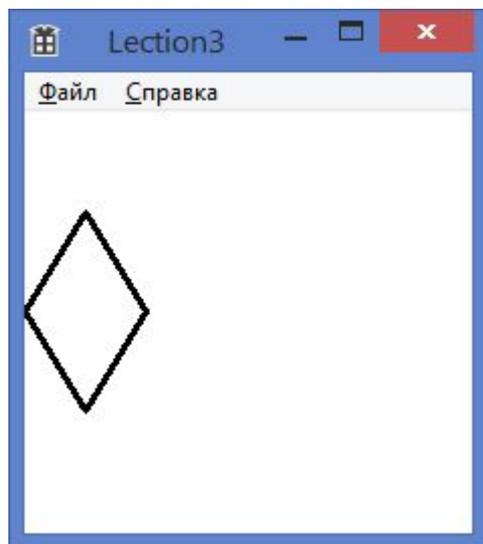
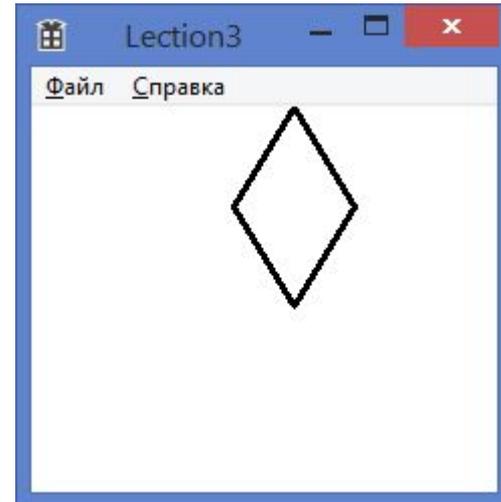
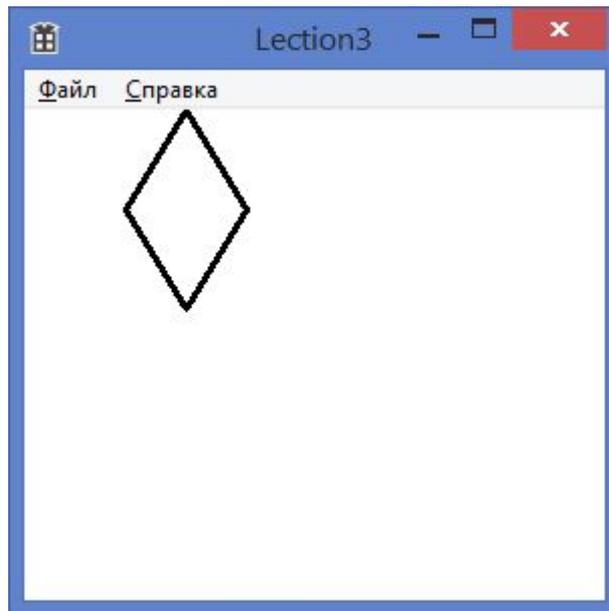
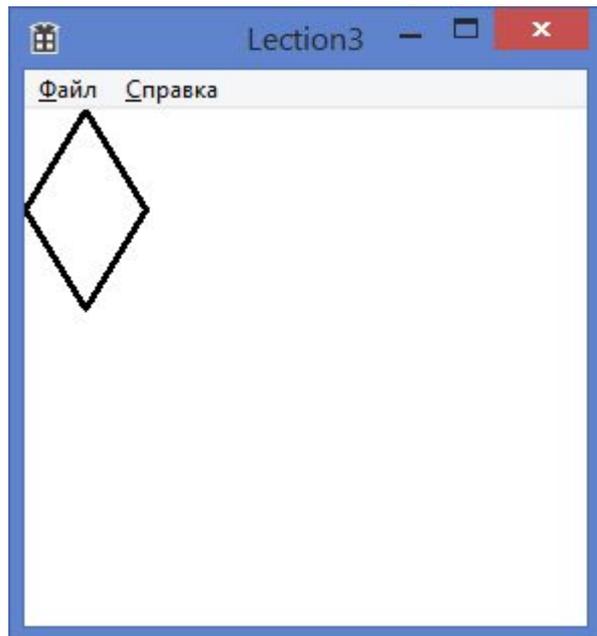
# Относительные координаты



```
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);
```

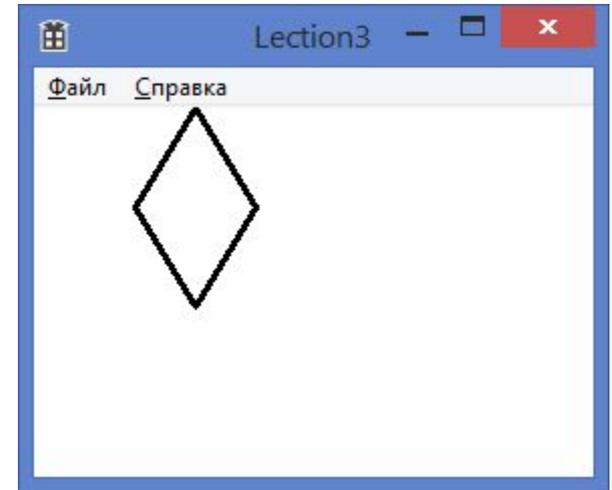
```
int x = 150;  
int y = 20;  
MoveToEx(hdc, x + 30, y, NULL);  
LineTo(hdc, x, y + 50);  
LineTo(hdc, x + 30, y + 100);  
LineTo(hdc, x + 60, y + 50);  
LineTo(hdc, x + 30, y);
```

# 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);
```

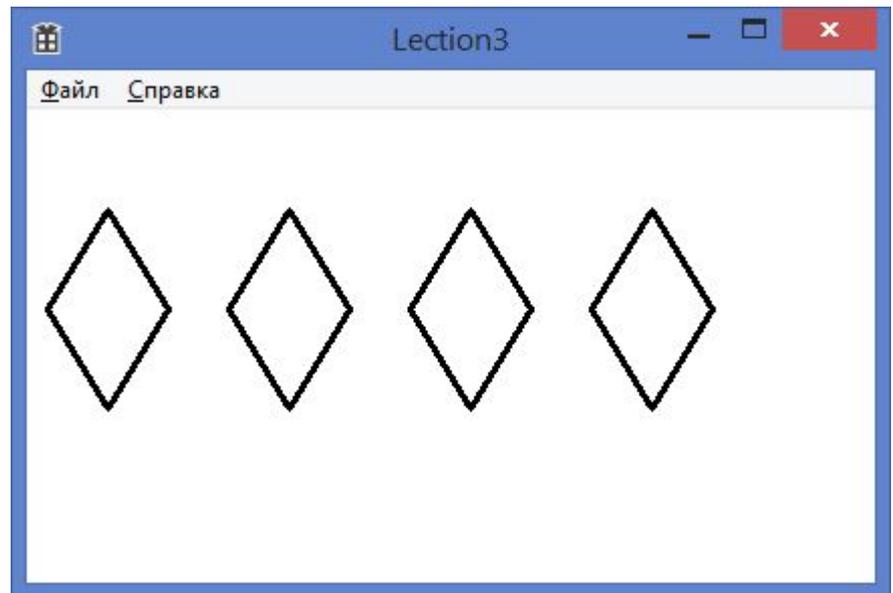
```
Romb(hdc, 10, 50);
```

```
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;
```

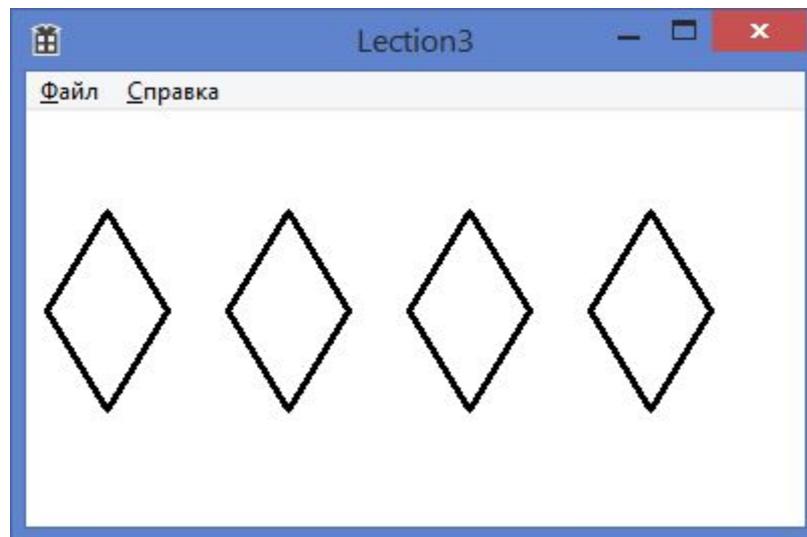
```
do {
```

```
    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;
```

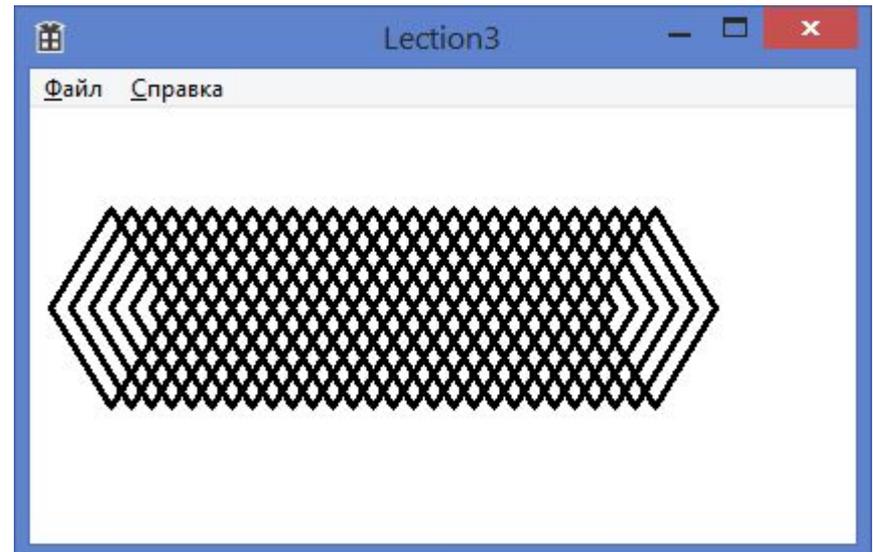
```
do {
```

