

Инструкция. Установка Visual Studio и SFML

Для того чтобы использовать библиотеку SFML, необходимо скачать и установить на свой компьютер Visual Studio и саму библиотеку. Сначала установим SFML.

- Перейди по ссылке <https://www.sfml-dev.org/download/sfml/2.6.1/>:

Download SFML 2.6.1

On Windows, choosing 32 or 64-bit libraries should be based on which platform you want to compile for, not which OS you have. Indeed, you can perfectly compile and run a 32-bit program on a 64-bit Windows. So you'll most likely want to target 32-bit platforms, to have the largest possible audience. Choose 64-bit packages only if you have good reasons.

Unless you are using a newer version of Visual Studio, the compiler versions have to match 100%!
Here are links to the specific MinGW compiler versions used to build the provided packages:
WinLibs MSVCRT 13.1.0 (32-bit), WinLibs MSVCRT 13.1.0 (64-bit)

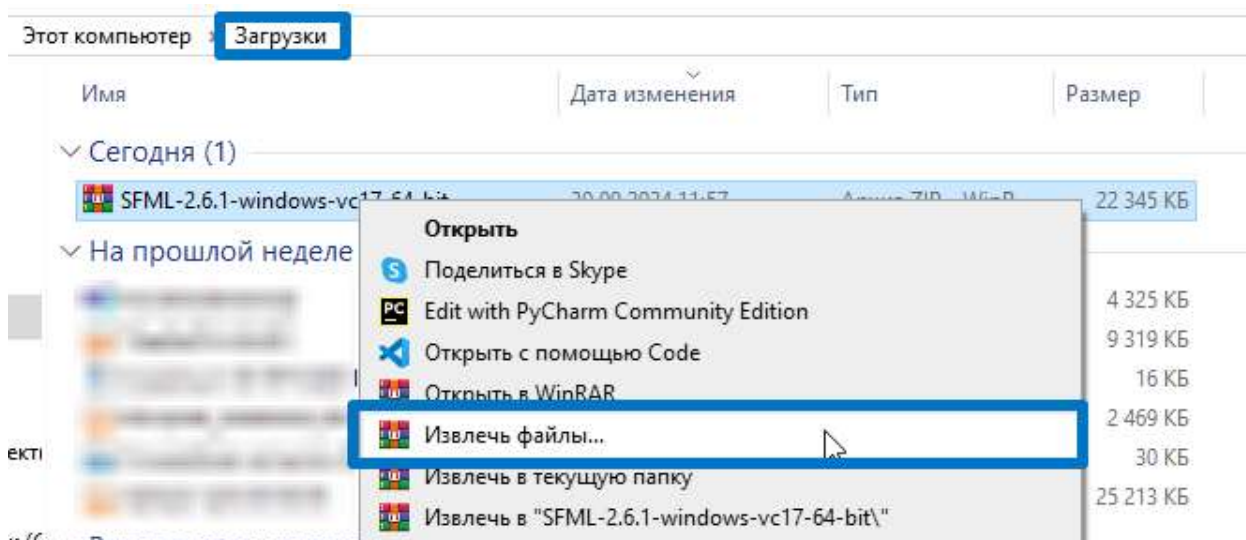
Visual C++ 17 (2022) - 32-bit	Download 20.3 MB	Visual C++ 17 (2022) - 64-bit	Download 21.8 MB
Visual C++ 16 (2019) - 32-bit	Download 19.3 MB	Visual C++ 16 (2019) - 64-bit	Download 20.7 MB
Visual C++ 15 (2017) - 32-bit	Download 17.6 MB	Visual C++ 15 (2017) - 64-bit	Download 19.4 MB
GCC 13.1.0 MinGW (DW2) - 32-bit	Download 17.9 MB	GCC 13.1.0 MinGW (SEH) - 64-bit	Download 18.9 MB

- Нажми кнопку **Download** напротив свежей версии (2022) для системы, под которую ты будешь вести разработку (у нас это будет 64-bit):

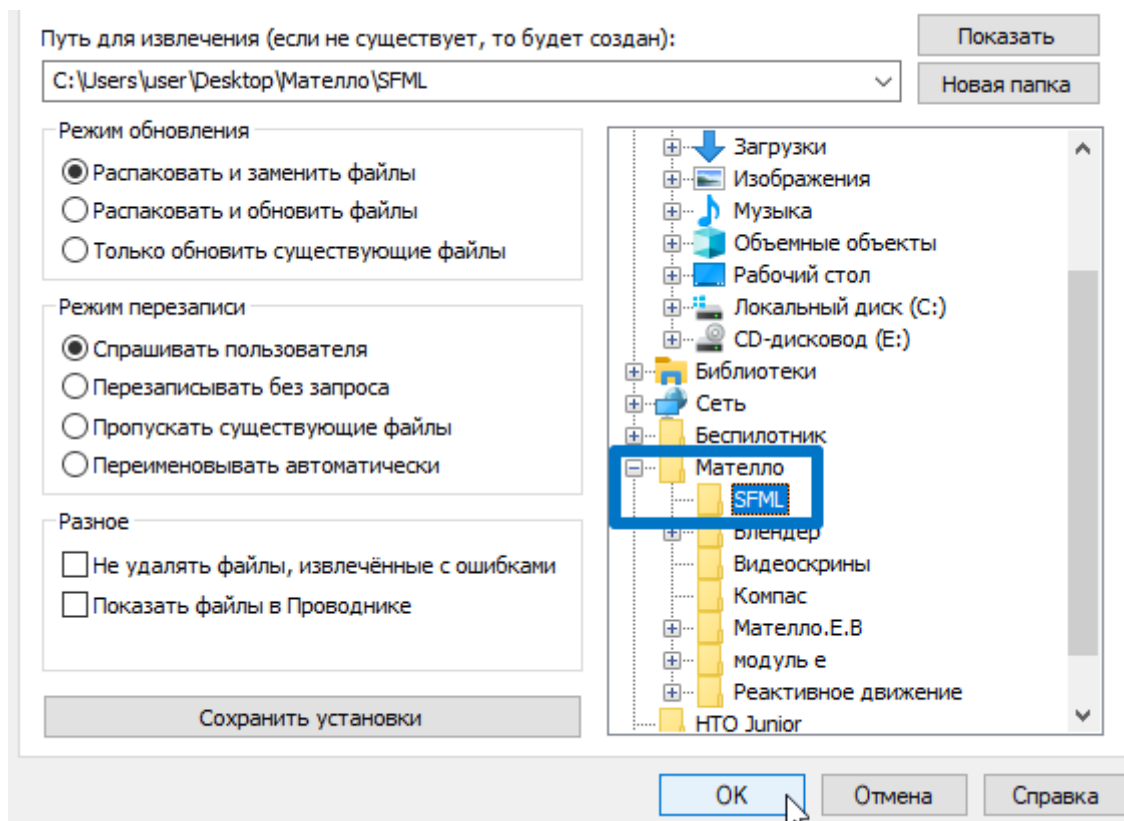
Visual C++ 17 (2022) - 64-bit	Download 21.8 MB
Visual C++ 16 (2019) - 64-bit	Download 20.7 MB
Visual C++ 15 (2017) - 64-bit	Download 19.4 MB
GCC 13.1.0 MinGW (SEH) - 64-bit	Download 18.9 MB

Библиотека скачается в **Загрузки** твоего компьютера в формате **zip**, архивом.

