

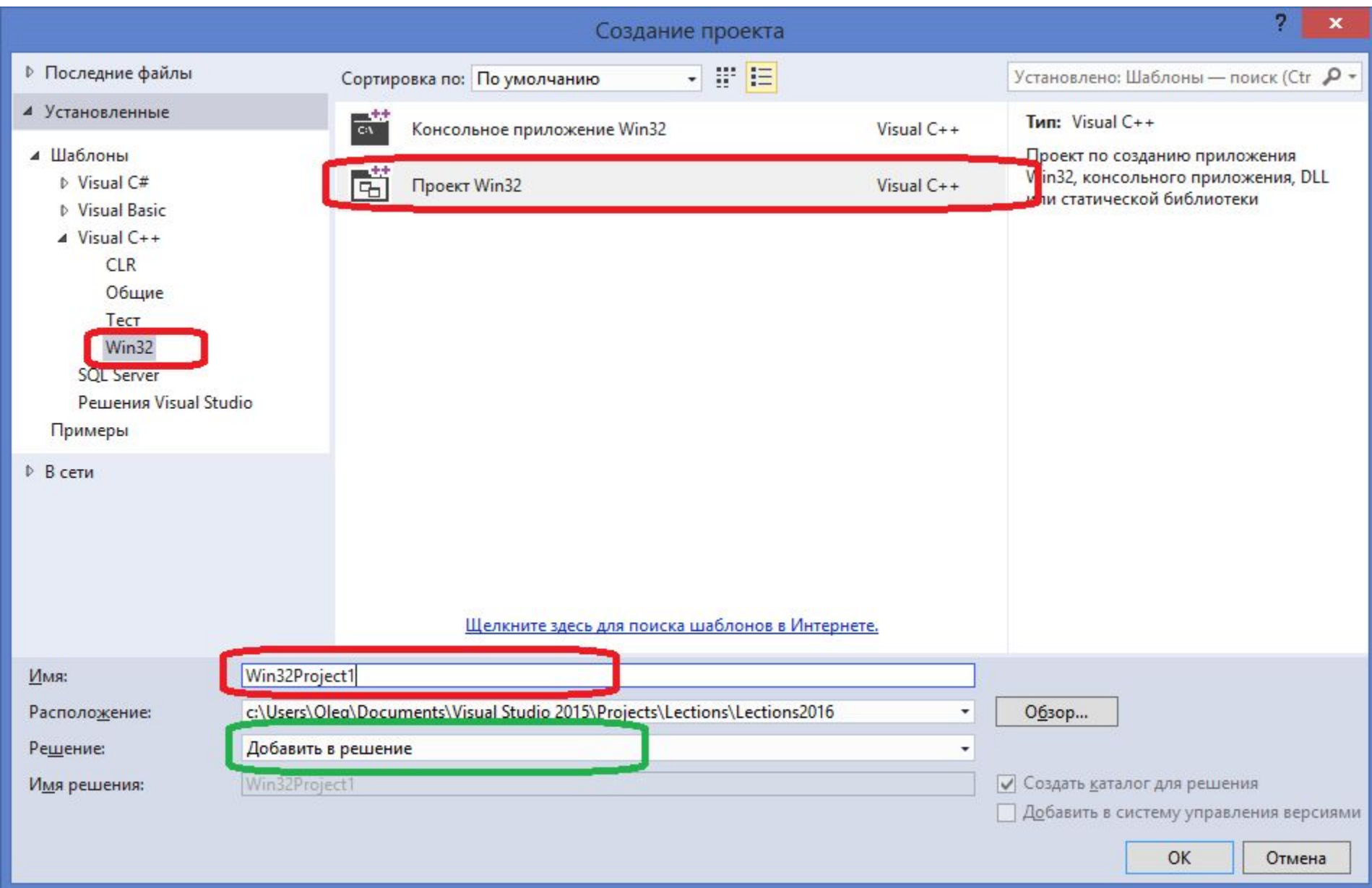
**Программирование и  
алгоритмизация  
Лабораторная работа №1**

**Простейшая  
графика**

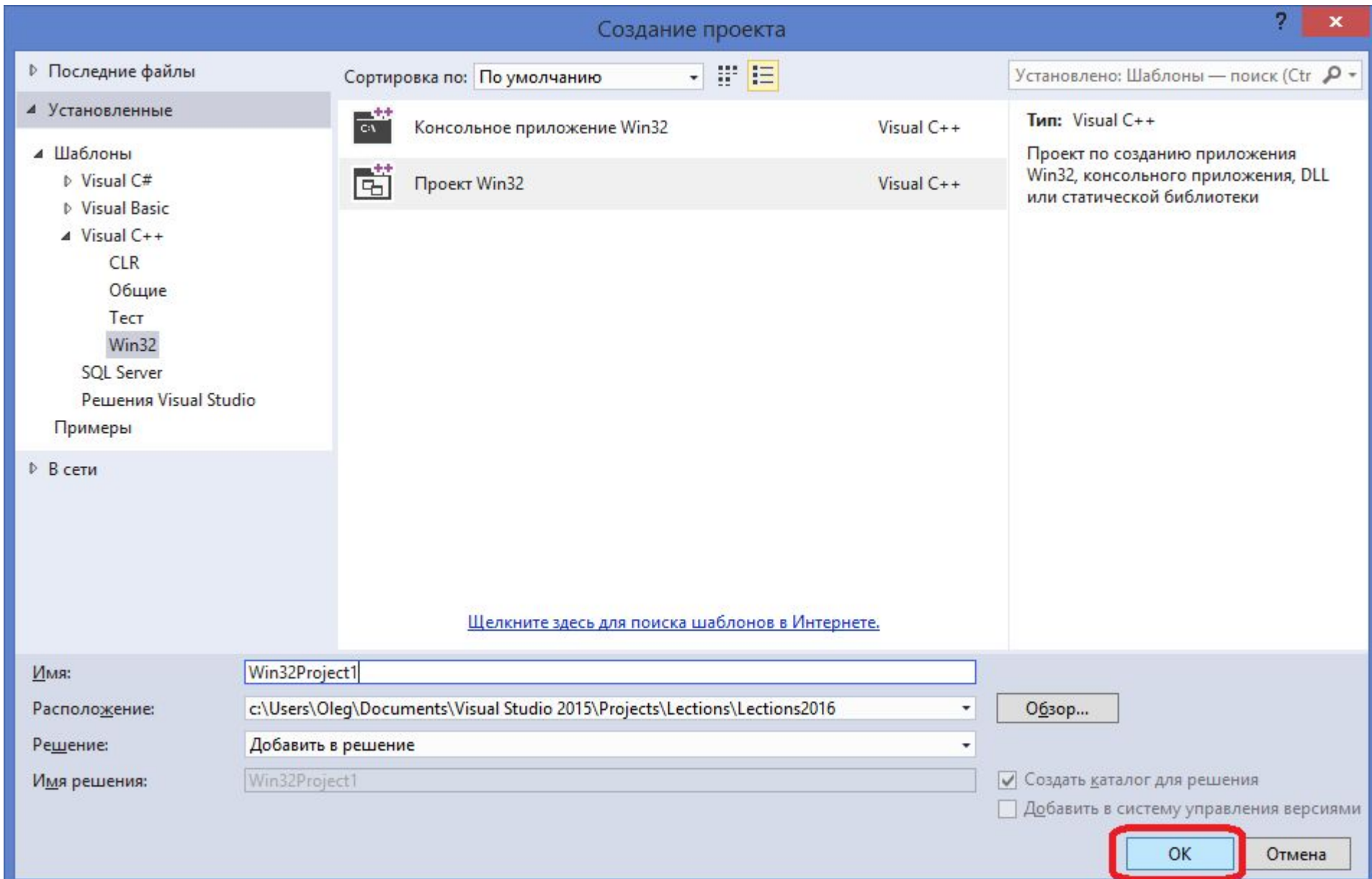
Власенко О.Ф.