```
    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;
```

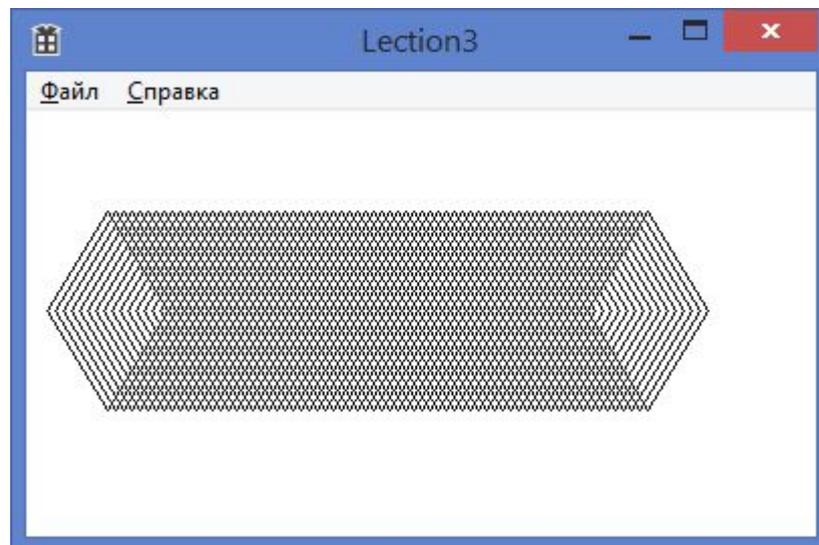
```
do {
```

```
    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;
```

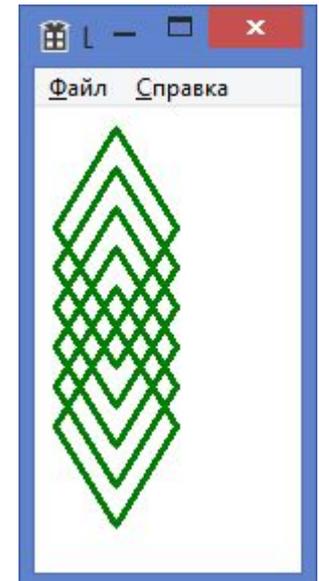
```
int y = 10;
```

```
do {
```

```
    Romb(hdc, x, y);
```

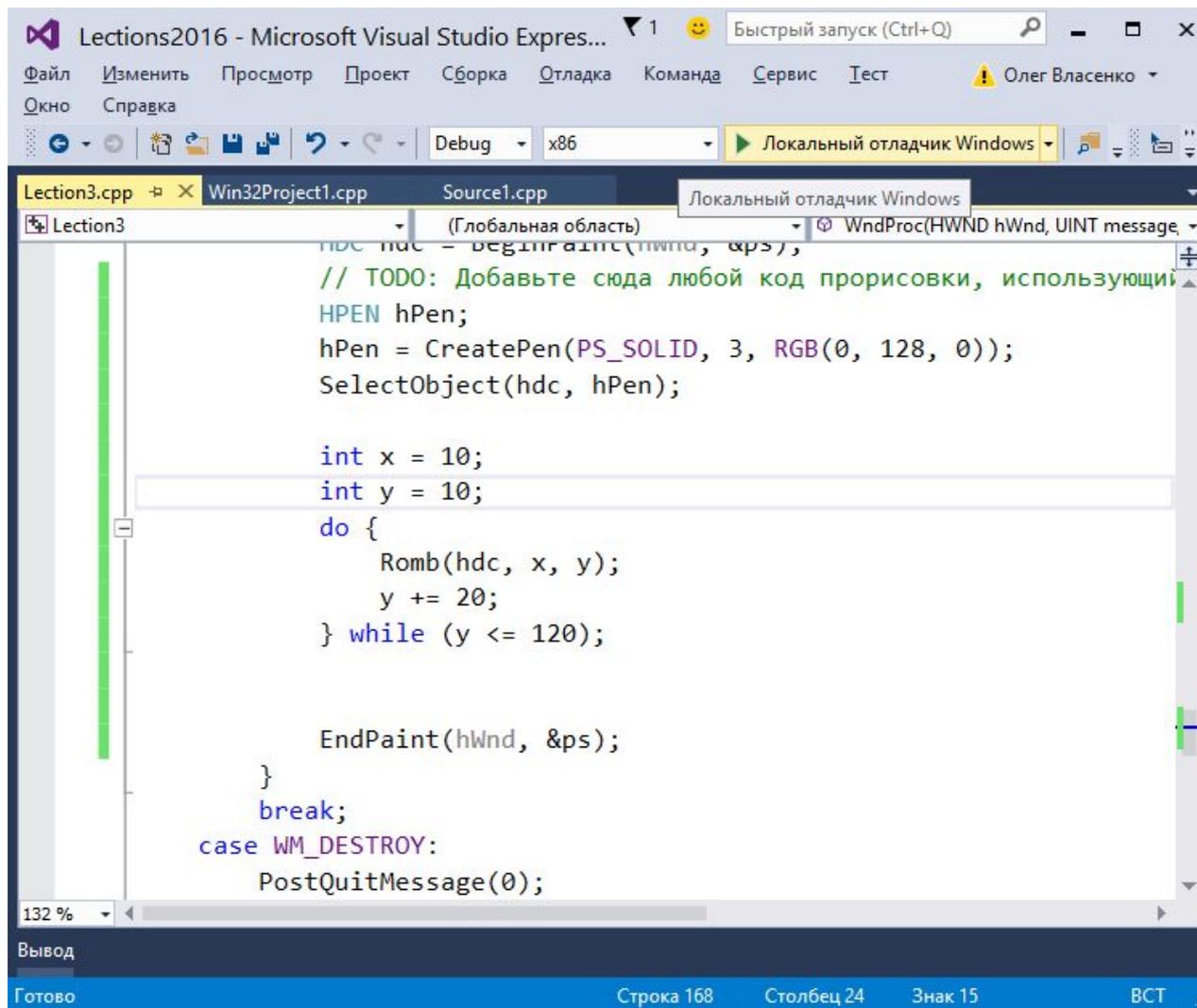
```
    y += 20;
```

```
} while (y <= 120);
```



# Трассировка циклического алгоритма (2)

## Установка точки останова

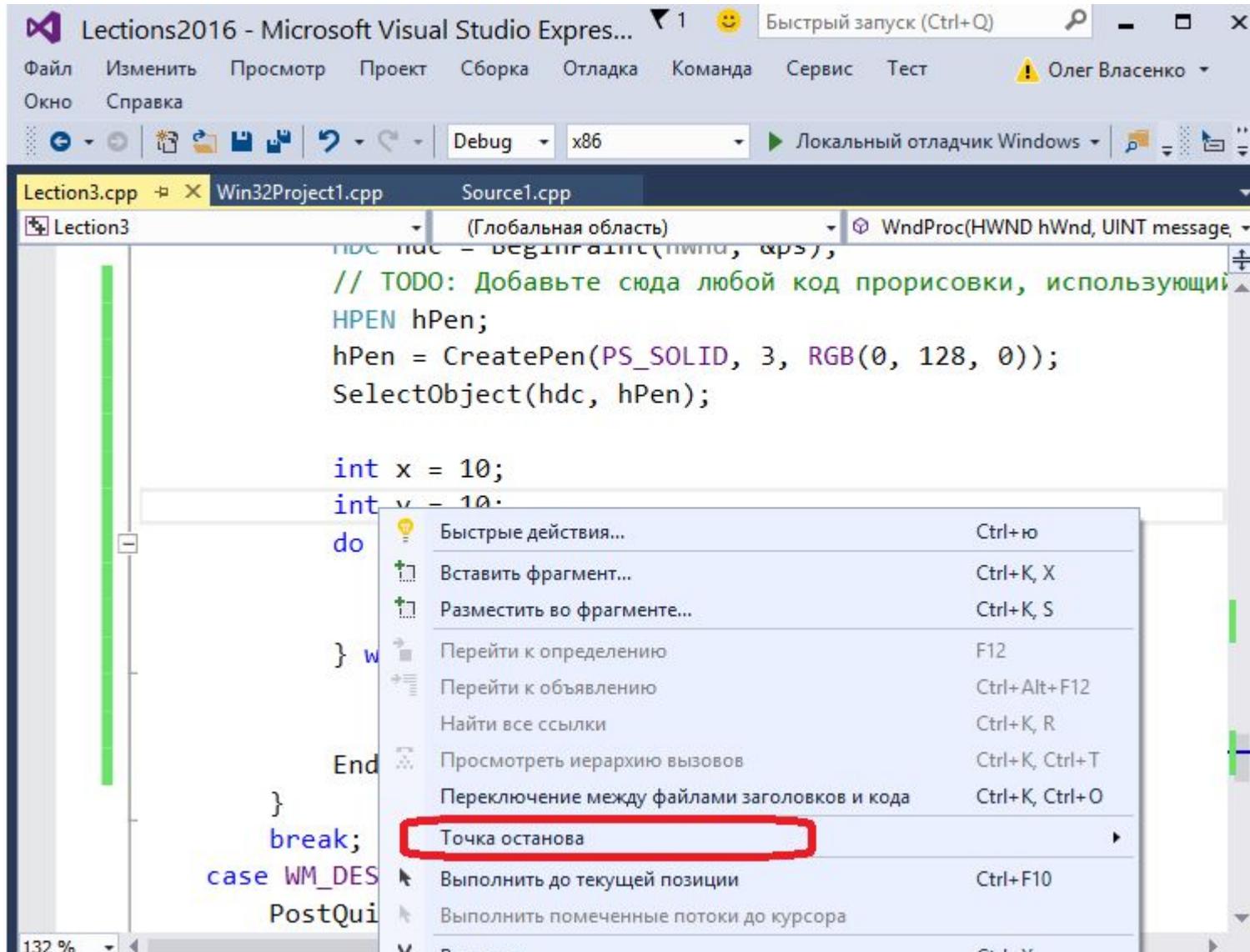


The screenshot shows the Microsoft Visual Studio Express IDE. The main window displays a C++ source file named 'Lecture3.cpp'. The code is as follows:

```
hdc hdc = BeginPaint(hwnd, &ps);  
// 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);
```

A green vertical bar on the left side of the code editor indicates a breakpoint is set on the first line of the `while` loop. The status bar at the bottom shows 'Готово' (Ready), 'Строка 168' (Line 168), 'Столбец 24' (Column 24), 'Знак 15' (Character 15), and 'ВСТ'.

# Трассировка циклического алгоритма (3)



# Трассировка циклического алгоритма (4)

The screenshot shows a code editor window with the following code:

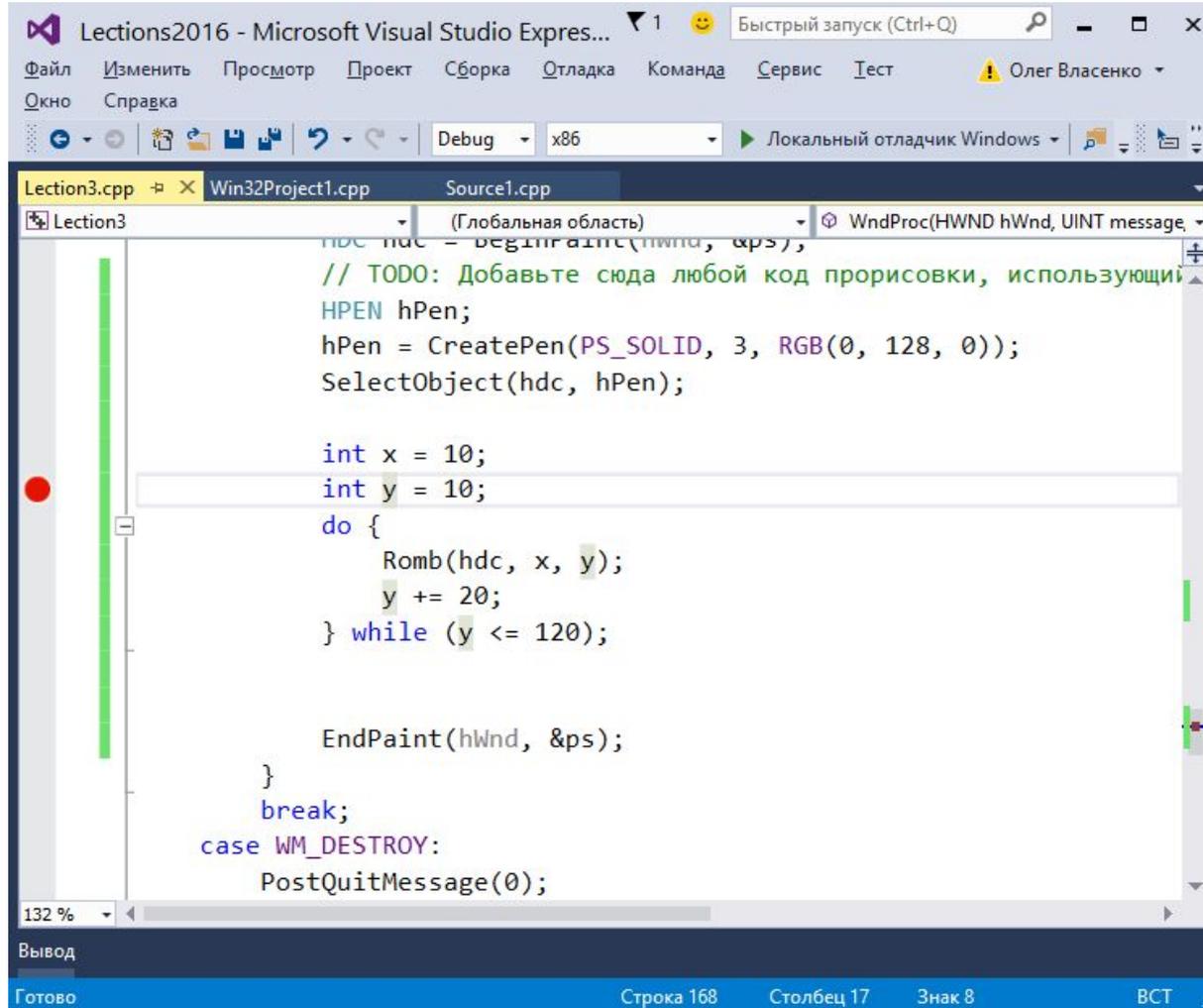
```
hdc = BeginPaint(hwnd, &ps);  
// TODO: Добавьте сюда любой код прорисовки, использующий...  
HPEN hPen;  
hPen = CreatePen(PS_SOLID, 3, RGB(0, 128, 0));  
SelectObject(hdc, hPen);  
  
int x = 10;  
int y = 10;  
do  
{  
    // ...  
} while (true);  
break;  
case WM_DESTROY:  
    PostQuitMessage(0);  
return 0;  
}
```

A context menu is open over the code, with the following options:

- 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** (Breakpoint) - selected, with a sub-menu open:
  - Вставить точку останова** (Add breakpoint) - highlighted in red
  - Вставить точку отслеживания (Add watchpoint)
- Выполнить до текущей позиции (Ctrl+F10)
- Выполнить помеченные потоки до курсора
- Вырезать (Ctrl+X)
- Копировать (Ctrl+C)
- Вставить (Ctrl+V)
- Структура

The status bar at the bottom shows "19 Трассировка циклического алгоритма (3)".

# Трассировка циклического алгоритма (5)



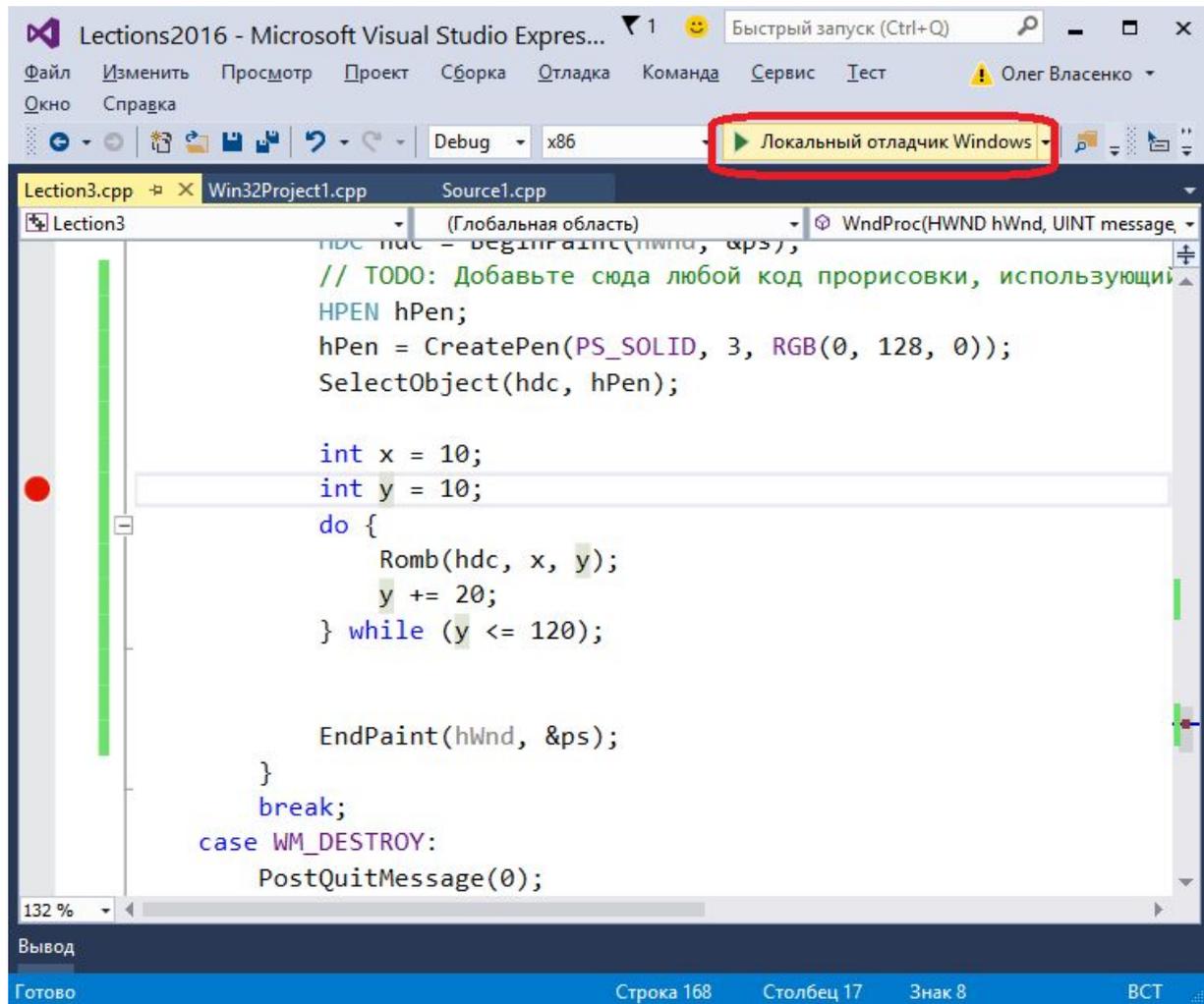
```
Lections2016 - Microsoft Visual Studio Express... 1 ☺ Быстрый запуск (Ctrl+Q)
Файл Изменить Просмотр Проект Сборка Отладка Команда Сервис Тест Олег Власенко
Окно Справка
Debug x86 Локальный отладчик Windows
Lecture3.cpp Win32Project1.cpp Source1.cpp
Lecture3 (Глобальная область) WndProc(HWND hWnd, UINT message,
pdc pdc = beginpaint(hwnd, &ps);
// 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



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

The screenshot shows the Microsoft Visual Studio IDE in debug mode. The main window displays the source code of `Lesson3.cpp` with a breakpoint set at the start of a `do` loop. The code includes a `while` loop for window creation and a `do` loop for drawing a rectangle.

```
WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam) {
    // TODO: Добавьте сюда любой код прорисовки, используя
    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);
    } while (true);
}
```

The **Локальные** (Locals) window shows the following variables:

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

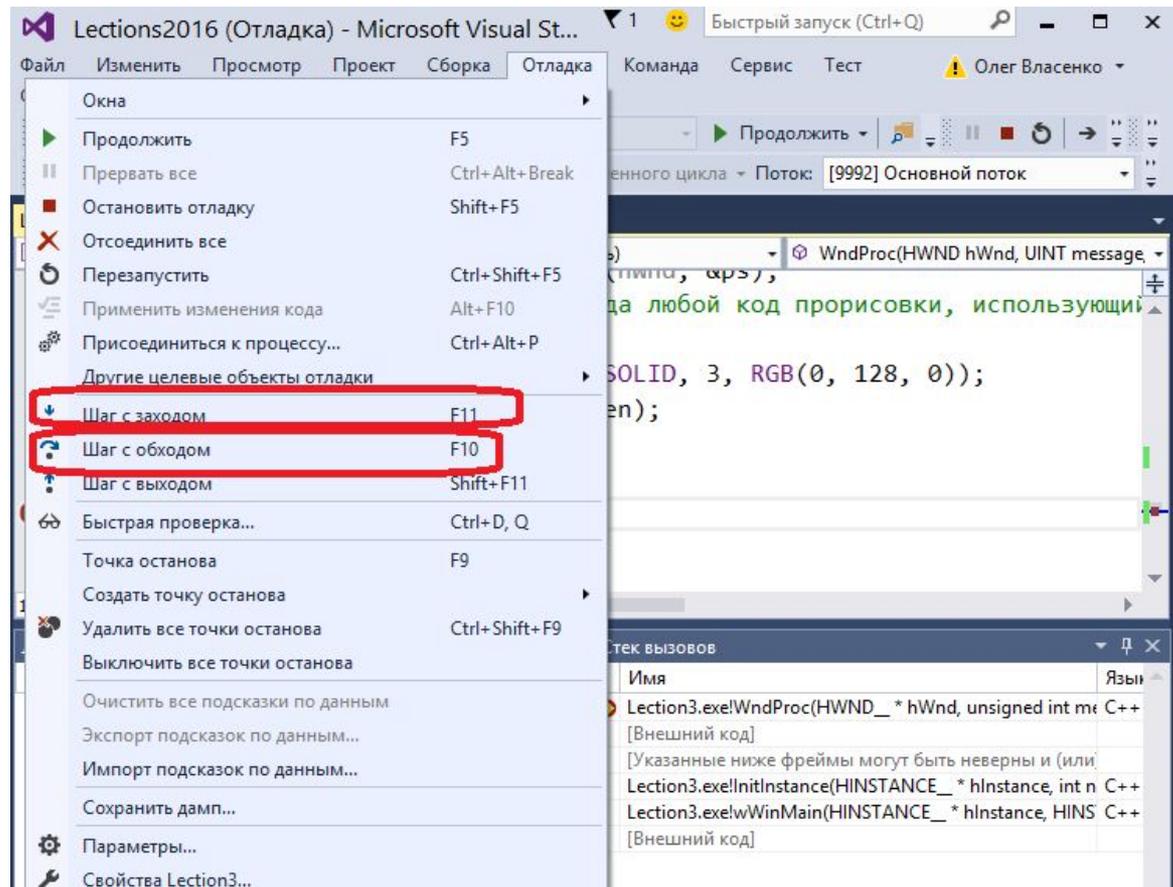
The **Стек вызовов** (Call Stack) window shows the following stack frames:

Имя	Язык
Lesson3.exe!WndProc(HWND_* hWnd, unsigned int message, WPARAM wParam, LPARAM lParam)	C++
[Внешний код]	
Lesson3.exe!InitInstance(HINSTANCE_* hInstance, int nCmdShow)	C++
Lesson3.exe!wWinMain(HINSTANCE_* hInstance, HINSTANCE_* hPrevInstance, LPWSTR lpCmdLine, int nCmdShow)	C++
[Внешний код]	

At the bottom of the window, the status bar displays: `Загрузка символов для oleacc.dll`, `Строка 168`, `Столбец 1`, `Знак 1`, and `ВСТ`.

# Трассировка циклического алгоритма (8)

## Пошаговая трассировка



# Трассировка циклического алгоритма (9)

## Шаг сделан (F10)

The screenshot shows the Microsoft Visual Studio IDE in debug mode. The main window displays the source code for `Lection3.cpp`. A red dot indicates a breakpoint is set at the beginning of a `while` loop. The code is as follows:

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

EndPoint(hWnd, &ps);
```

The **Локальные** (Locals) window shows the following variables:

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

The **Стек вызовов** (Call Stack) window shows the following stack frames:

- Lection3.exe!WndProc(HWND\_ \* hWnd, unsigned int message) C++ [Внешний код]
- [Указанные ниже фреймы могут быть неверны и (или) неактуальны]
- Lection3.exe!InitInstance(HINSTANCE\_ \* hInstance, int nCmdShow) C++
- Lection3.exe!WinMain(HINSTANCE\_ \* hInstance, HINSTANCE\_ \* hPrevInstance, int nCmdShow) C++
- [Внешний код]

The status bar at the bottom indicates: `Загрузка символов для oleacc.dll`, `Строка 170`, `Столбец 1`, `Знак 1`, `ВСТ`.

# Трассировка циклического алгоритма (10)

## Еще 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

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

132 %

Локальные		
Имя	Значение	Тип
x	10	int
y	10	int
hPen	0xa930090a {unused=???	HPEN_
ps	{hdc=0x4b0127ff {unused=???	tagPAINT
hdc	0x4b0127ff {unused=???	HDC_*
hWnd	0x00150a46 {unused=???	HWND_*
message	15	unsigned
wParam	0	unsigned

Стек вызовов	
Имя	Язык
Lection3.exe!WndProc(HWND_*, unsigned int m...	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]	

Локальные Контрольные значения 1

Загрузка символов для oleacc.dll Строка 171 Столбец 1 Знак 1 ВСТ

# Трассировка циклического алгоритма (11)

## Еще 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] Lesson3.exe События жизненного цикла Поток: [9992] Основной поток

```
Lesson3.cpp Win32Project1.cpp Source1.cpp
Lesson3 (Глобальная область) WndProc(HWND hWnd, UINT message)
int x = 10;
int y = 10;
do {
    Romb(hdc, x, y);
    y += 20;
} while (y <= 120); ≤ 1 мс прошло
```

132 %

Локальные		
Имя	Значение	Тип
x	10	int
y	30	int
hPen	0xa930090a {unused=???	HPEN_*
ps	{hdc=0x4b0127ff {unused=???	tagPAINT
hdc	0x4b0127ff {unused=???	HDC_*
hWnd	0x00150a46 {unused=???	HWND_*
message	15	unsigned int
wParam	0	unsigned int

Стек вызовов	
Имя	Язык
Lesson3.exe!WndProc(HWND_* hWnd, unsigned int message) [Внешний код]	C++
[Указанные ниже фреймы могут быть неверны и (или) неактуальны]	
Lesson3.exe!InitInstance(HINSTANCE_* hInstance, int nCmdShow) [Внешний код]	C++
Lesson3.exe!WinMain(HINSTANCE_* hInstance, HINSTANCE_* hPrevInstance, LPSTR lpCmdLine, int nCmdShow) [Внешний код]	C++

Локальные Контрольные значения 1

Загрузка символов для oleacc.dll Строка 172 Столбец 1 Знак 1 ВСТ

# Трассировка циклического алгоритма (12)

## 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

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=???	tagPAINT
hdc	0x4b0127ff {unused=???	HDC_*
hWnd	0x00150a46 {unused=???	HWND_*
message	15	unsigned
wParam	0	unsigned

Стек вызовов	
Имя	Язык
Lection3.exe!WndProc(HWND_* hWnd, unsigned int message) [Внешний код]	C++
[Указанные ниже фреймы могут быть неверны и (или) неактуальны]	
Lection3.exe!InitInstance(HINSTANCE_* hInstance, int nCmdShow) [Внешний код]	C++
Lection3.exe!wWinMain(HINSTANCE_* hInstance, HINSTANCE_* hPrevInstance, LPWSTR lpCmdLine, int nCmdShow) [Внешний код]	C++

Готово Строка 170 Столбец 1 Знак 1 ВСТ

# Трассировка циклического алгоритма (13)

## Еще шаг сделан

The screenshot shows the Microsoft Visual Studio IDE in debug mode. The main window displays a C++ source file with the following code:

```
int x = 10;
int y = 10;
do {
    Romb(hdc, x, y);
    y += 20; ≤ 2 мс прошло
} while (y <= 120);
```

The code is being executed, and a red dot indicates the current execution point. The 'Локальные' (Locals) window shows the following variables:

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

The 'Стек вызовов' (Call Stack) window shows the following stack frames:

Имя	Язык
Lection3.exe!WndProc(HWND_ * hWnd, unsigned int message, WPARAM wParam)	C++
[Внешний код]	
[Указанные ниже фреймы могут быть неверны и (или) отсутствовать]	
Lection3.exe!InitInstance(HINSTANCE_ * hInstance, int nCmdShow)	C++
Lection3.exe!wWinMain(HINSTANCE_ * hInstance, HINSTANCE_ * hPrevInstance, LPWSTR lpCmdLine, int nCmdShow)	C++
[Внешний код]	

The status bar at the bottom indicates 'Готово' (Ready), 'Строка 171' (Line 171), 'Столбец 1' (Column 1), 'Знак 1' (Character 1), and 'ВСТ' (Breakpoint).

# Трассировка циклического алгоритма (14)

## 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

Процесс: [7264] Lesson3.exe Поток: [9992] Основной поток

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

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

Имя	Язык
Lesson3.exe!WndProc(HWND_ * hWnd, unsigned int message, WPARAM wParam)	C++
[Внешний код]	
[Указанные ниже фреймы могут быть неверны и (или) неактуальны]	
Lesson3.exe!InitInstance(HINSTANCE_ * hInstance, int nCmdShow)	C++
Lesson3.exe!wWinMain(HINSTANCE_ * hInstance, HINSTANCE_ * hPrevInstance, LPWSTR lpCmdLine, int nCmdShow)	C++
[Внешний код]	

Готово Строка 172 Столбец 1 Знак 1 ВСТ

# Трассировка циклического алгоритма (15)

## Еще шаг сделан

The screenshot shows the Microsoft Visual Studio IDE with the following details:

- Title Bar:** Lessons2016 (Отладка) - Microsoft Visual St... Быстрый запуск (Ctrl+Q)
- Menu Bar:** Файл, Изменить, Просмотр, Проект, Сборка, Отладка, Команда, Сервис, Тест, Олер Власенко
- Toolbar:** Includes standard IDE icons and a 'Debug' dropdown menu.
- Process/Thread:** Процесс: [7264] Lesson3.exe; Поток: [9992] Основной поток
- Code Editor:** Shows the source code for Lesson3.cpp. The code is:

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

A red circle is drawn on the left side of the code editor, and a yellow arrow points to the first line of the do-while loop.
- Locals Window:** Lists local variables and their values:

Имя	Значение	Тип
x	10	int
y	50	int
hPen	0xa930090a {unused=???	HPEN_
ps	{hdc=0x4b0127ff {unused=???	tagPAIN
hdc	0x4b0127ff {unused=???	HDC_*
hWnd	0x00150a46 {unused=???	HWND_
message	15	unsigned
wParam	0	unsigned
- Call Stack Window:** Shows the current function call:

Имя	Язык
Lesson3.exe!WndProc(HWND_ * hWnd, unsigned int m...	C++
[Внешний код]	
[Указанные ниже фреймы могут быть неверны и (или)	
Lesson3.exe!InitInstance(HINSTANCE_ * hInstance, int n...	C++
Lesson3.exe!wWinMain(HINSTANCE_ * hInstance, HINS...	C++
[Внешний код]	
- Status Bar:** Готово, Строка 170, Столбец 1, Знак 1, ВСТ

# Трассировка циклического алгоритма (16)

## Возобновить выполнение

The screenshot shows the Microsoft Visual Studio debugger interface. The main window displays the source code of a C++ program. The code is as follows:

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

The 'Continue' button (represented by a green play icon) is highlighted with a red circle. Below the code editor, there are two windows:

- Локальные (Locals):** A table showing the values of local variables.
- Стек вызовов (Call Stack):** A list of the current call stack frames.

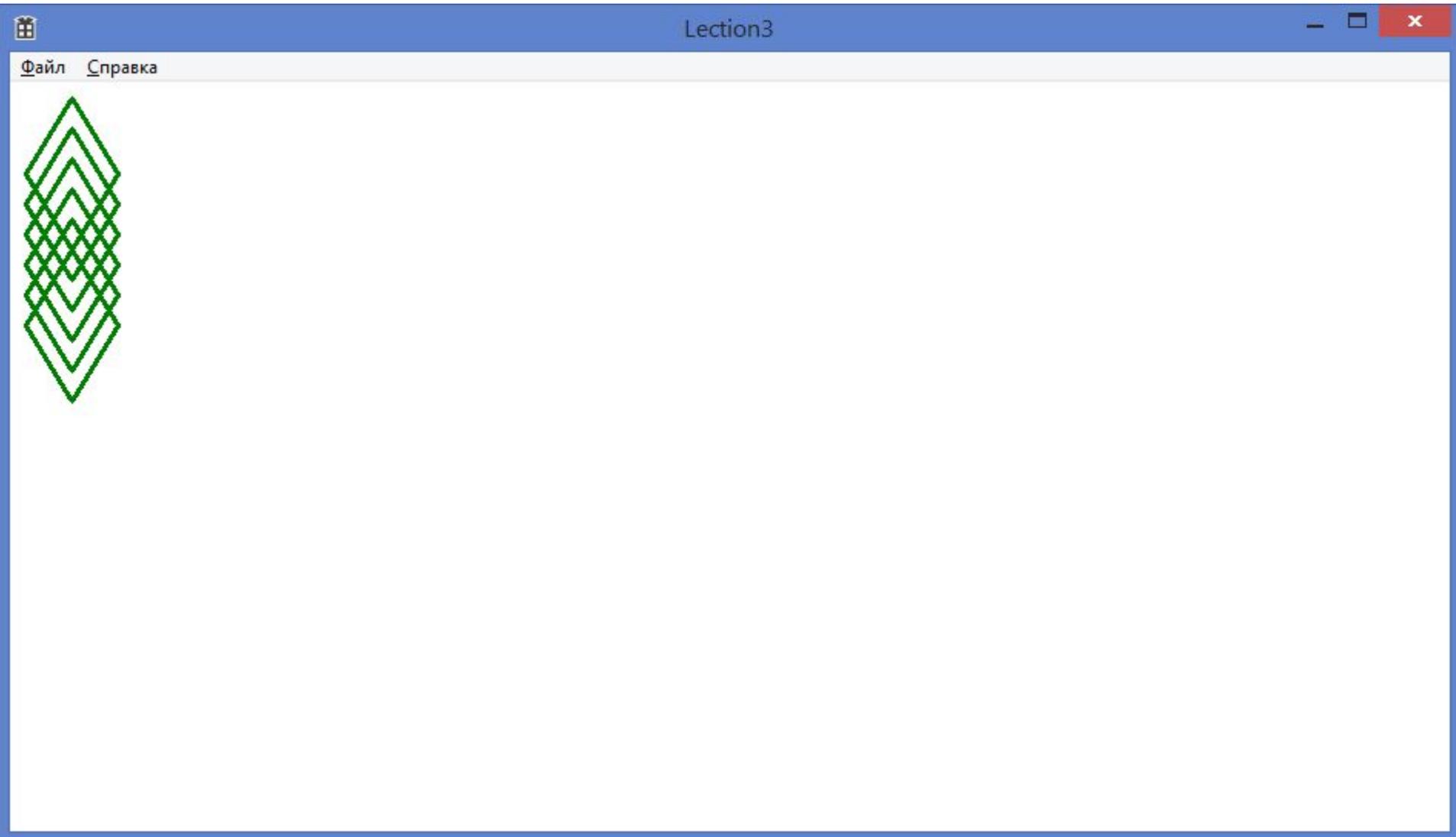
Имя	Значение	Тип
x	10	int
y	50	int
hPen	0xa930090a {unused=???}	HPEN_*
ps	{hdc=0x4b0127ff {unused=???} fEra	tagPAIN
hdc	0x4b0127ff {unused=???}	HDC_*
hWnd	0x00150a46 {unused=???}	HWND_*
message	15	unsigned
wParam	0	unsigned

Имя	Язык
Lection3.exe!WndProc(HWND_* hWnd, unsigned int m	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]	

At the bottom of the window, the status bar shows: Готово, Строка 170, Столбец 1, Знак 1, ВСТ.

# Трассировка циклического алгоритма (17)

## Приложение работает!



# Изменяем сразу и X и Y

...

```
int x = 10;
```

```
int y = 50;
```

```
do {
```

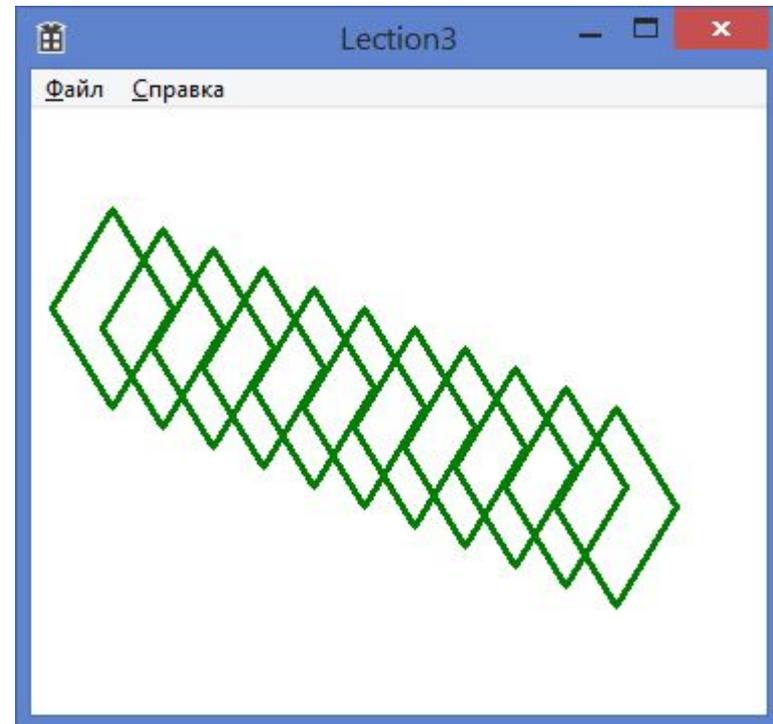
```
    Romb(hdc, x, y);
```

```
    x += 25;
```

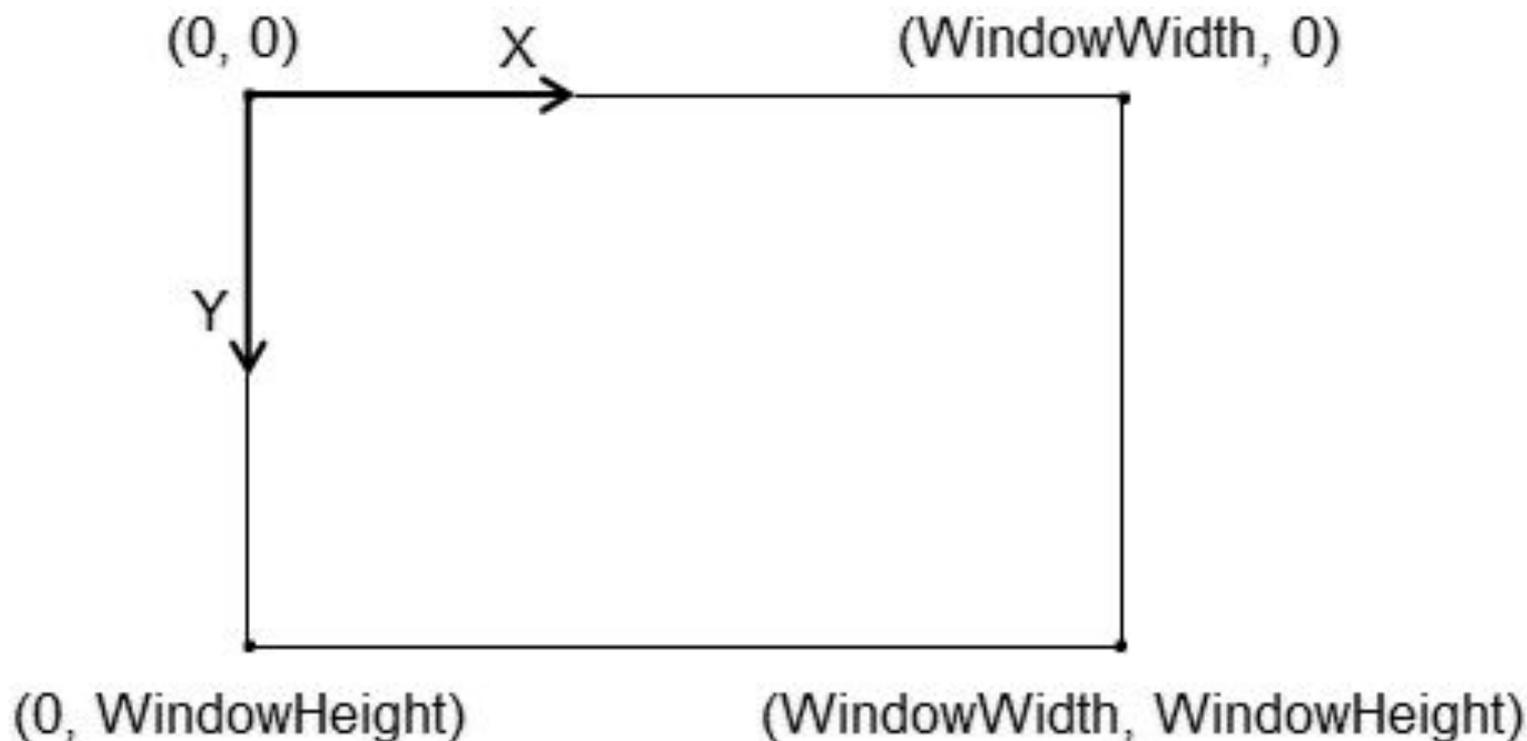
```
    y += 10;
```