- Зайди в **Загрузки**, нажми на скачанный архив правой кнопкой и выбери строку **Извлечь файлы...**:



- Выбери папку, в которую будешь извлекать архив, и нажми **ОК**:

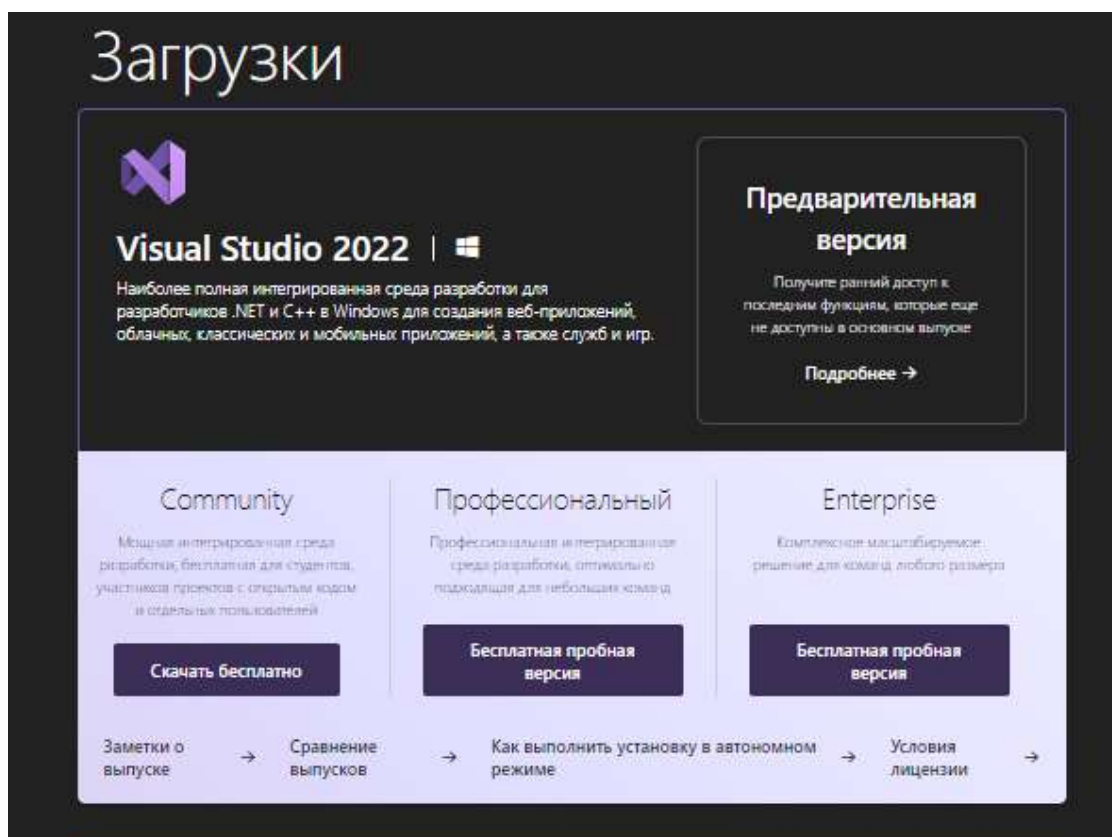


- Проверь папку. Там должна появиться новая папка с файлами библиотеки **SFML-2.6.1**:

Этот компьютер > Рабочий стол > Мателло > SFML >				
Имя	Дата изменения	Тип	Размер	
SFML-2.6.1	04.11.2023 5:56	Папка с файлами		

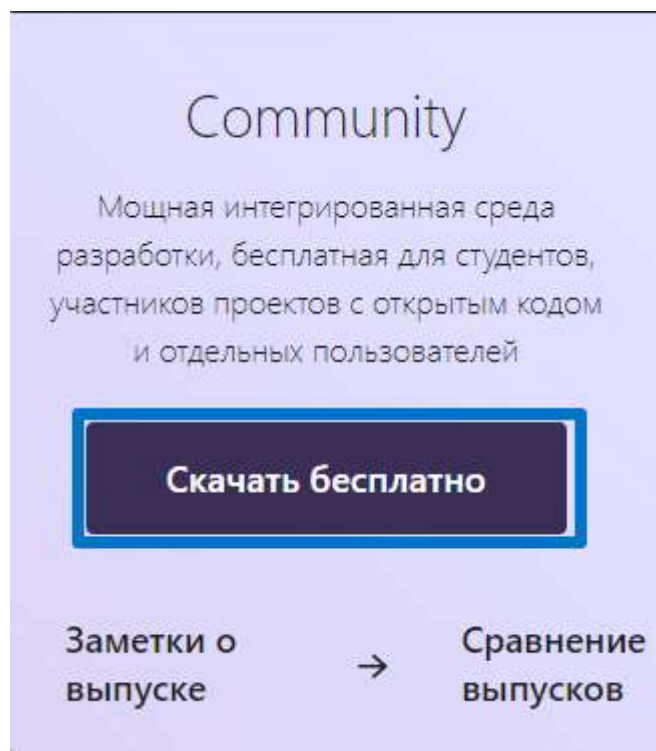
После установки библиотеки займёмся установкой **Visual Studio**.

- Перейди по ссылке <https://visualstudio.microsoft.com/ru/downloads/>:

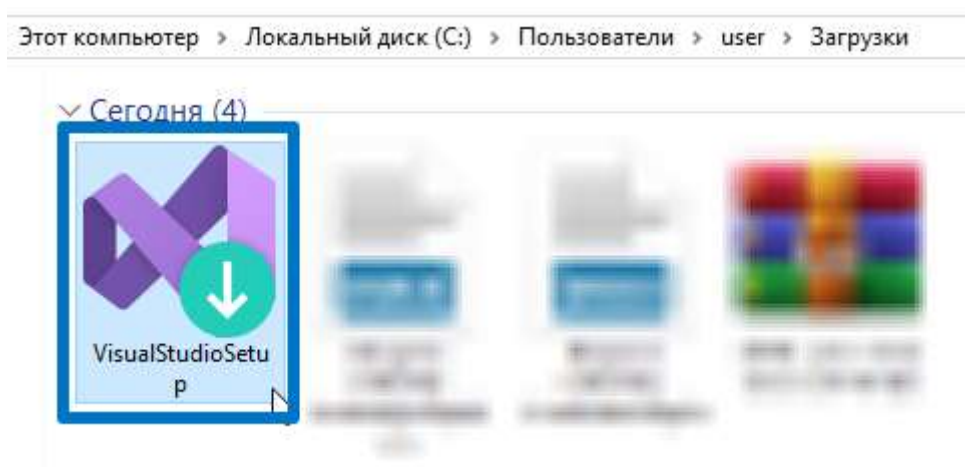


Нам подойдёт версия **Community**. На начальном этапе её будет достаточно, а также эта версия всегда остаётся бесплатной. Также версия Visual Studio и версия SFML должны быть одинаковыми — в нашем случае это версия 22.

- Нажми **Скачать бесплатно**:



- Открой скачанный файл из загрузок двойным щелчком мыши:



После установки появятся **настройки**, в которых можно выбрать возможности установленного **Visual Studio**. Выбери **все проекты с C++**:

Изменение — Visual Studio Community 2022 — 17.11.4

Рабочие нагрузки

Отдельные компоненты

Языковые пакеты

Расположения установки



Разработка классических приложений на C++
Создавайте современные приложения C++ для Windows с использованием любых удобных инструме...