# Создание win32 приложения в VS

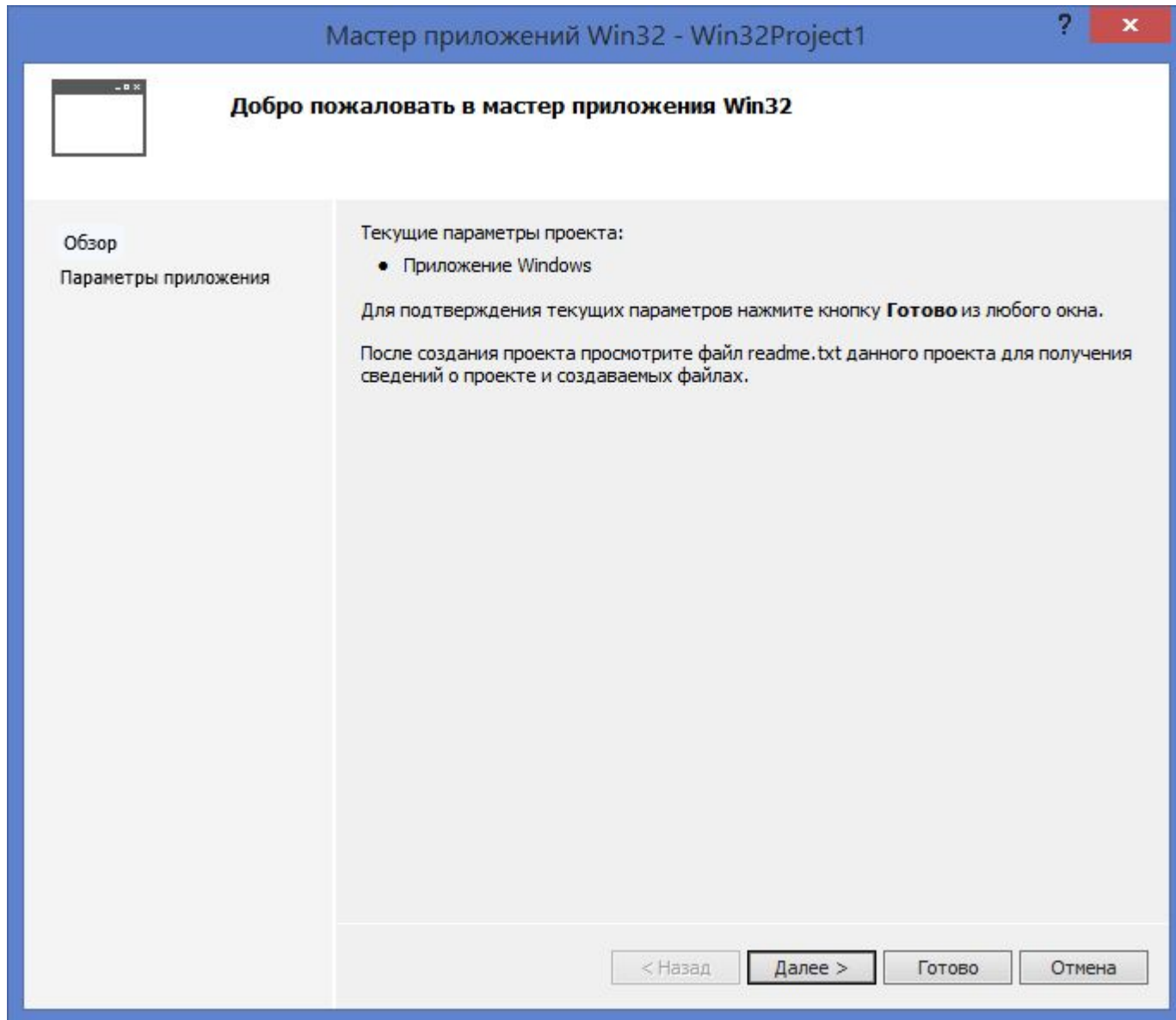
# Создание win32 приложения в VS



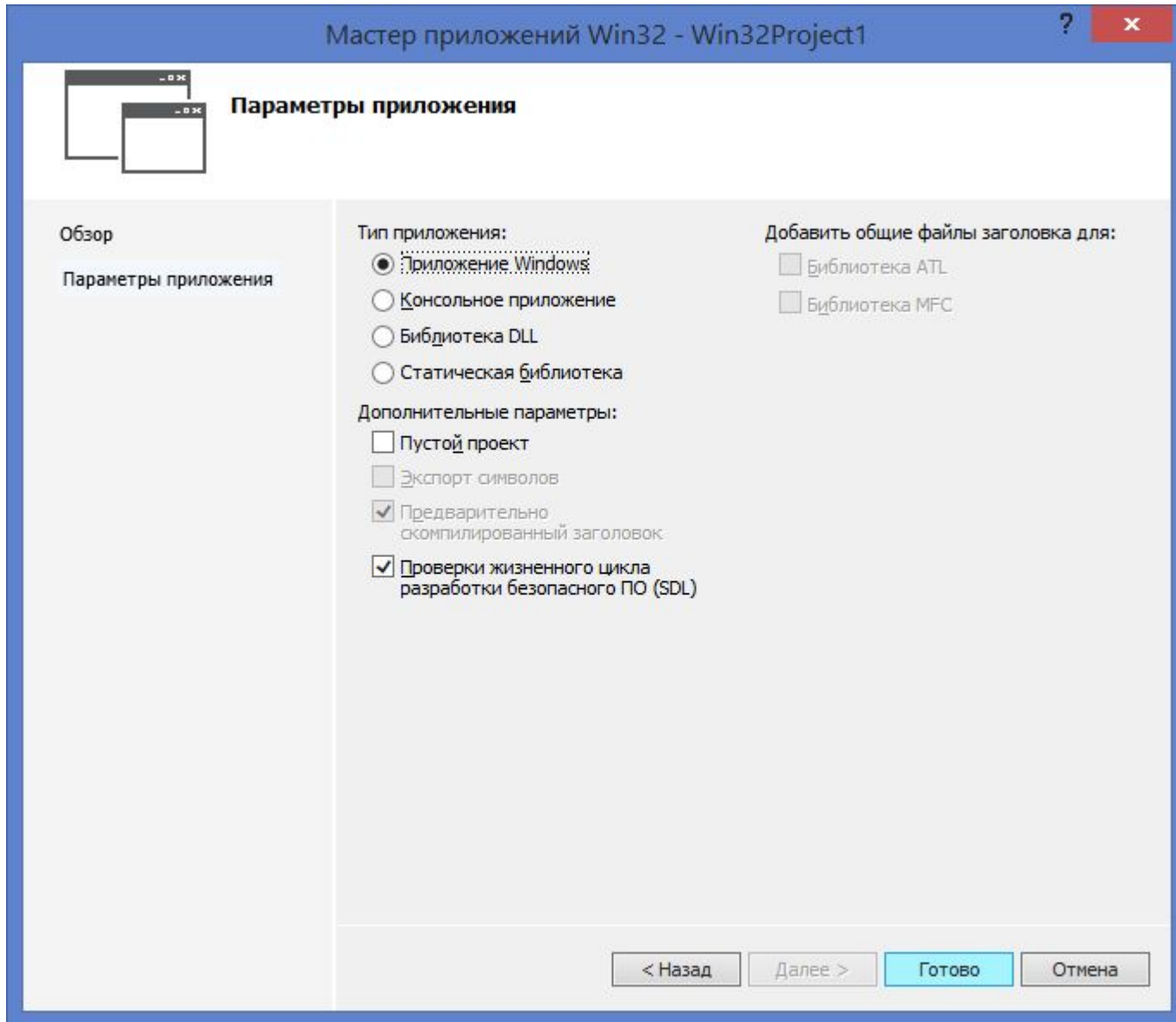
# Создание win32 приложения в VS (2)



# Создание win32 приложения в VS (3)



# Создание win32 приложения в VS (4)



# Создание win32 приложения в VS (5)

Лекции2016 - Microsoft Visual Studio Express 2015 для Windows Desktop

Файл Изменить Просмотр Проект Сборка Отладка Команда Сервис Тест Окно Справка

Debug x86 Локальный отладчик Windows

Обозреватель решений

Обозреватель решений — поиск (Ctrl+;)

Решение "Lectiions2016" (проектов: 2)

- Lection2
  - Внешние зависимости
  - Заголовочные файлы
  - Ссылки
  - Файлы исходного кода
    - Source1.cpp
  - Файлы ресурсов
- Win32Project1
  - Внешние зависимости
  - Заголовочные файлы
    - Resource.h
    - stdafx.h
    - targetver.h
    - Win32Project1.h
  - Ссылки
  - Файлы исходного кода
    - stdafx.cpp
    - Win32Project1.cpp
  - Файлы ресурсов
    - small.ico
    - Win32Project1.ico
    - Win32Project1.rc
    - ReadMe.txt

```
// Win32Project1.cpp: определяет точку входа для приложения.
//

#include "stdafx.h"
#include "Win32Project1.h"

#define MAX_LOADSTRING 100

// Глобальные переменные:
HINSTANCE hInst; // текущий экземпляр
WCHAR szTitle[MAX_LOADSTRING]; // Текст строки заголовка
WCHAR szWindowClass[MAX_LOADSTRING]; // имя класса главного окна

// Отправить объявления функций, включенных в этот модуль кода:
ATOM MyRegisterClass(HINSTANCE hInstance);
BOOL InitInstance(HINSTANCE, int);
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
```

Список ошибок

Все решение 0 Ошибки 0 Предупреждения 0 Сообщения Сборка и IntelliSense Поиск по списку ошибок