```
} while (x <= 280);
```

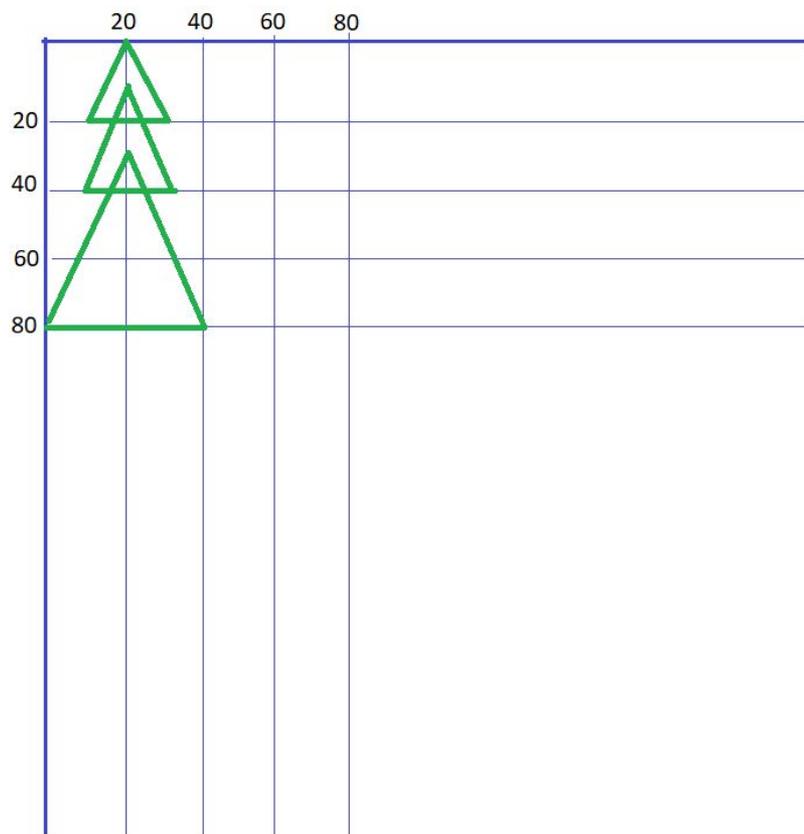
...



# Автомобиль деда Мороза - логотип



# Автомобиль деда Мороза - логотип



# Автомобиль деда Мороза - логотип

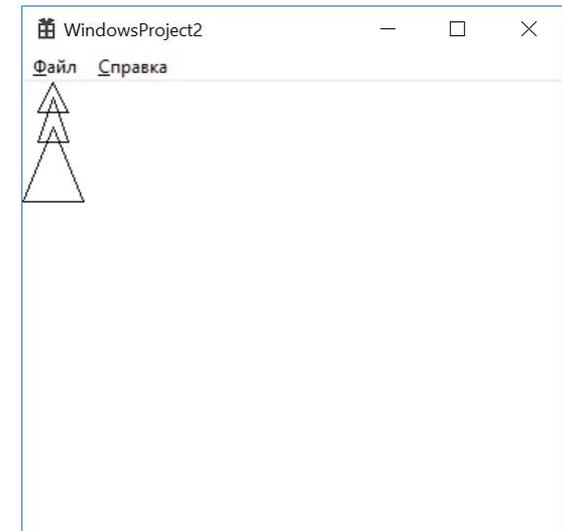
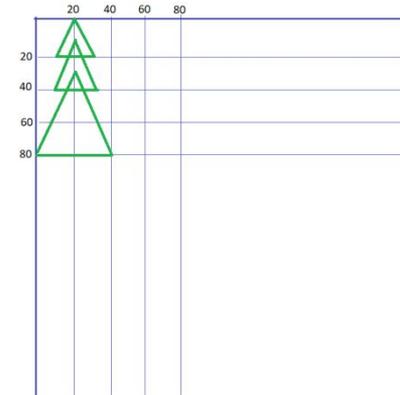
```
case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: Добавьте сюда любой код прорисовки, использующий HDC...

    // 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);

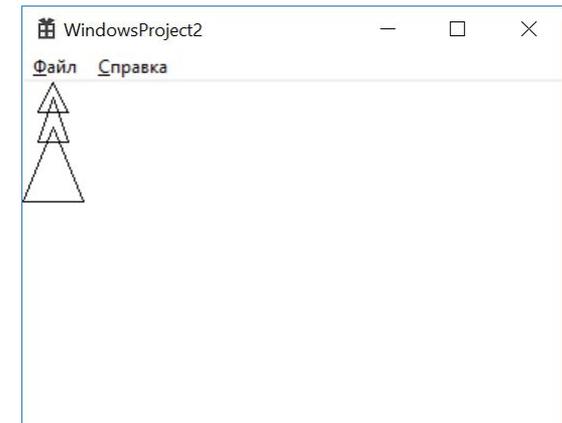
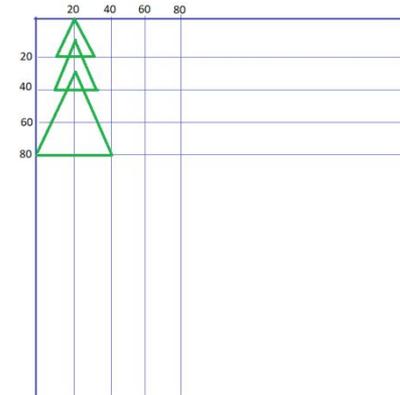
    // нижний треугольник
    MoveToEx(hdc, 20, 30, NULL);
    LineTo(hdc, 40, 80);
    LineTo(hdc, 0, 80);
    LineTo(hdc, 20, 30);

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



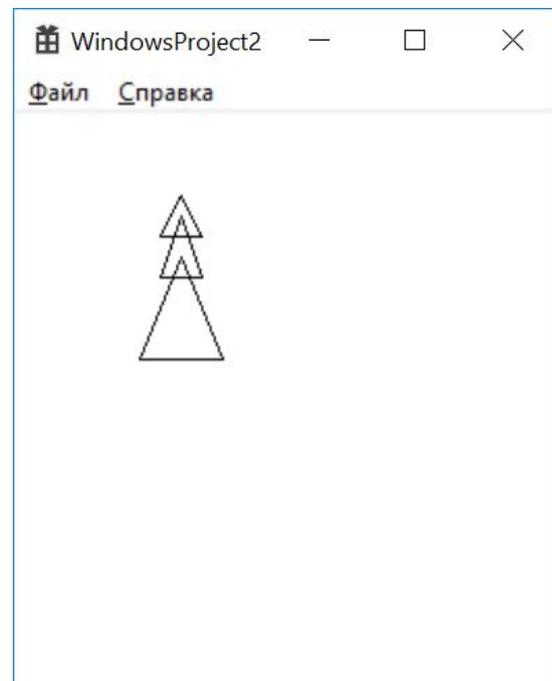
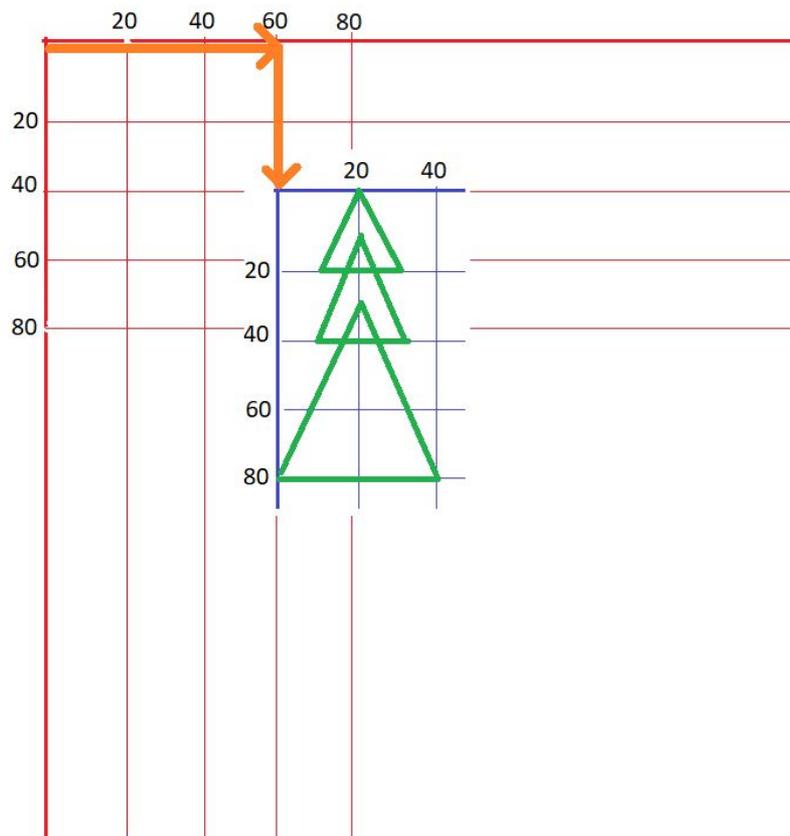
# Вынесли код в отдельную функцию

```
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);  
}  
...
```