☒



Разработка мобильных приложений на языке C++
Создавайте кросс-платформенные приложения для iOS, Android или Windows на языке C++.

☒



Разработка Windows-приложений
Создавайте приложения для платформы Windows, используя WinUI с C# или, при необходимости, C++.

☒



Разработка игр с помощью Unity
В Unity, мощной кросс-платформенной среде разработки, можно создавать игры в формате 2D и 3D.

☐



Разработка игр на C++
Все возможности C++ для создания высококлассных игр на платформе DirectX, Unreal или Cocos2d.

☒

Игры (2)

Если настройки не открылись, открой их самостоятельно на **главном экране** (1) и нажми **Запустить** (2):

Все установки обновлены



Visual Studio Community 2022
17.11.4
Мощная интегрированная среда разработки, бесплатная для студентов, участников проектов с открытым кодом и отдельных пользователей.
[Заметки о выпуске](#)

1

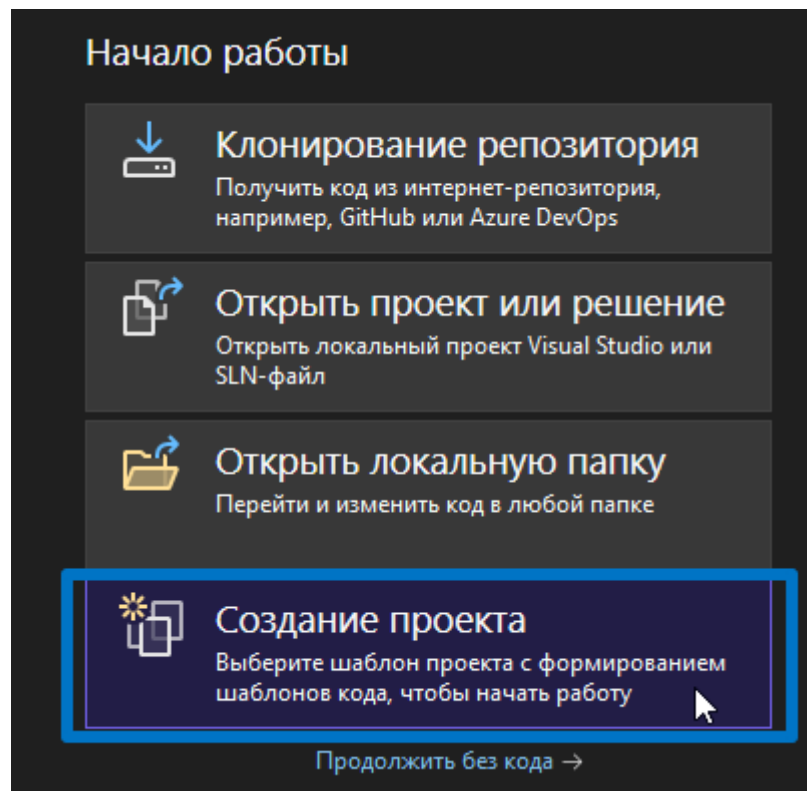
2

Изменить

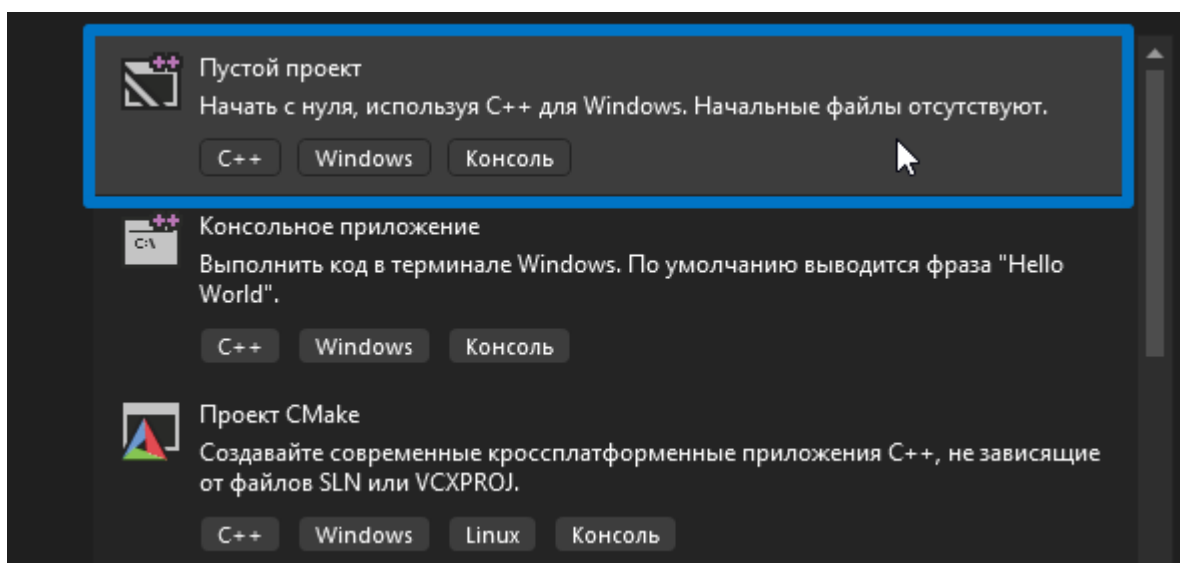
Запустить

Дополнительно ▼

- Нажми **Создание проекта**:



- Выбери шаблон **Пустой проект** и нажми **Далее** в правом нижнем углу:



- Введи **имя проекта** и нажми **Создать** в правом нижнем углу. Поставь **галочку** рядом со строчкой **Поместить решение и проект в одном каталоге**:

Настроить новый проект

Пустой проект C++ Windows Консоль

Имя проекта

test-SFML

Расположение

C:\Users\user\source\repos

Имя решения ⓘ

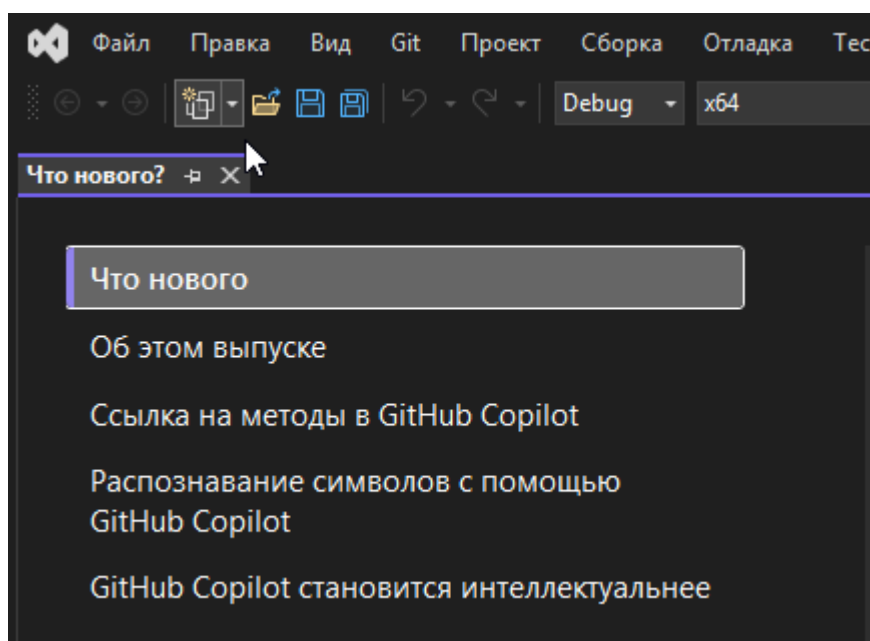
test-SFML

☒ Поместить решение и проект в одном каталоге

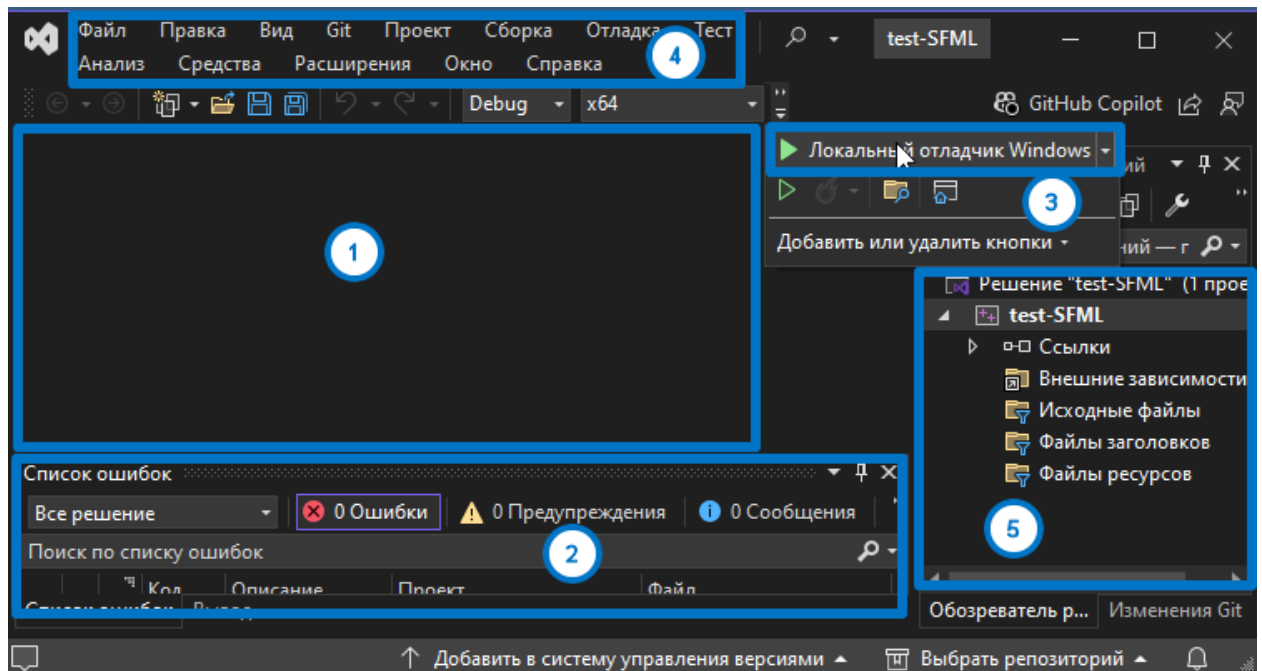
Проект будет создан в "C:\Users\user\source\repos\test-SFML\"

В результате у тебя откроется новый проект.

Если у тебя появилась такая вкладка, просто закрой её:



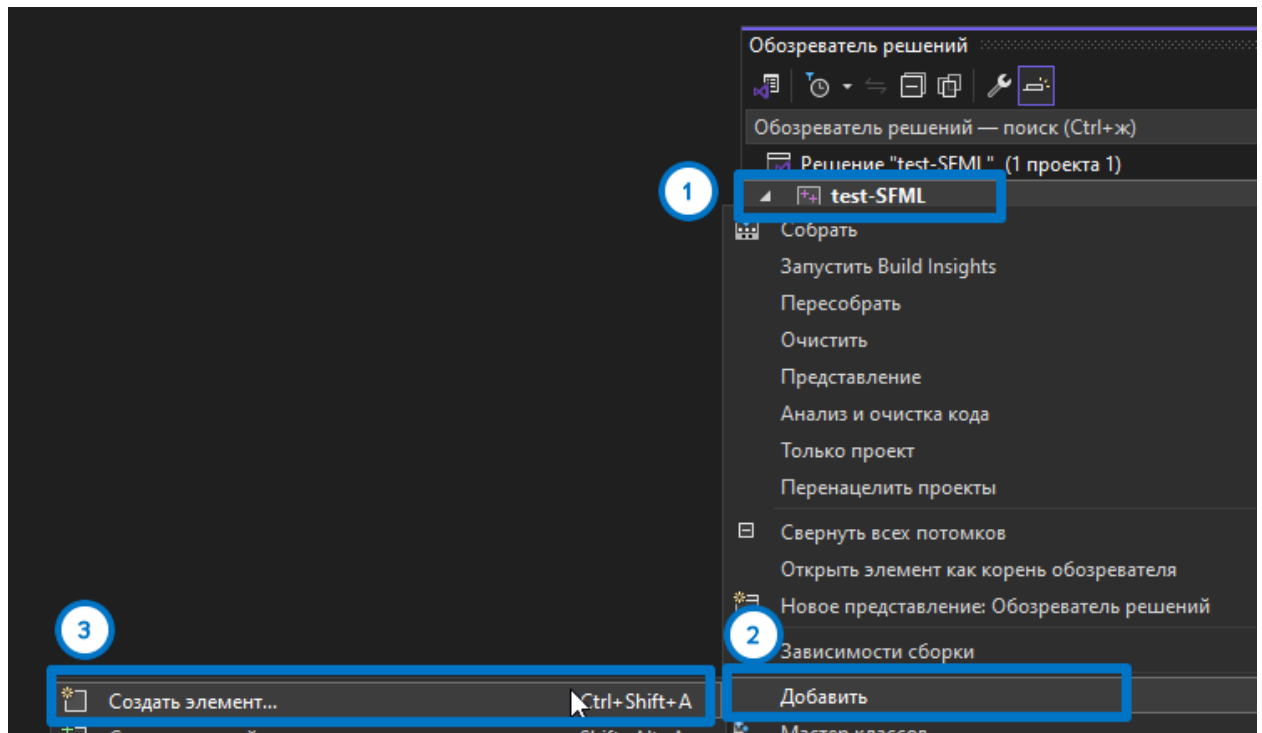
Давай немного познакомимся с интерфейсом программы:



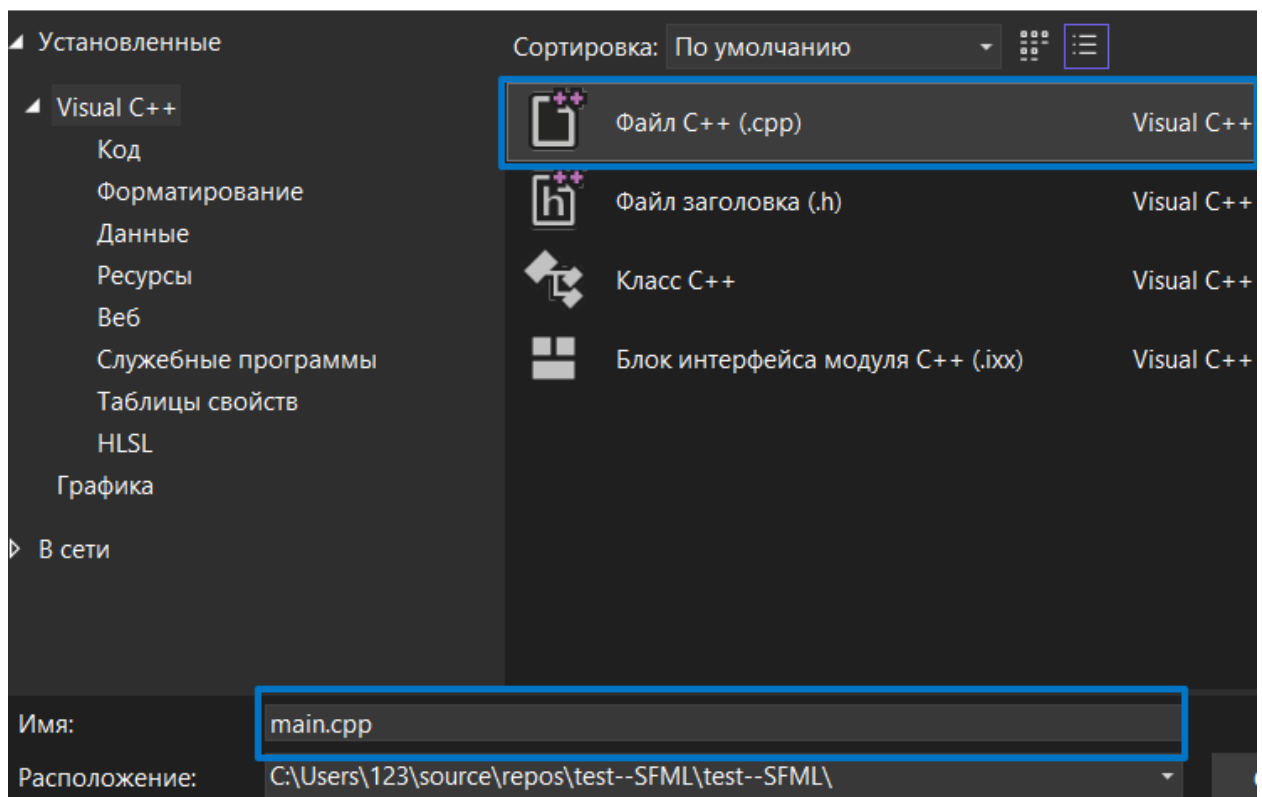
1. Окно редактора кода.
2. Консоль.
3. Кнопка отладки и запуска кода.
4. Меню.
5. Дерево проекта.

Теперь нужно настроить этот проект под использование библиотеки SFML.

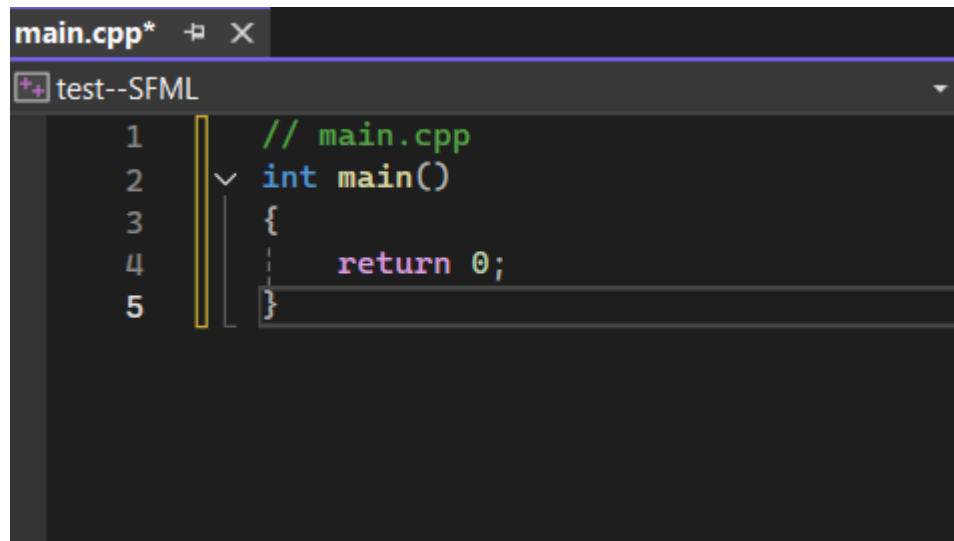
- Добавь новый файл. Для этого нажми правой кнопкой на название проекта в **дерева справа** (1), выбери строку **Добавить** (2), нажми **Создать элемент...** (3):



- Назови новый элемент **main.cpp** и нажми на кнопку **Добавить**. Это файл с главной функцией проекта, она есть в любом проекте подобного типа:



- Добавь в новый файл **пустую функцию main**. Она нужна для дальнейших настроек работы:

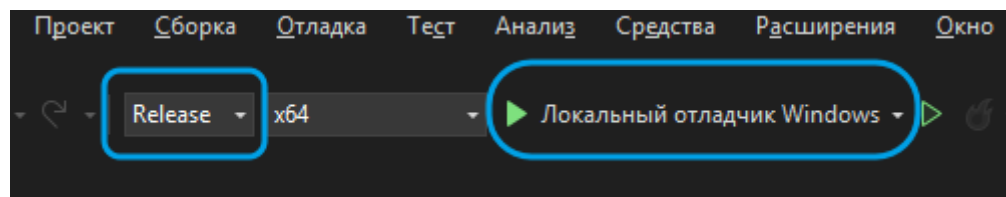
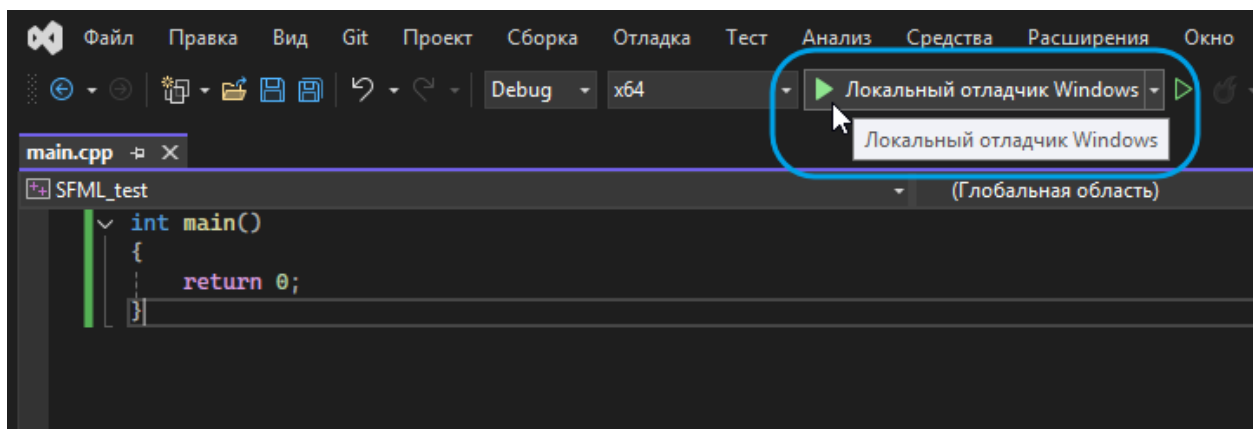


```

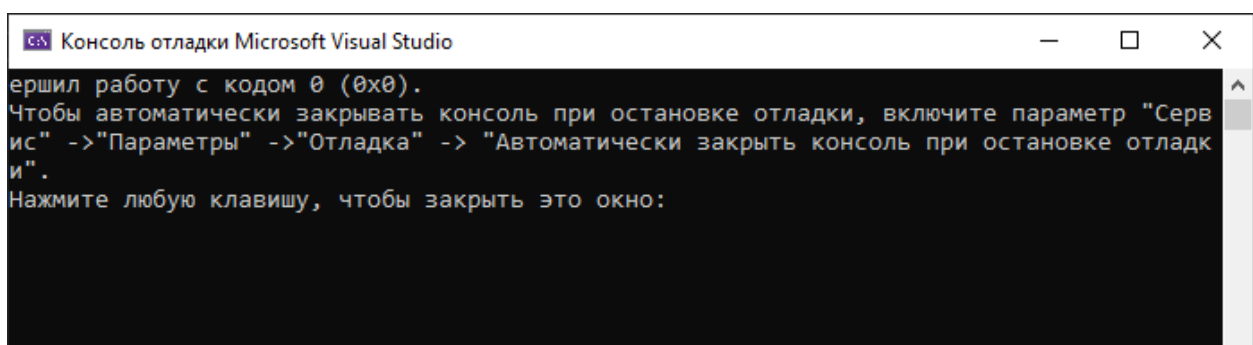
1 // main.cpp
2 int main()
3 {
4     return 0;
5 }