Код	Описание	Проект	Файл	Ст...
-----	----------	--------	------	-------

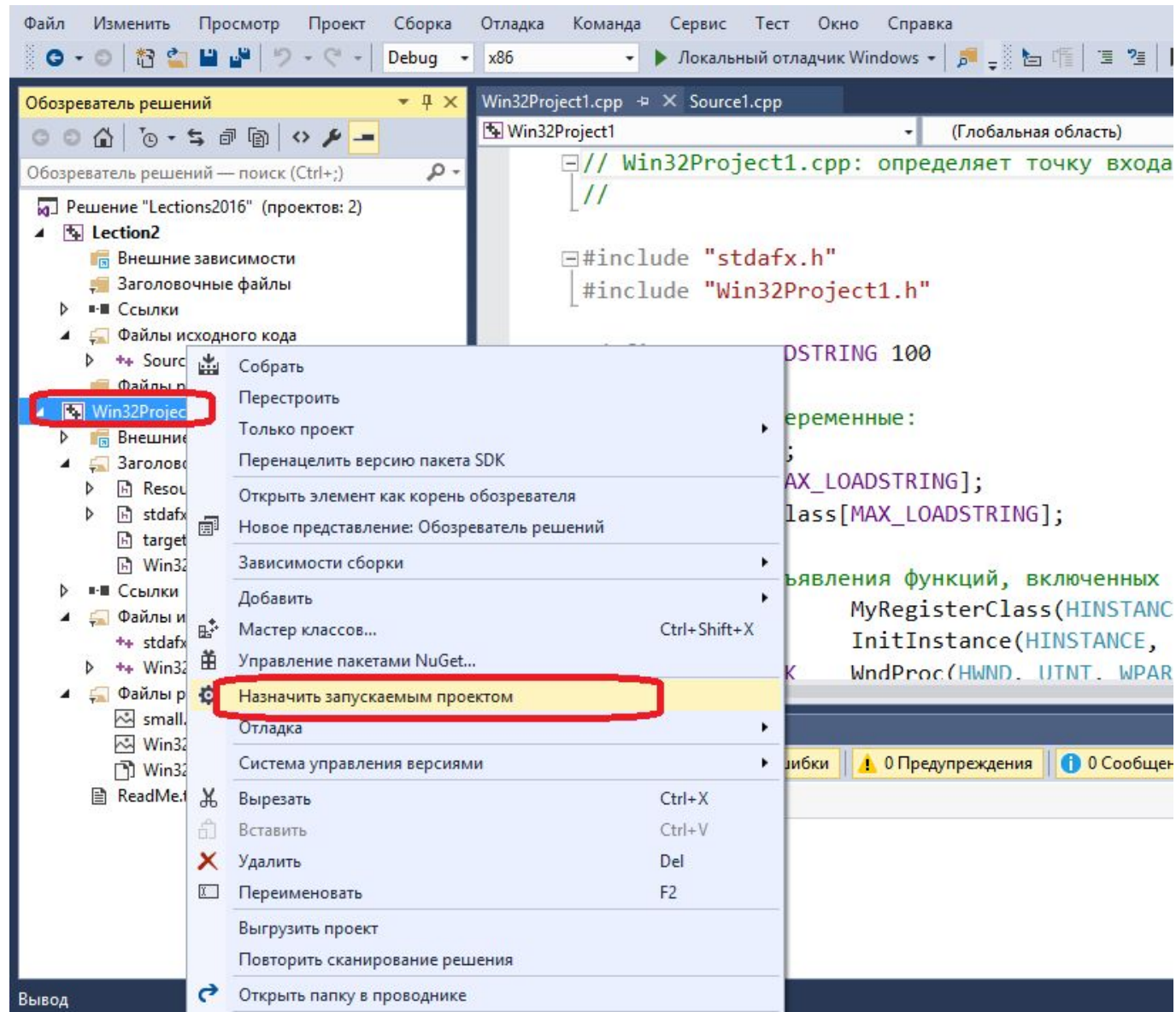
Вывод

Синтаксический анализ включенных файлов... (33 из 48) - c:\Program Files (x86)\Windows Kits\8.1\Include\um\winspool.h