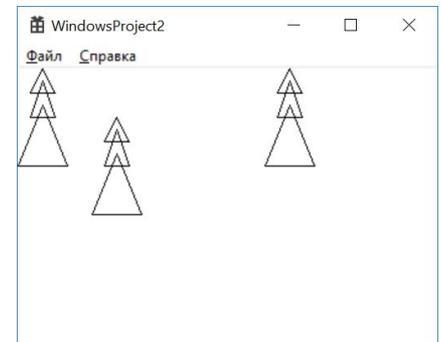
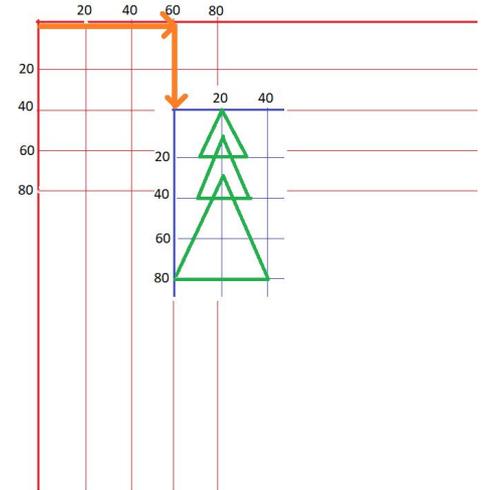
```
case WM_PAINT:  
{  
    PAINTSTRUCT ps;  
    HDC hdc = BeginPaint(hWnd, &ps);  
    // TODO: Добавьте сюда любой код прорисовки, использующий HDC...  
  
    StClausAuto(hdc);  
  
    EndPaint(hWnd, &ps);  
}  
break;
```

# Относительные координаты



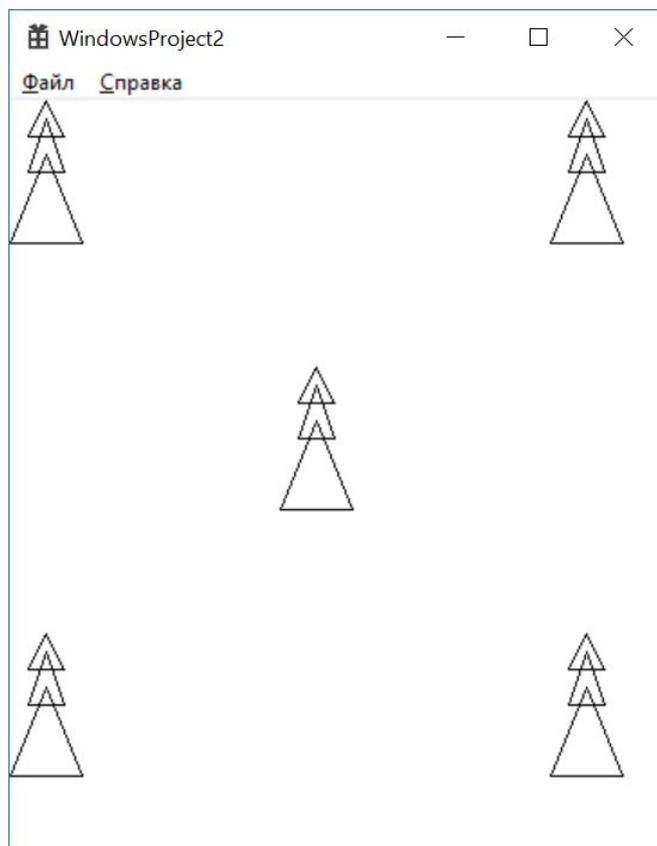
# Относительные координаты

```
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);  
}  
...
```



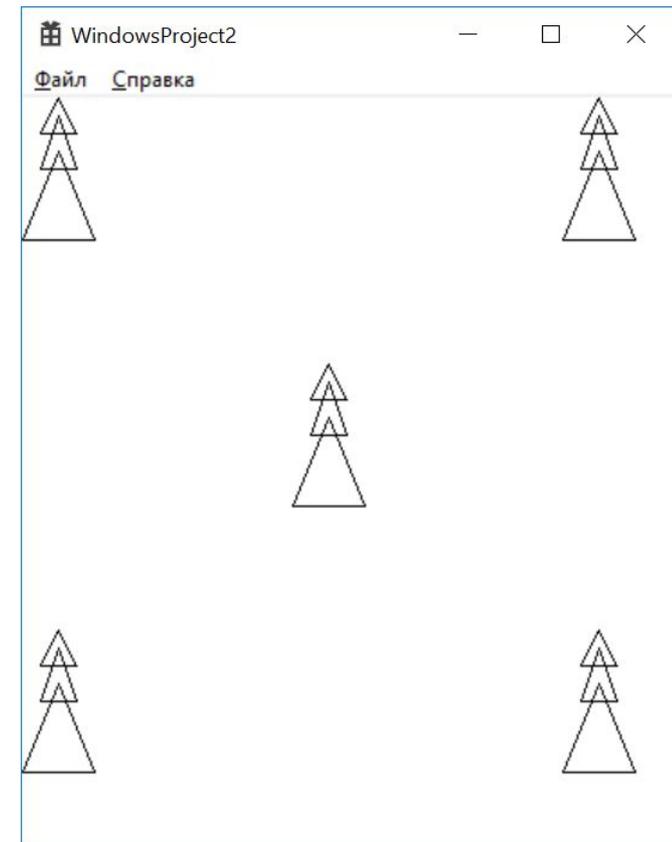
```
case WM_PAINT:  
{  
    PAINTSTRUCT ps;  
    HDC hdc = BeginPaint(hWnd, &ps);  
    // TODO: Добавьте сюда любой код прорисовки, использующий HDC...  
  
    StClausAuto(hdc, 60, 40);  
    StClausAuto(hdc, 0, 0);  
    StClausAuto(hdc, 200, 0);  
  
    EndPaint(hWnd, &ps);  
}  
break;
```

# Задача 1.1: 5 логотипов по углам и в центре

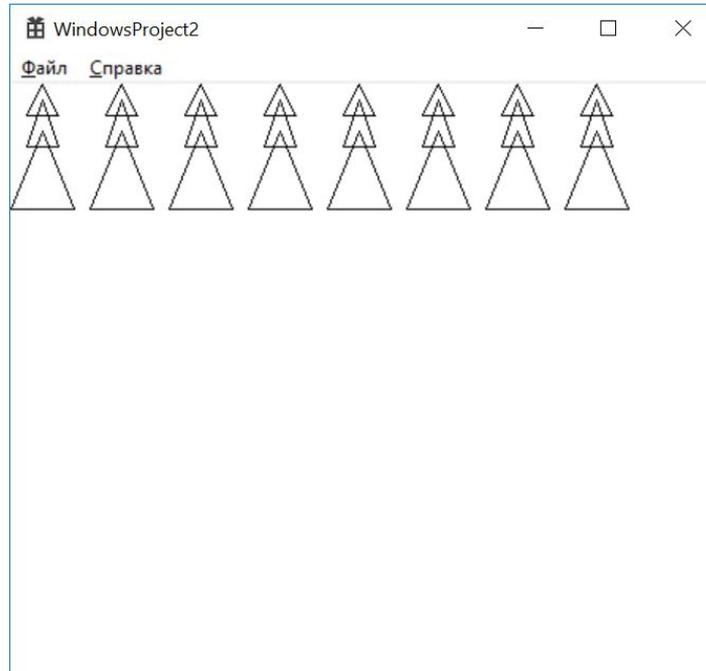


# Задача 1.1: 5 логотипов по углам и в центре

```
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;
```



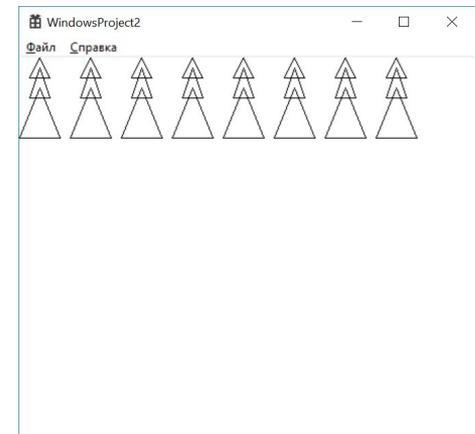
# Задача 1.2: 8 логотипов в горизонтальную ЛИНИЮ



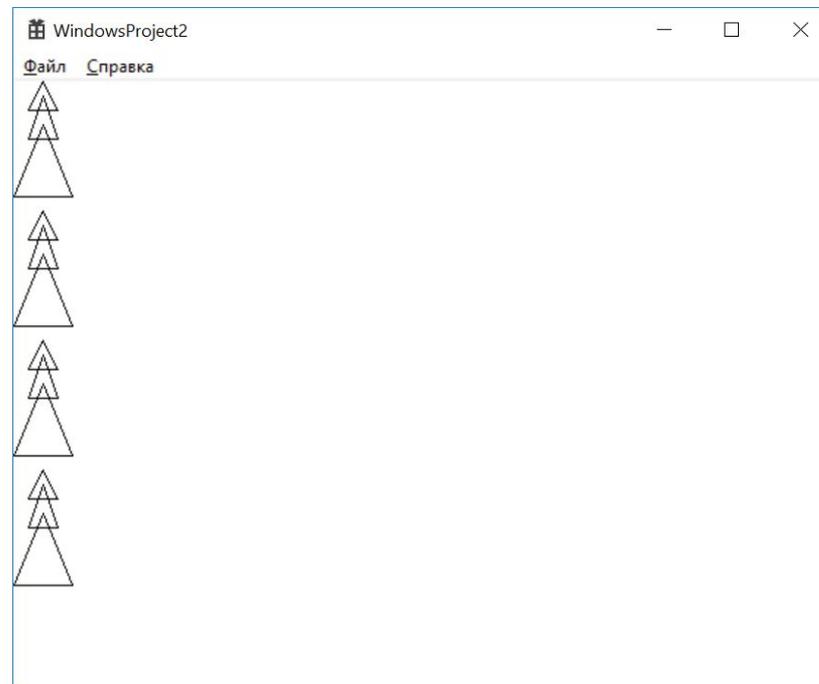
# Задача 1.2: 8 логотипов в горизонтальную линию