```

- Обязательно запусти компиляцию проекта (чтобы в папке с проектом у нас появились все необходимые для дальнейшей работы файлы) в режиме **Debug** и в режиме **Release**:



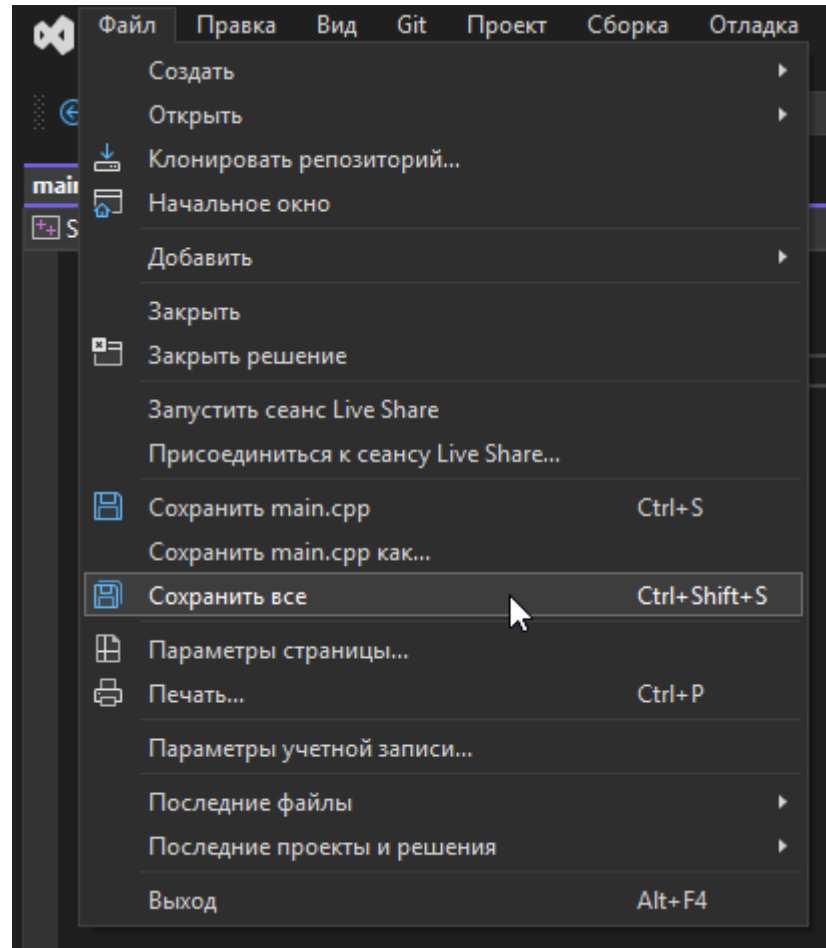
- Если у тебя появилось окно консоли отладки, то просто закрой его (сейчас оно нам не нужно):



Во время разработки проекта мы будем пользоваться режимом **Debug**.

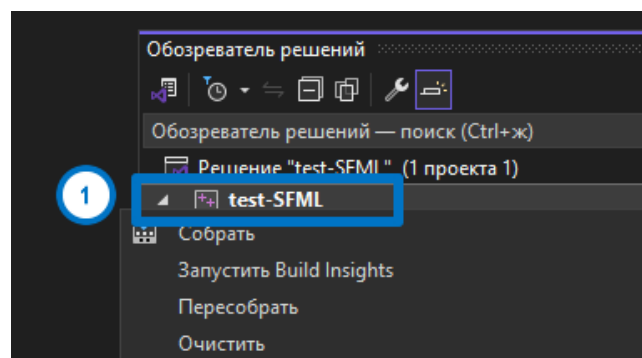
Режим **Debug** обычно используется в качестве конфигурации по умолчанию. Конфигурация **Release** используется во время финальной сборки программы для её дальнейшего выпуска.

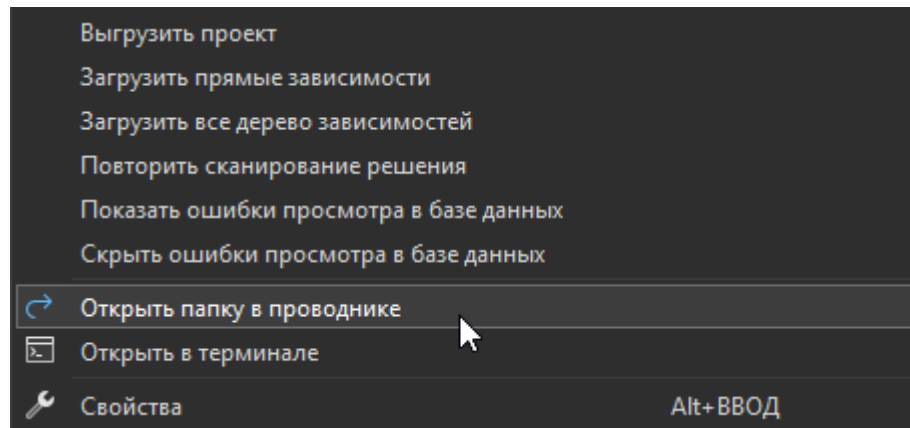
- Нажми в верхнем меню команду **Файл** → **Сохранить все** (или сочетание клавиш **Ctrl+S**):