Строка 1 Столбец 1 Знак 1 ВСТ

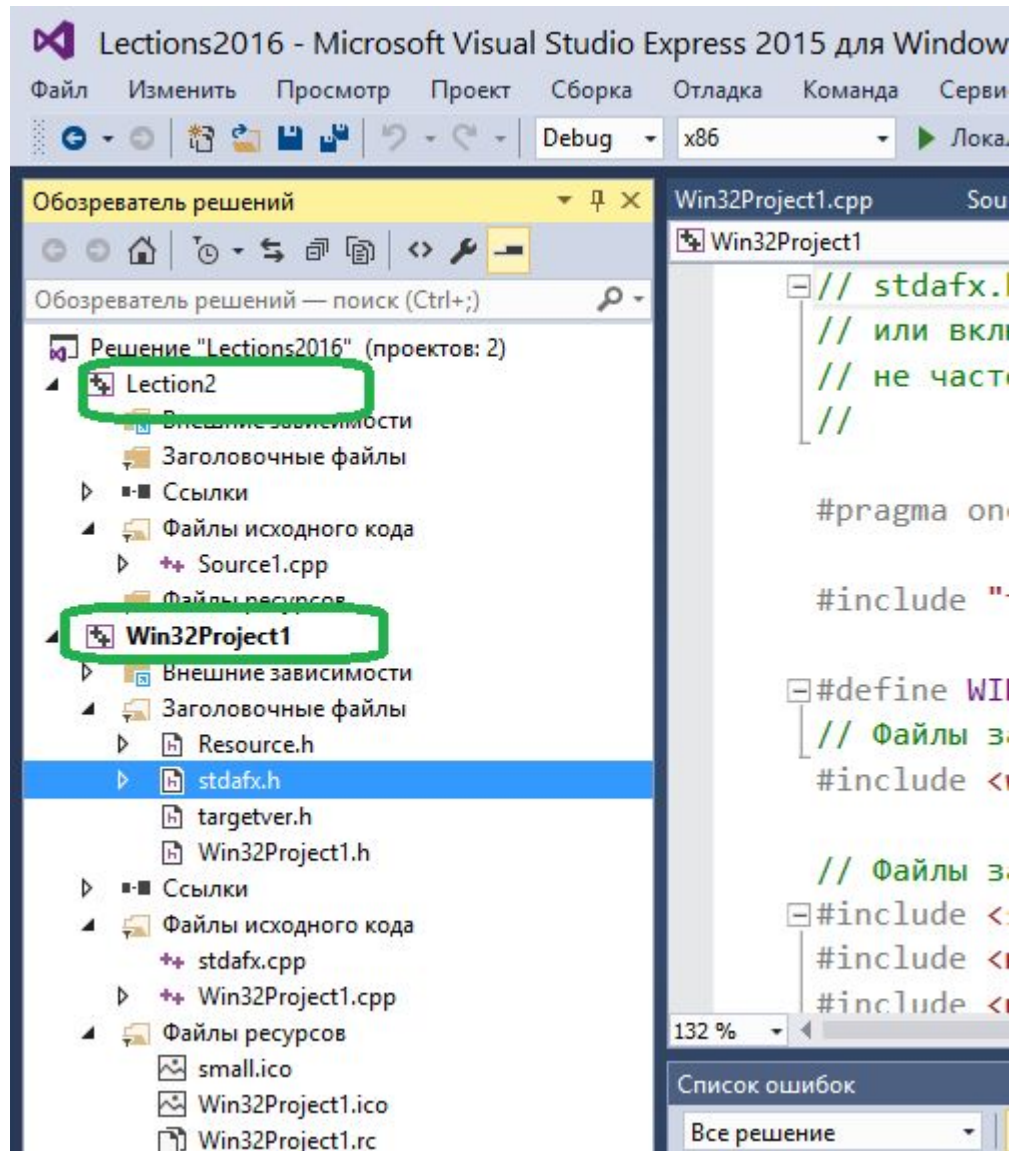


# Создание win32 приложения в VS (6)

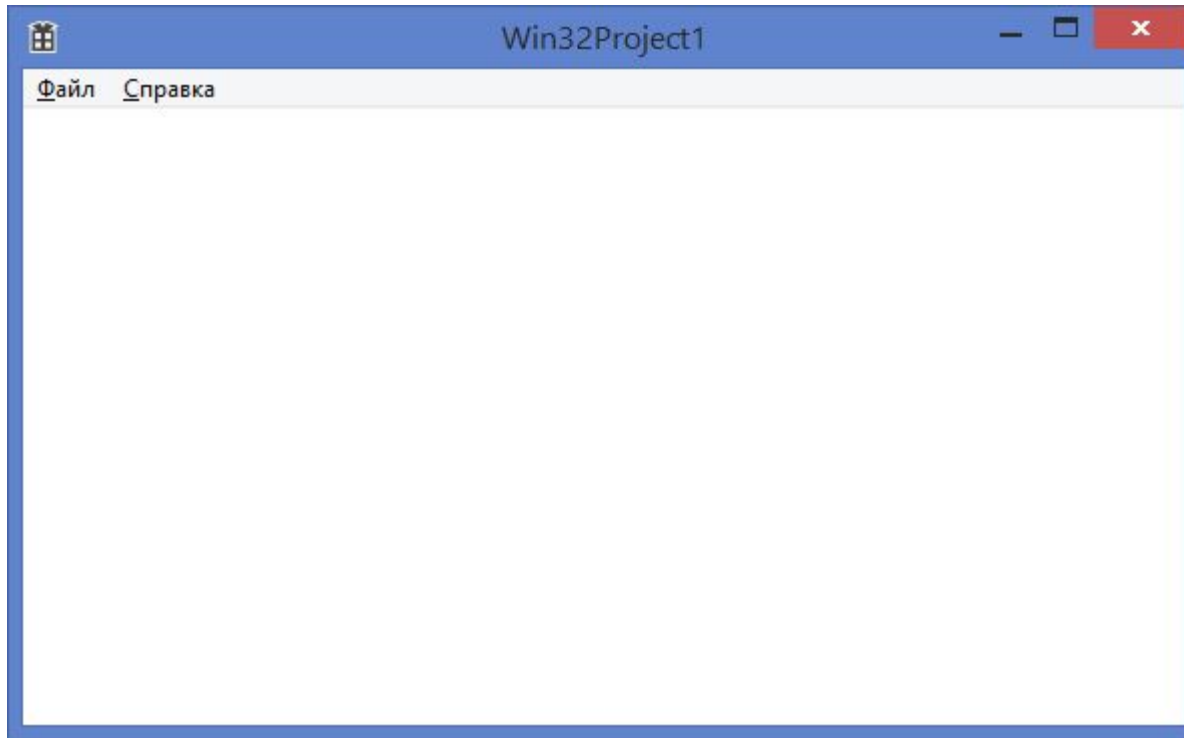




# Создание win32 приложения в VS (7) – создано!



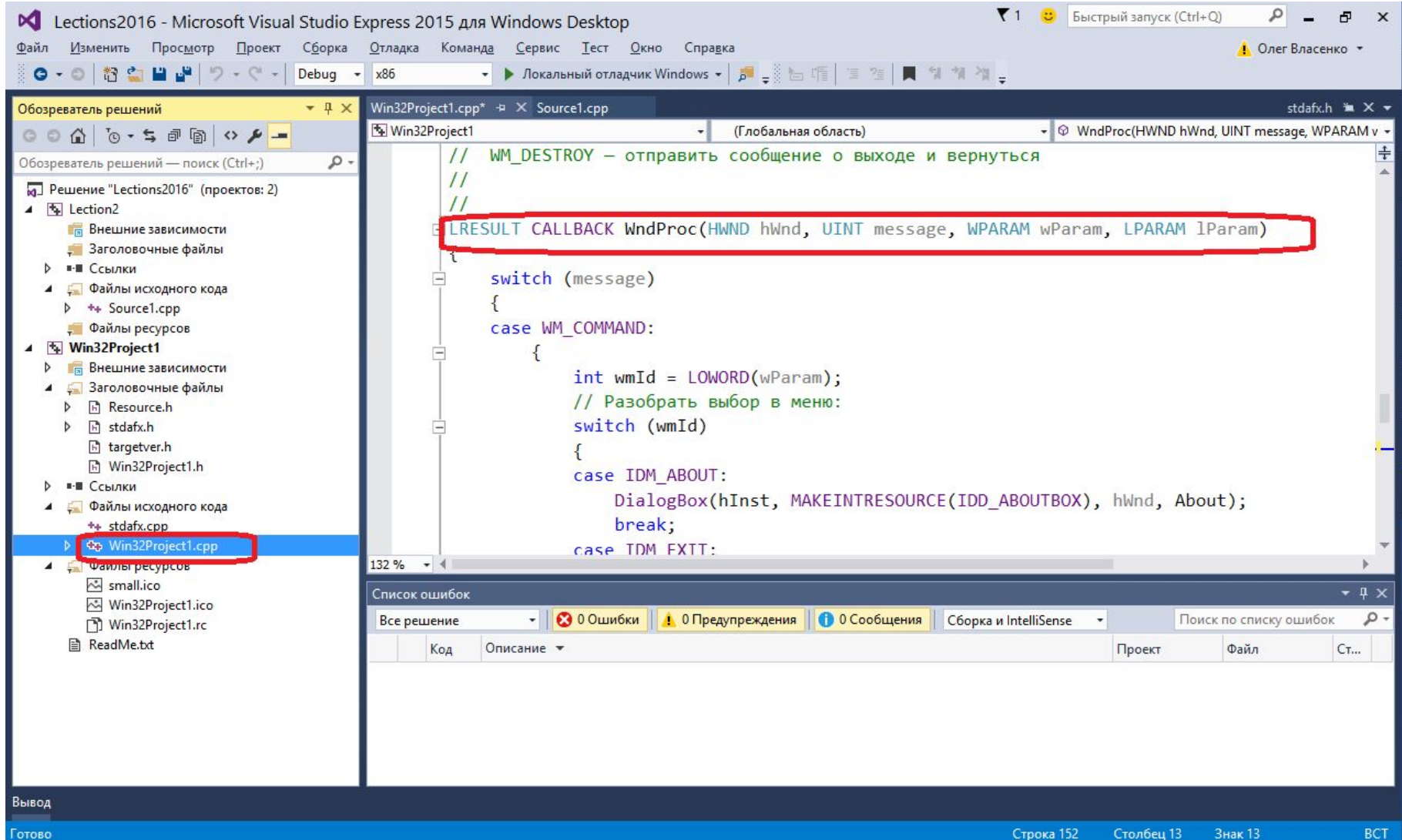
# Запущенное win32 приложение!



# Где в коде рисовать картинки?

Файл Win32Project1.cpp

Функция WndProc()



The screenshot shows the Microsoft Visual Studio Express 2015 IDE. The main window displays the source code for Win32Project1.cpp. The function signature for WndProc is highlighted with a red rectangle:

```
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
```

The code also includes comments in Russian: `// WM_DESTROY – отправить сообщение о выходе и вернуться`, `//`, `//`, and `// Разобрать выбор в меню:`. The function body contains a `switch (message)` block with a `case WM_COMMAND:` section, which further contains a `switch (wmId)` block with `case IDM_ABOUT:` and `case IDM_FXTT:` cases.

The Solution Explorer on the left shows the project structure, with `Win32Project1.cpp` selected and highlighted in blue. The Error List at the bottom shows 0 errors, 0 warnings, and 0 messages.