## 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;
```

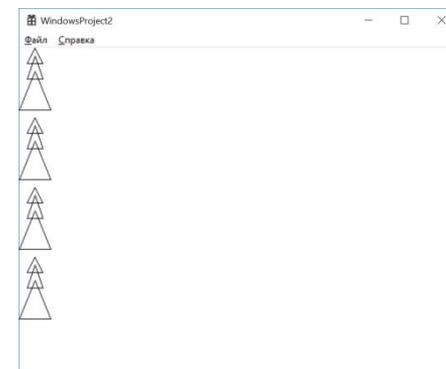


# Задача 1.3: 4 логотипа в вертикальную ЛИНИЮ

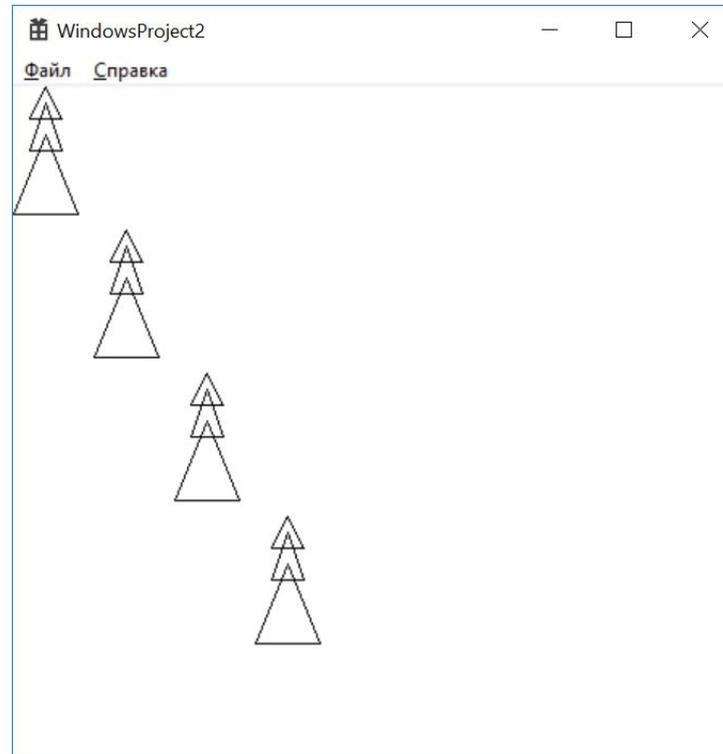


# Задача 1.3 - РЕШЕНИЕ: 4 логотипа в вертикальную линию

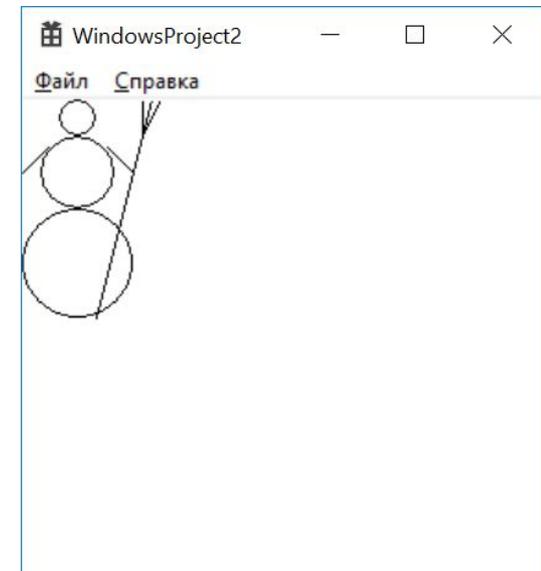
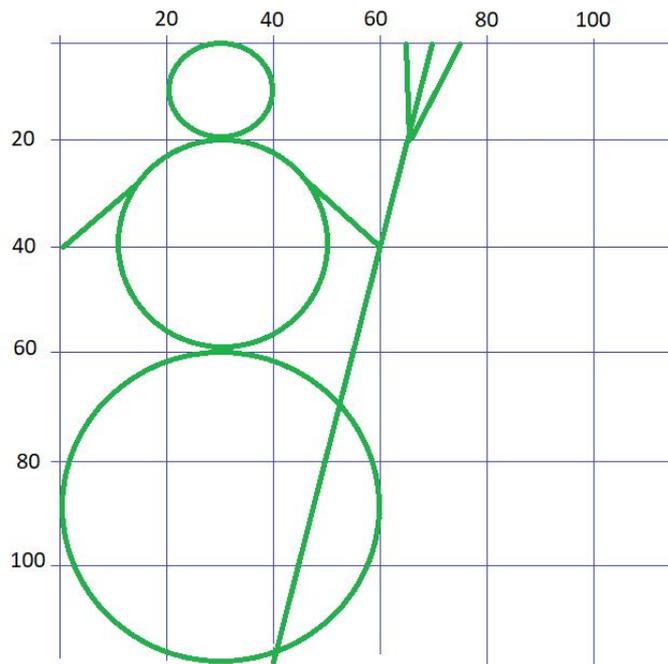
```
int y = 0;  
do {  
    StClausAuto(hdc, 0, y);  
    y += 90;  
} while (y < 300);
```



# Задача 1.4: 4 логотипа в диагональную ЛИНИЮ



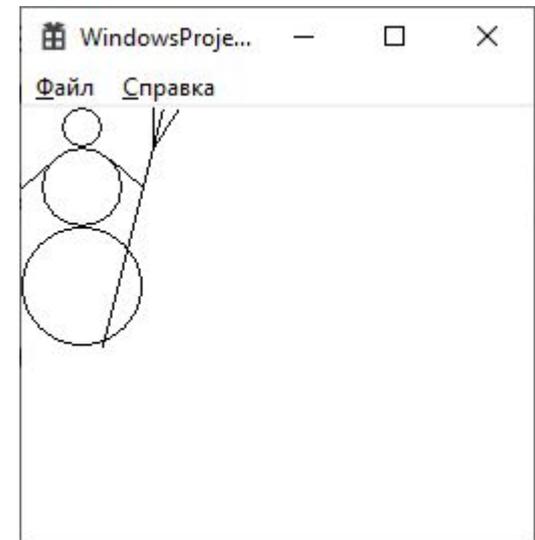
# Задача 2.1 – Создать картинку по образцу



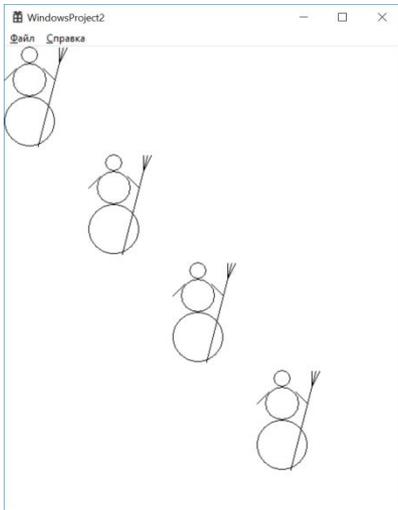
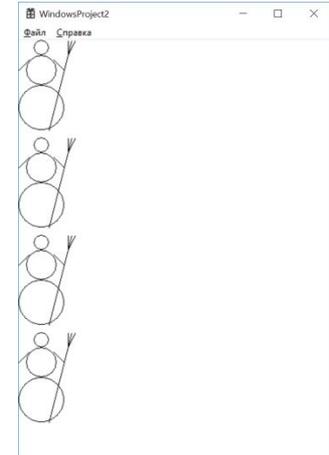
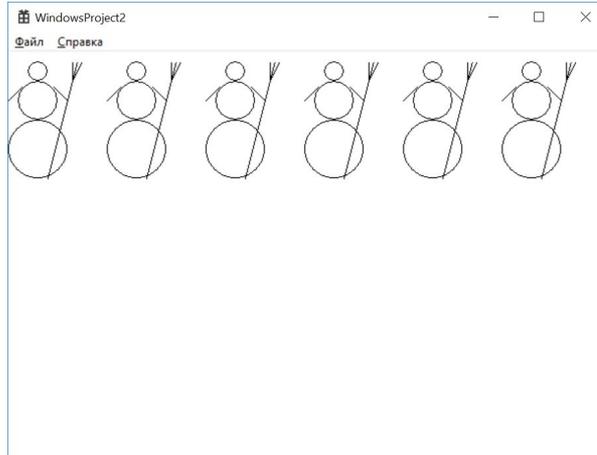
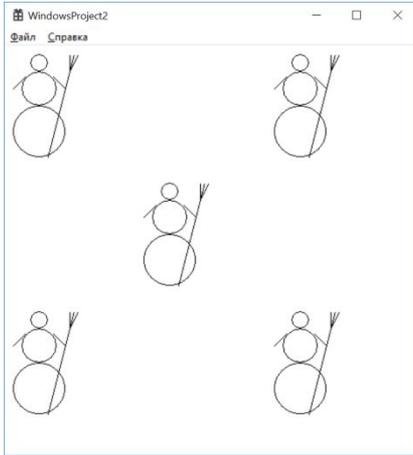
# Задача 2.2 – Создать функцию SnowWoman

```
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;
```

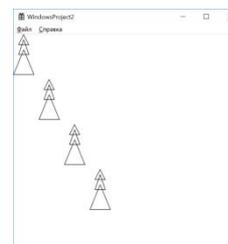
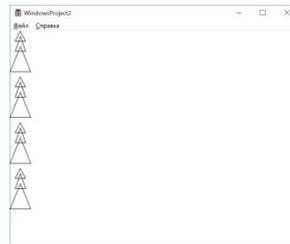
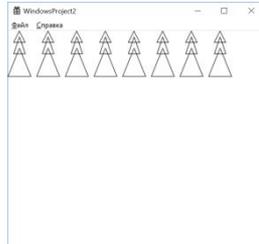
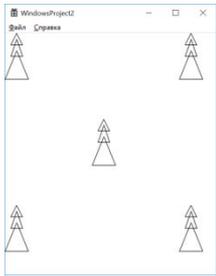


# Задача 2.3 – 2.6 – Создать 4 рисунка из снежных баб



# 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 функция 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 картинка – 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»