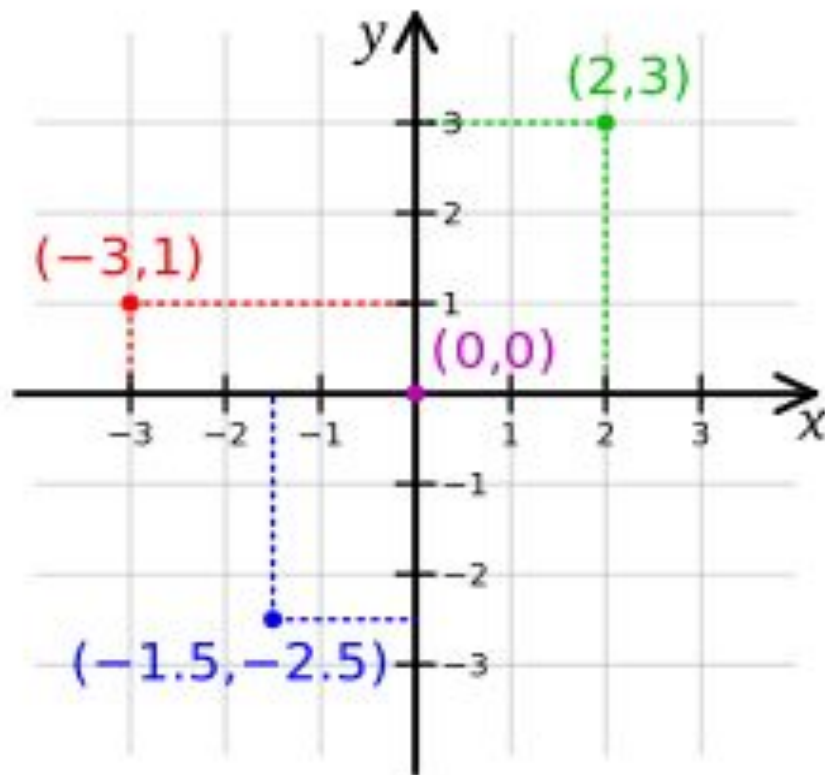
Вывод  
Готово

Строка 152    Столбец 13    Знак 13    ВСТ

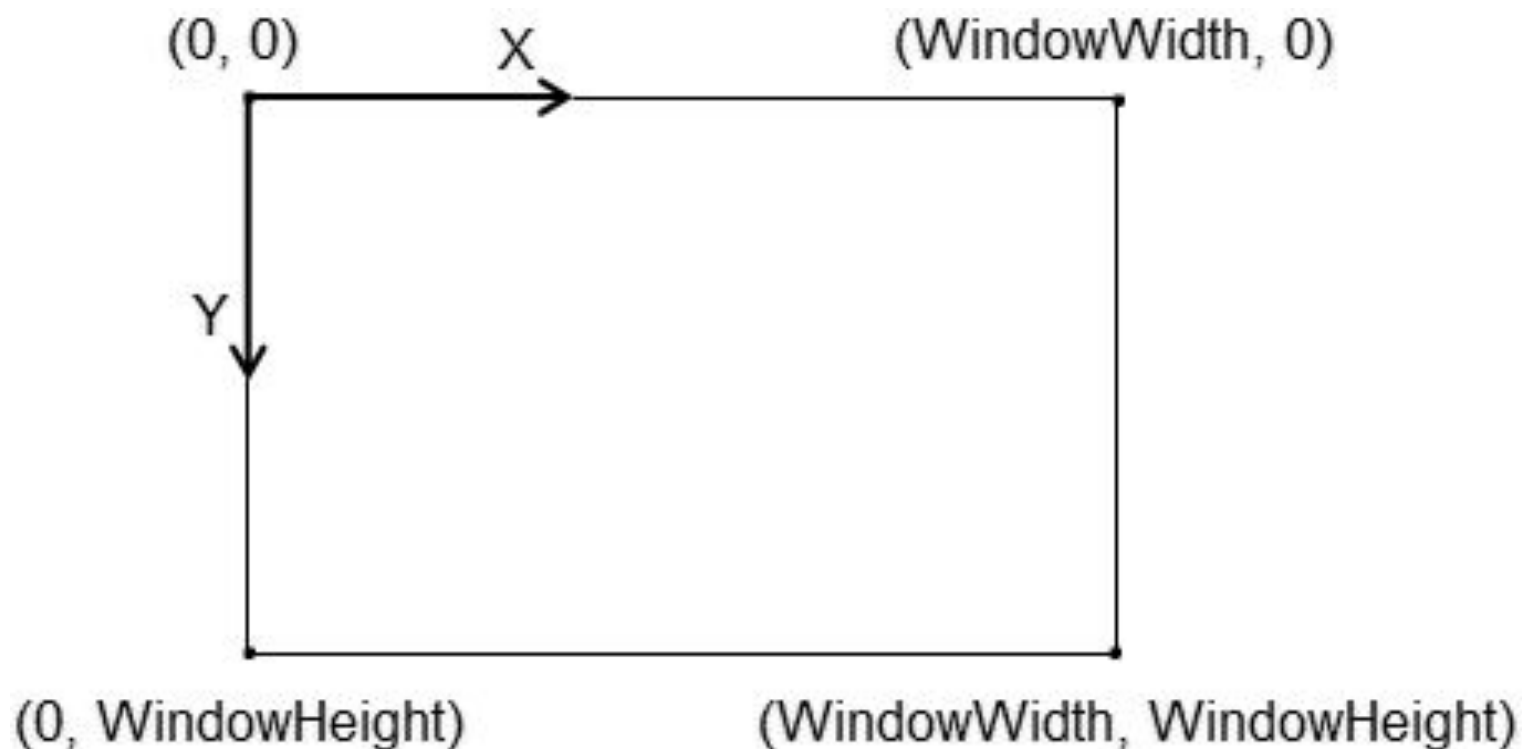
# Где в коде рисовать картинки? (2)

```
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam,
LPARAM lParam)
{
    switch (message)
    {
        ...
        case WM_PAINT:
            {
                PAINTSTRUCT ps;
                HDC hdc = BeginPaint(hWnd, &ps);
                // TODO: Добавьте сюда любой код прорисовки, использующий
                HDC...
                EndPaint(hWnd, &ps);
            }
            break;
        ...
        default:
            return DefWindowProc(hWnd, message, wParam, lParam);
    }
    return 0;
}
```

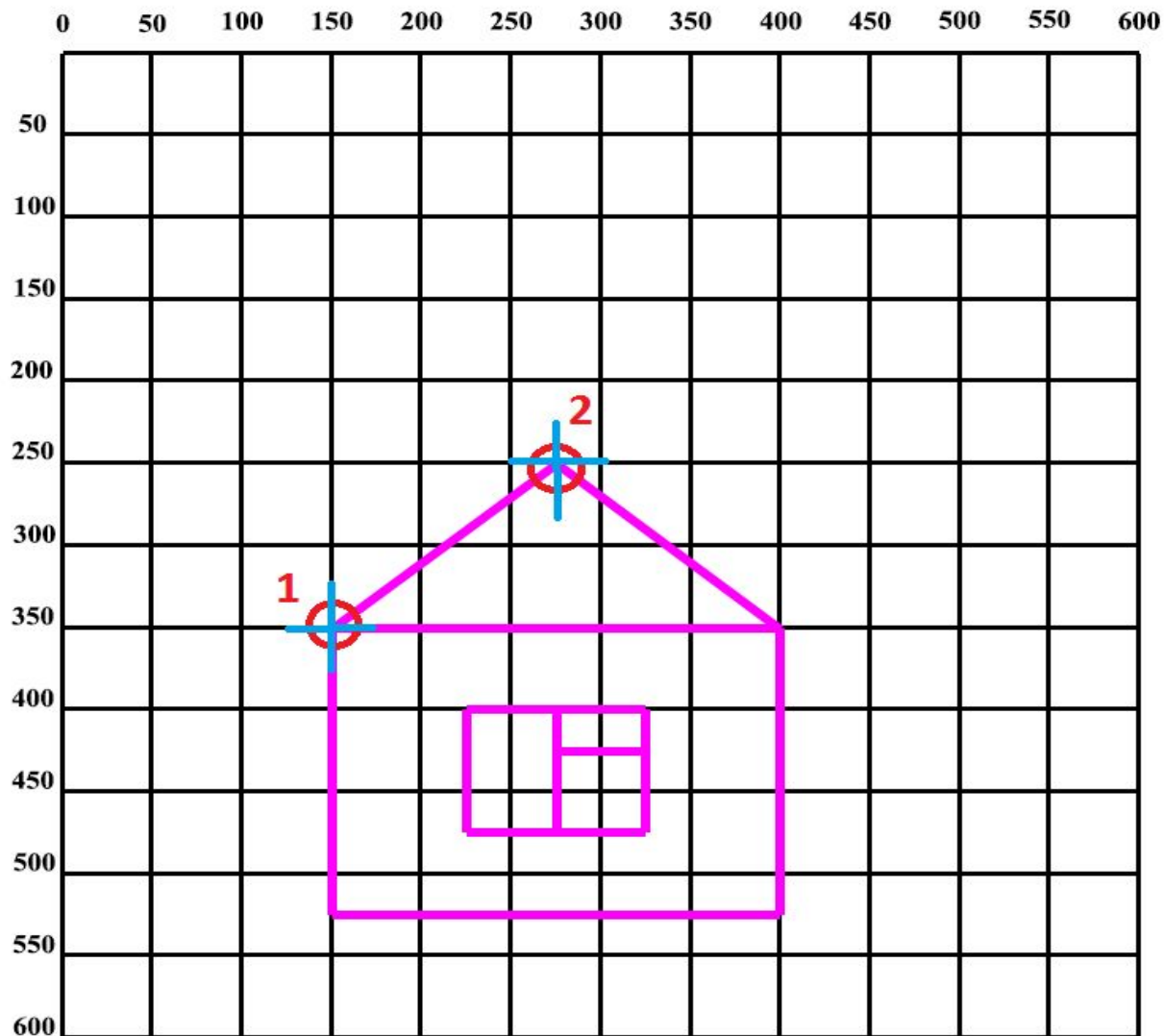
# Точки в Декартовой системе координат



# Экранная система координат



# Оцифровка точек в координатной сетке





# Рисуем линии

```
case WM_PAINT:
```

```
{
```

```
    PAINTSTRUCT ps;
```

```
    HDC hdc = BeginPaint(hWnd, &ps);
```

```
// Перемещаем "курсор" рисования линии в точку (x = 10, y = 30)
```

```
    MoveToEx(hdc, 10, 30, NULL);
```

```
// Рисуем линию из текущей позиции курсора в точку (x = 10, y = 100)
```

```
// "Курсор" после отрисовки находится в новой точке (x = 10, y = 100)
```

```
    LineTo(hdc, 10, 100);
```

```
// Рисуем линию от предыдущей точки (x = 10, y = 100) до точки (x = 150, y = 100)
```

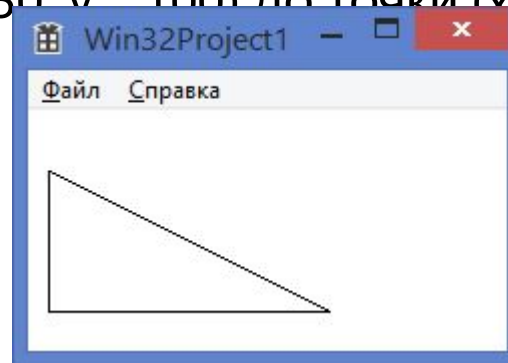
```
    LineTo(hdc, 150, 100);
```

```
// Рисуем линию от предыдущей точки (x = 150, y = 100) до точки (x = 10, y = 30)
```

```
    LineTo(hdc, 10, 30);
```

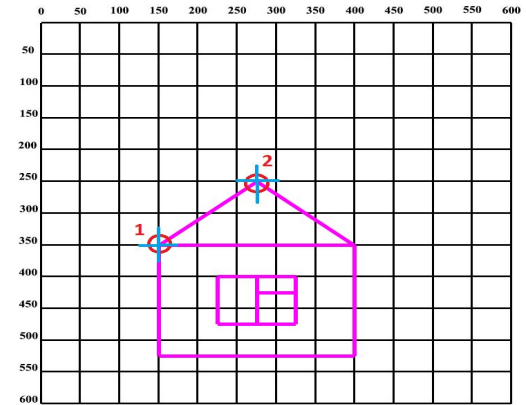
```
    EndPaint(hWnd, &ps);
```

```
}
```



# Задача 1

В Windows приложение добавить рисунок домика.



# Задача 1: Решение

В Windows приложение добавить рисунок домика.

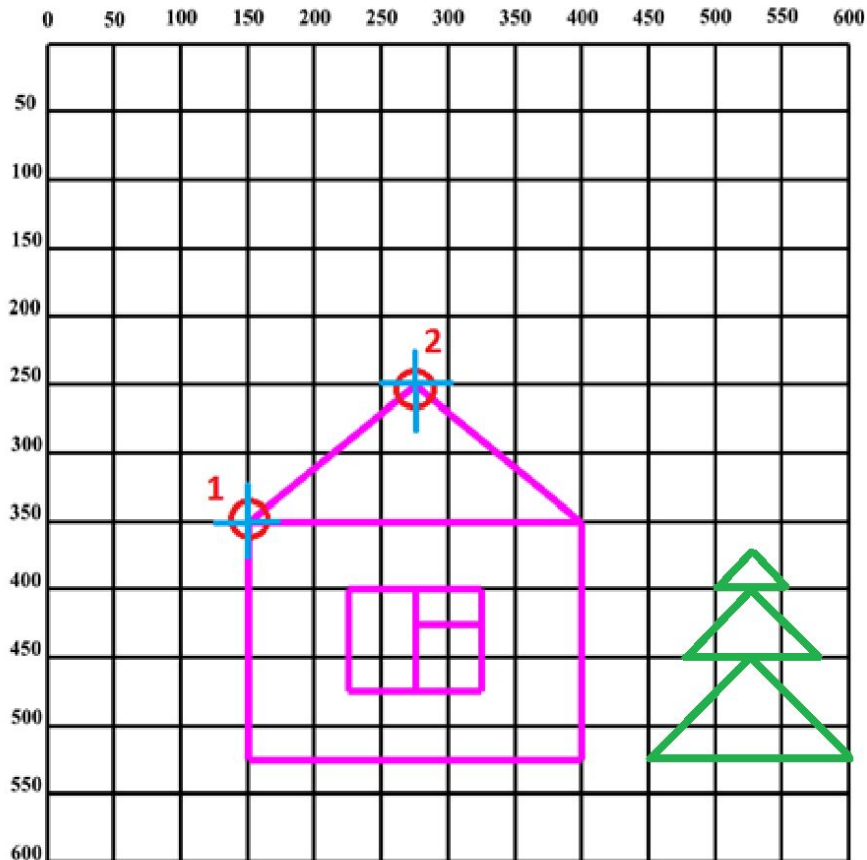
```
case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: Добавьте сюда любой код прорисовки, использующий HDC...

    // Крыша
    MoveToEx(hdc, 150, 350, NULL);
    LineTo(hdc, 275, 250);
    LineTo(hdc, 400, 350);
    // Дом
    LineTo(hdc, 400, 525);
    LineTo(hdc, 150, 525);
    LineTo(hdc, 150, 350);
    LineTo(hdc, 400, 350);
    // Окно
    MoveToEx(hdc, 225, 400, NULL);
    LineTo(hdc, 225, 475);
    LineTo(hdc, 325, 475);
    LineTo(hdc, 325, 400);
    LineTo(hdc, 225, 400);
    // Рама
    MoveToEx(hdc, 275, 400, NULL);
    LineTo(hdc, 275, 475);
    MoveToEx(hdc, 275, 425, NULL);
    LineTo(hdc, 325, 425);

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

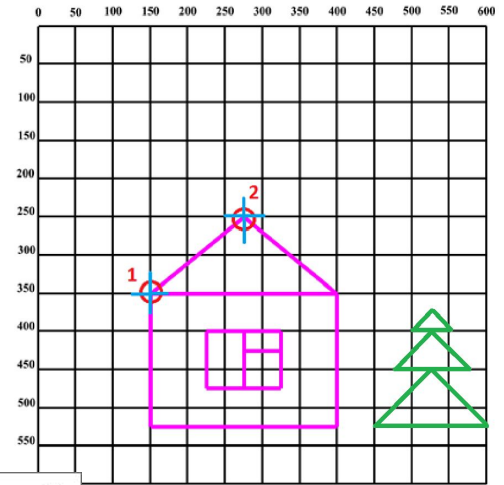
# Задача 2

Рядом с домиком добавь ёлочку



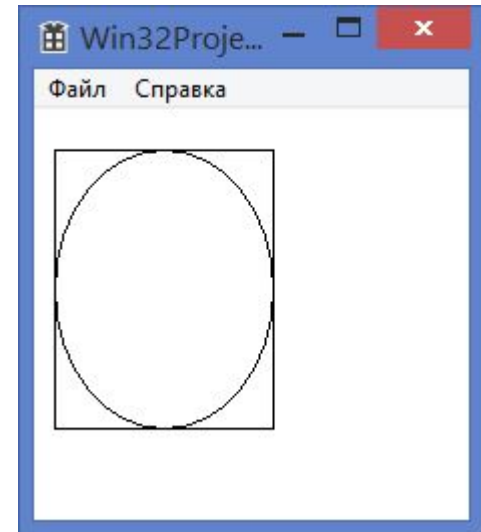
# Задача 2

Рядом с домиком добавьте ёлочку



# Рисуем эллипс

```
case WM_PAINT:  
    {  
        PAINTSTRUCT ps;  
        HDC hdc = BeginPaint(hWnd, &ps);  
  
        // Рисуем фиксированный прямоугольник  
        Rectangle(hdc, 10, 20, 120, 160);  
  
        // Рисуем эллипс, вписанный в прямоугольник  
        Ellipse(hdc, 10, 20, 120, 160);  
  
        EndPaint(hWnd, &ps);  
    }
```

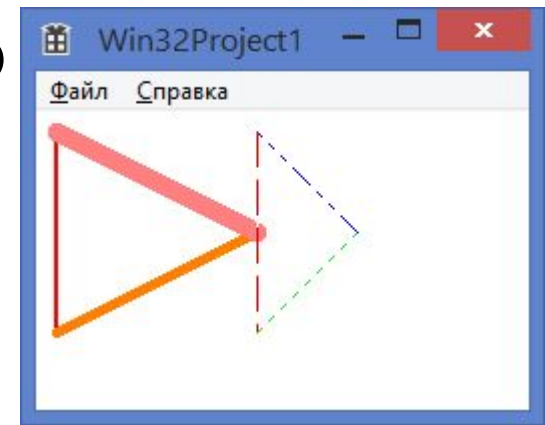


# Такое разное перо

```
case WM_PAINT: {
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
HPEN hPen;
hPen = CreatePen(PS_SOLID, 2, RGB(255, 0, 0));
SelectObject(hdc, hPen);
MoveToEx(hdc, 10, 10, NULL);
LineTo(hdc, 10, 110);

hPen = CreatePen(PS_SOLID, 5, RGB(255, 128, 0));
SelectObject(hdc, hPen);
LineTo(hdc, 110, 60);

hPen = CreatePen(PS_SOLID, 10, RGB(255, 128, 128));
SelectObject(hdc, hPen);
LineTo(hdc, 10, 10);
}
```





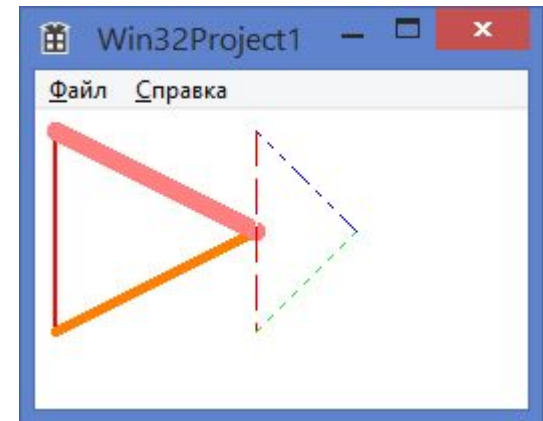
# Такое разное перо (2)

```
hPen = CreatePen(PS_DASH, 1, RGB(255, 0, 0));  
SelectObject(hdc, hPen);  
MoveToEx(hdc, 110, 10, NULL);  
LineTo(hdc, 110, 110);
```

```
hPen = CreatePen(PS_DOT, 1, RGB(0, 255, 0));  
SelectObject(hdc, hPen);  
LineTo(hdc, 160, 60);
```

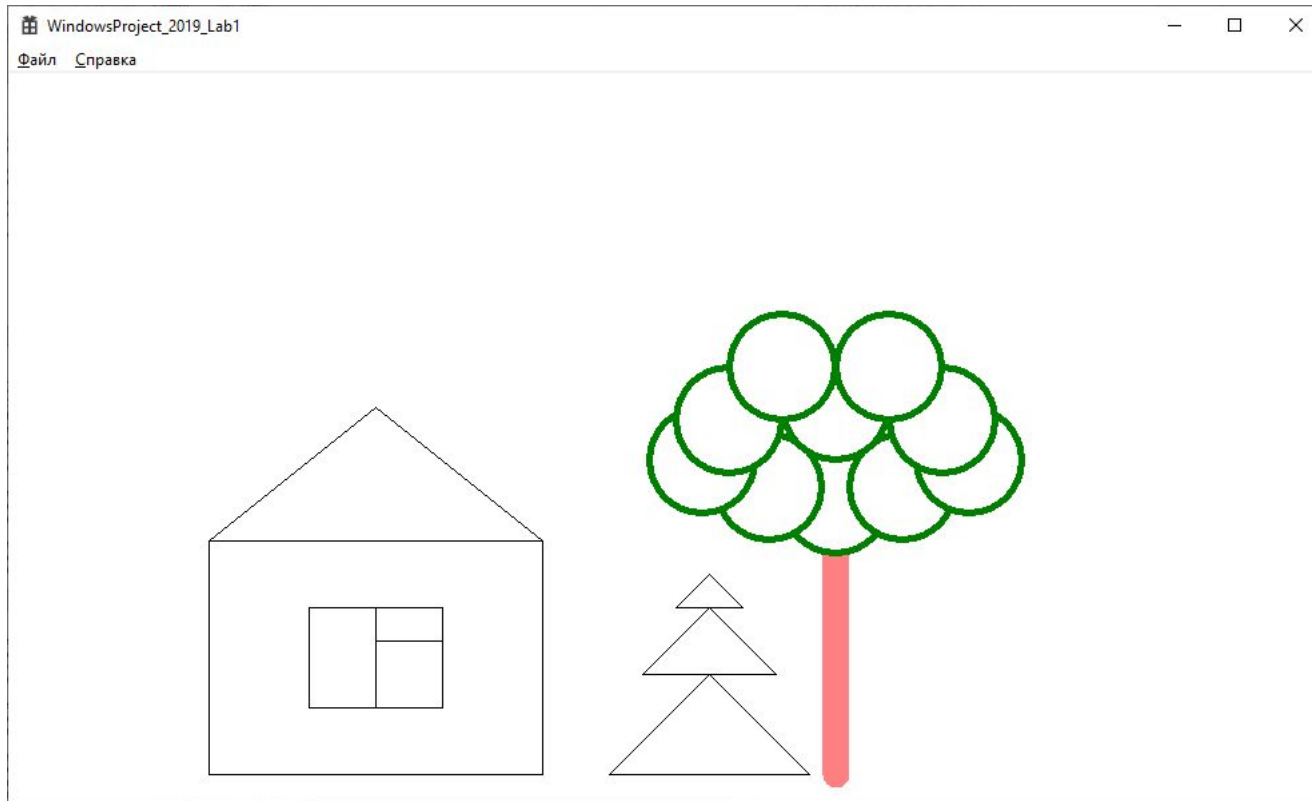
```
hPen = CreatePen(PS_DASHDOTDOT, 1, RGB(0, 0, 255));  
SelectObject(hdc, hPen);  
LineTo(hdc, 110, 10);  
DeleteObject(hPen);
```

```
EndPoint(hWnd, &ps);  
}
```



# Задача 3

Рядом с ёлкой добавьте дерево (по вашей схеме – но состоящее из эллипсов)



# Задача 3: Решение

Моё решение. Ваше может быть сильно другим

```
// Ствол дерева
```

```
HPEN hPen;  
hPen = CreatePen(PS_SOLID, 20, RGB(255, 128, 128));  
SelectObject(hdc, hPen);  
MoveToEx(hdc, 620, 525, NULL);  
LineTo(hdc, 620, 350);
```

```
// Крона дерева
```

```
hPen = CreatePen(PS_SOLID, 5, RGB(0, 128, 0));  
SelectObject(hdc, hPen);  
Ellipse(hdc, 580, 280, 660, 360);  
Ellipse(hdc, 580-50, 280-10, 660-50, 360-10);  
Ellipse(hdc, 580+50, 280-10, 660+50, 360-10);  
Ellipse(hdc, 580-100, 280-30, 660-100, 360-30);  
Ellipse(hdc, 580+100, 280-30, 660+100, 360-30);  
Ellipse(hdc, 580-80, 280-60, 660-80, 360-60);  
Ellipse(hdc, 580 + 80, 280 - 60, 660 + 80, 360 - 60);  
Ellipse(hdc, 580 + 0, 280 - 70, 660 + 0, 360 - 70);  
Ellipse(hdc, 580 - 40, 280 - 100, 660 - 40, 360 - 100);  
Ellipse(hdc, 580 + 40, 280 - 100, 660 + 40, 360 - 100);
```

# Кисть

```
case WM_PAINT: {  
    PAINTSTRUCT ps;  
    HDC hdc = BeginPaint(hWnd, &ps);
```

```
HBRUSH hBrush;
```

```
hBrush = CreateSolidBrush(RGB(0, 0, 255));
```

```
SelectObject(hdc, hBrush);
```

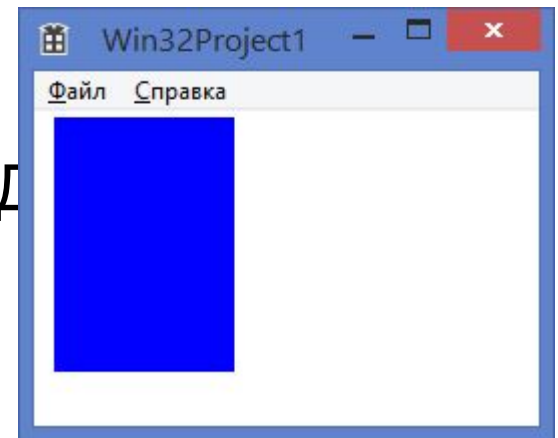
```
RECT rect = { 10, 3, 100, 130 };
```

```
FillRect(hdc, &rect, hBrush);
```

```
// СЛЕДУЮЩИЙ КОД ВСТАВИТЬ СЮДА
```

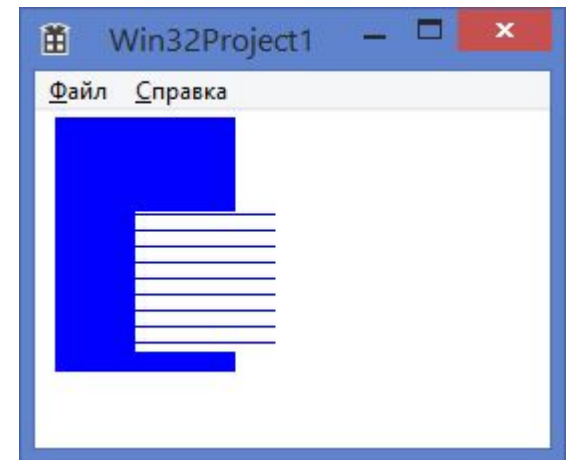
```
EndPaint(hWnd, &ps);
```

```
}
```



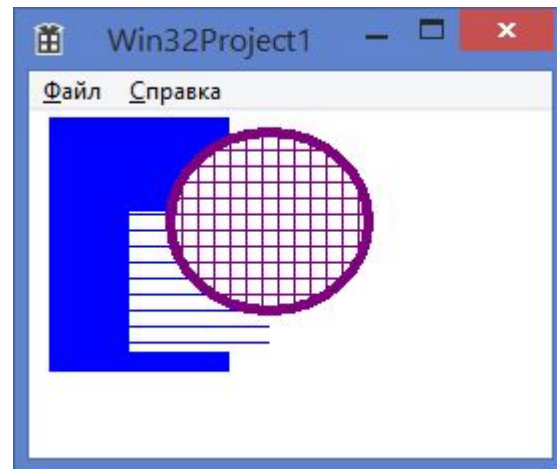
# Кисть (2)

```
hBrush = CreateHatchBrush(HS_HORIZONTAL, RGB(0, 0, 255));  
SelectObject(hdc, hBrush);  
RECT rect2 = { 50, 50, 120, 120 };  
FillRect(hdc, &rect2, hBrush);
```



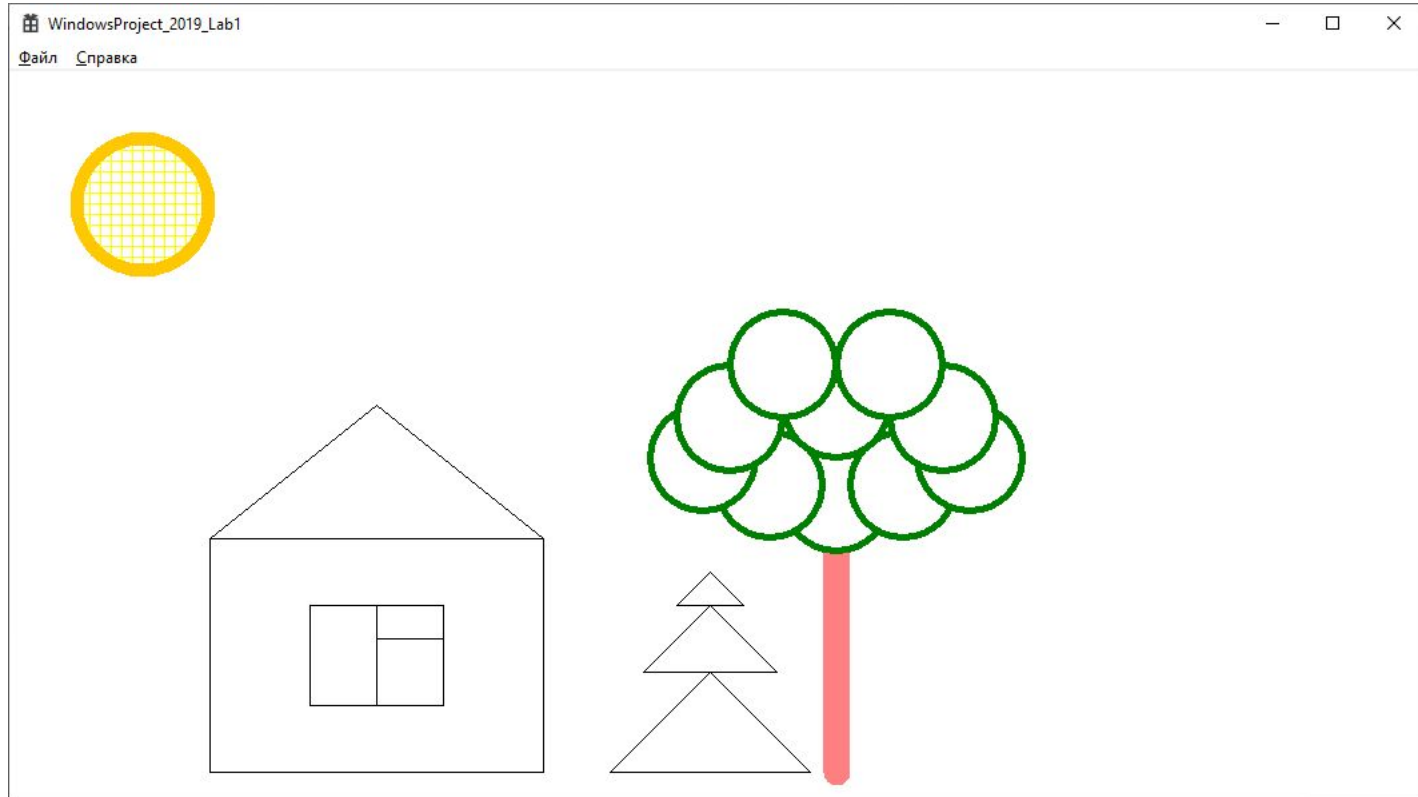
# Кисть (3)

```
hBrush = CreateHatchBrush(HS_CROSS, RGB(128, 0, 128));  
SelectObject(hdc, hBrush);  
HPEN hPen;  
hPen = CreatePen(PS_SOLID, 5, RGB(128, 0, 128));  
SelectObject(hdc, hPen);  
Ellipse( hdc, 70, 10, 170, 100);  
  
DeleteObject(hBrush);
```



# Задача 4

Добавьте солнышко





# Задача 4: Решение

Моё решение. Ваше может быть сильно другим

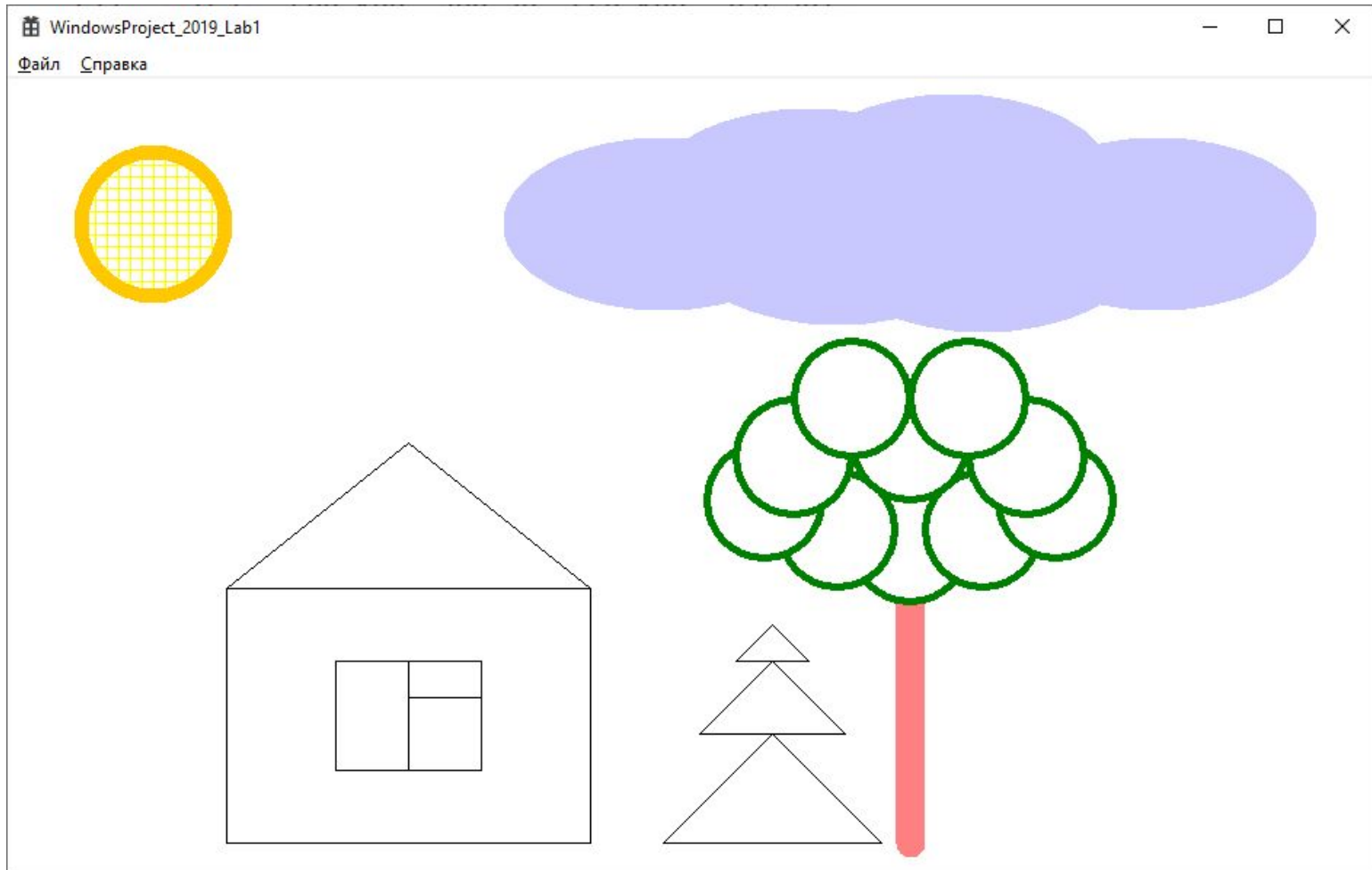
```
// Солнышко
HBRUSH hBrush;
hBrush = CreateHatchBrush(HS_CROSS, RGB(255, 255, 0));
SelectObject(hdc, hBrush);

hPen = CreatePen(PS_SOLID, 10, RGB(255, 200, 0));
SelectObject(hdc, hPen);

Ellipse(hdc, 50, 50, 150, 150);
```

# Задача 5

Добавьте облако



# Задача 6

Добавьте то, что считаете необходимым  
добавить!

# Задача 7

Раскрасьте и украсьте дом!

# Домашнее задание – часть 1

Дорисовать рисунок из классной работы, добавив к нему подходящее сооружение или механизм:

- Антенна
- Автомобиль
- Велосипед
- Гараж
- Поезд
- Самолет
- Яхта
- Любое другое !

Добавленное сооружение/механизм должен состоять не менее чем из 15 отрезков, эллипсов или других элементов.

Оно должно быть цветным и содержать заливку.

# Домашнее задание – часть 2

Надо выбрать любой логотип автомобиля (можно из вариантов ниже) и нарисовать его контуром схематично.



# Домашнее задание – часть 3 (по желанию)

Удивите меня сложным и красивым рисунком!

**ВАЖНО** - нельзя использовать циклы, развилки, генератор случайных чисел, вызовы собственных функций, в том числе нельзя использовать рекурсию!

Это именно сложный и красивый рисунок, а не красивая математическая формула!



# Домашнее задание - оформление

Для сдачи домашней работы нужно подготовить отчет.

Отчет состоит из:

- 1) Стандартного титульного листа
- 2) Задания на лабораторную работу (отдельная страница)
- 3) Полной распечатки кода программы
- 4) Скриншот работающей программы
- 5) Листы с расчетами всех рисунков (листы могут быть весьма неряшливо оформлены – но они должны доказывать, что вы самостоятельно рассчитывали параметры рисунков)
- 6) Заключение – выводы, чему научился

# Домашнее задание – ВАЖНО

Все работы должны быть уникальны и выполнены лично.

Авторство подтверждается вручную отрисованными листами с расчетами координат точек и элементов.

Если будут замечены одинаковые работы – автором будет считаться первый сдавший.

Второй и следующие – должны выполнить тотально другой вариант.