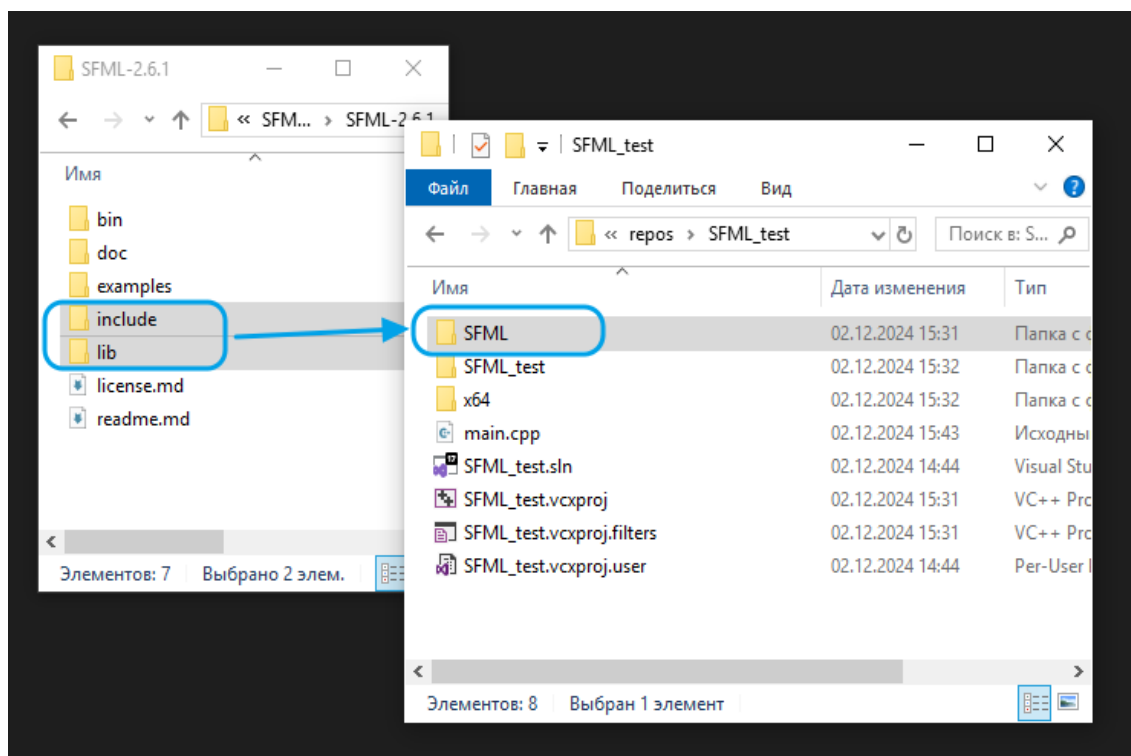
Сейчас давай переместим все необходимые файлы библиотеки SFML прямо в наш проект.

- Открой папку с созданным в Visual Studio проектом:
 - Чтобы это сделать, в древе проекта правой кнопкой мыши кликни по названию проекта и в выпадающем меню выбери пункт **Открыть папку в проводнике** (так ты всегда сможешь посмотреть, где на компьютере хранится проект):

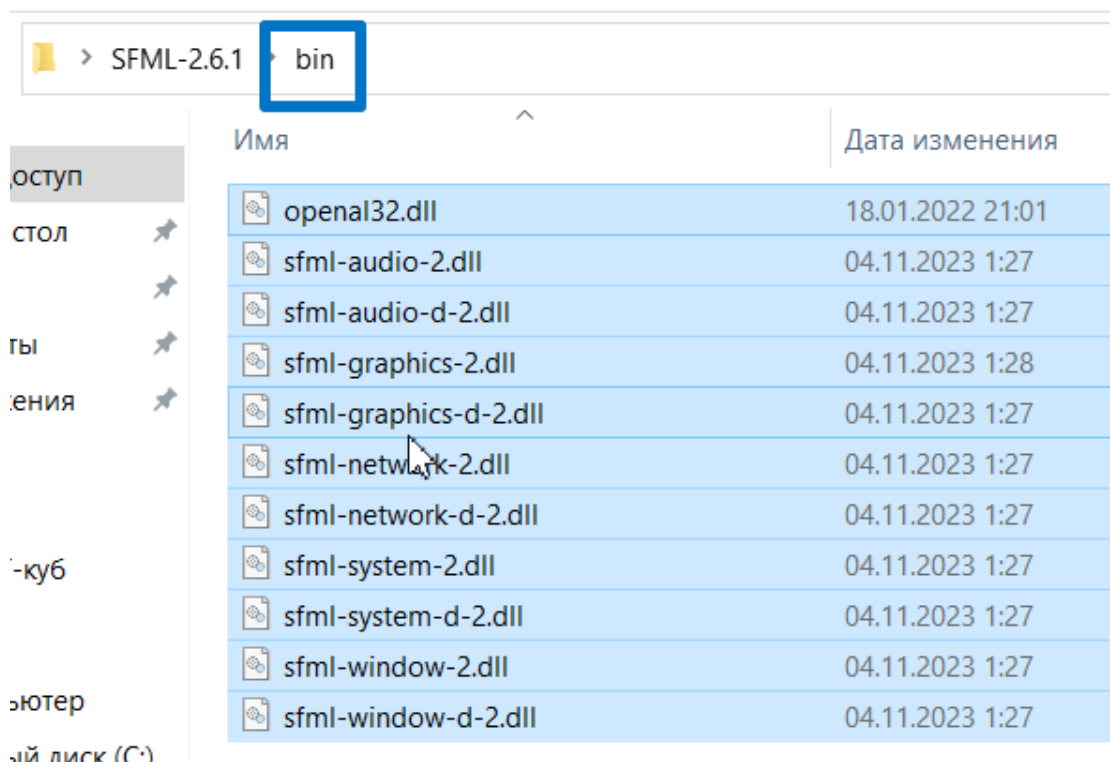




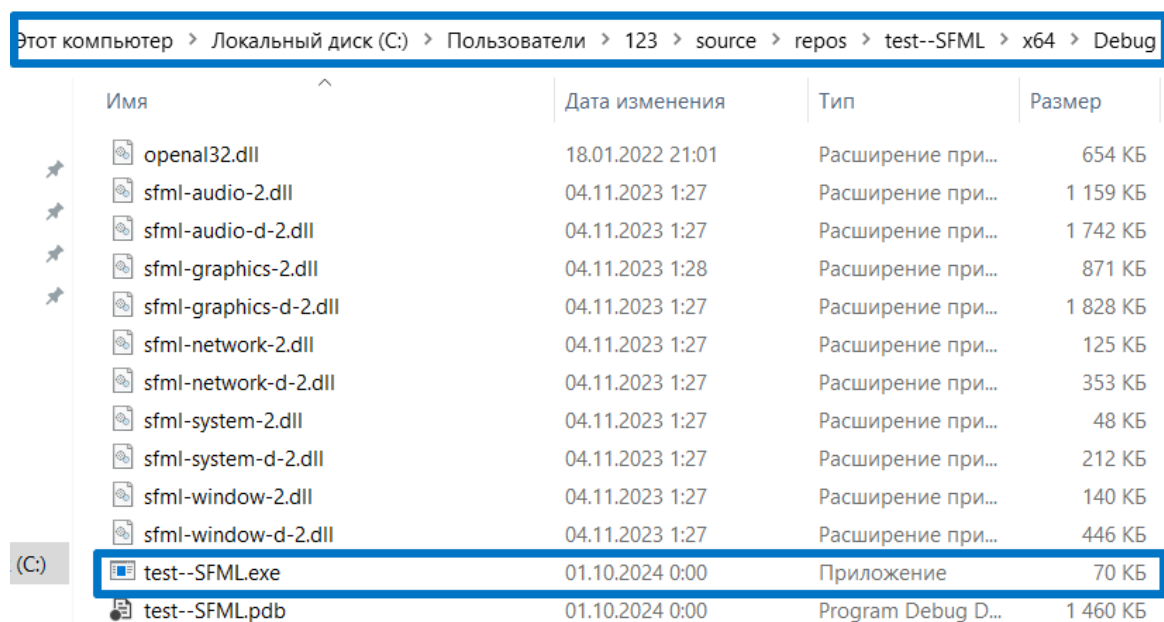
- В папке с проектом Visual Studio создай папку с именем **SFML**.
- Из папки с библиотекой **SFML-2.6.1** скопируй целиком папки **include**, **lib** и вставь их в папку **SFML** в папке проекта:



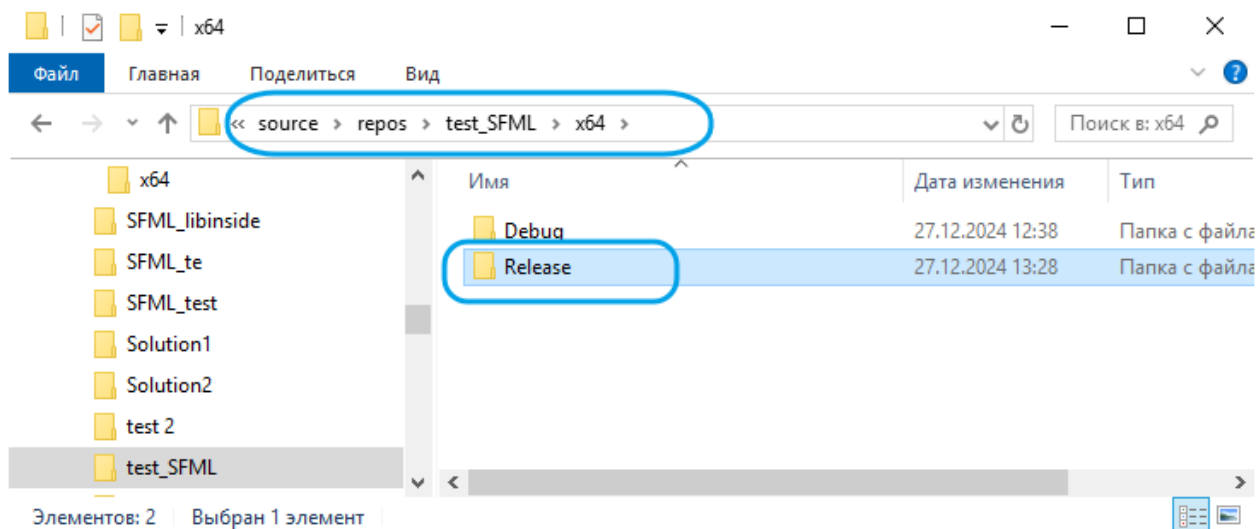
- Далее в папке с библиотекой **SFML-2.6.1** открой папку **bin** и скопируй из неё все файлы:



- Вставь их в папку **Debug** с файлом своего **проекта в формате .exe**. Если ты не менял(а) ничего в настройках, эта папка расположена примерно по такому пути:



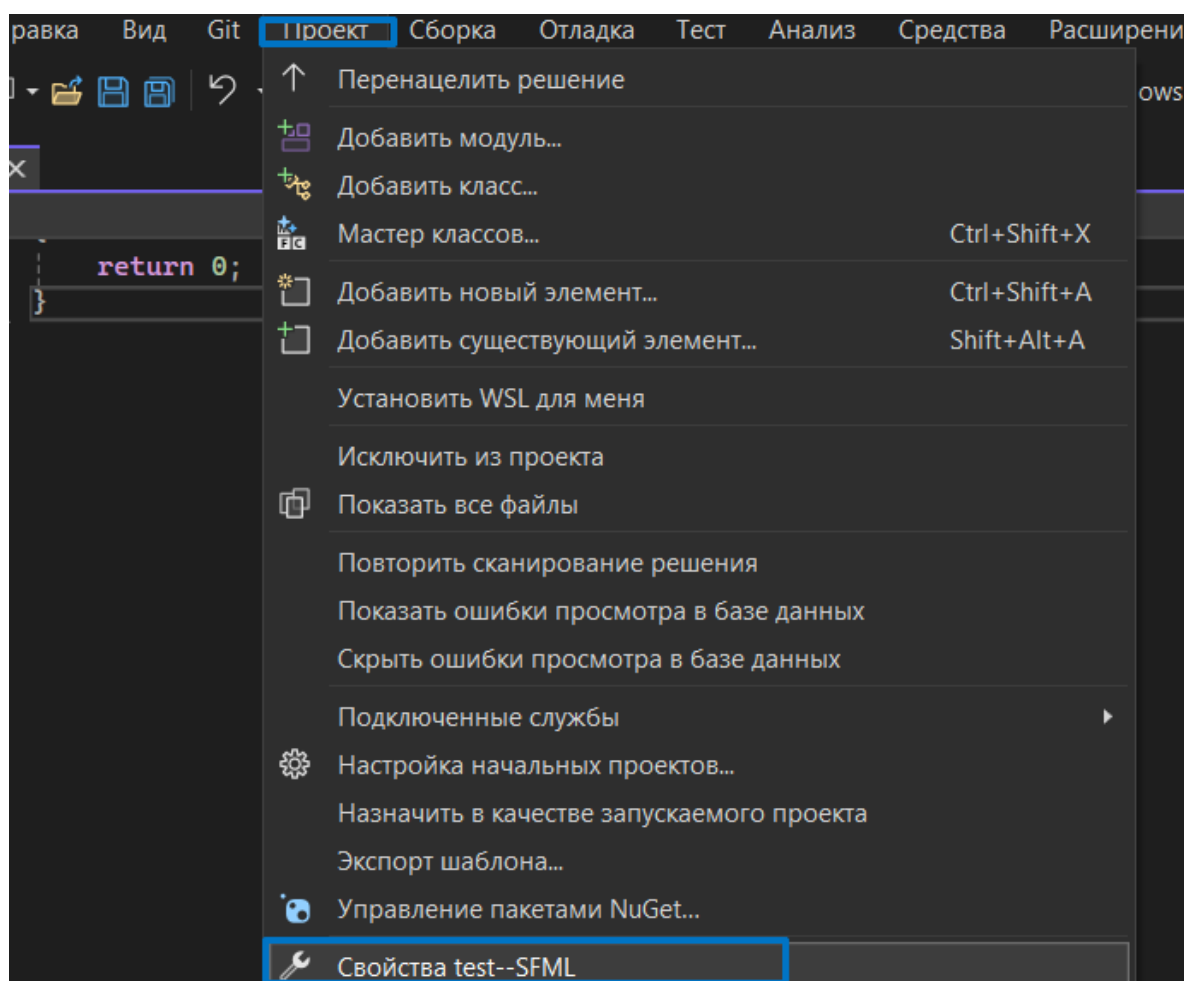
- А также в папку **Release** с файлом своего **проекта в формате .exe**. Если ты не менял(а) ничего в настройках, эта папка расположена примерно по такому пути:



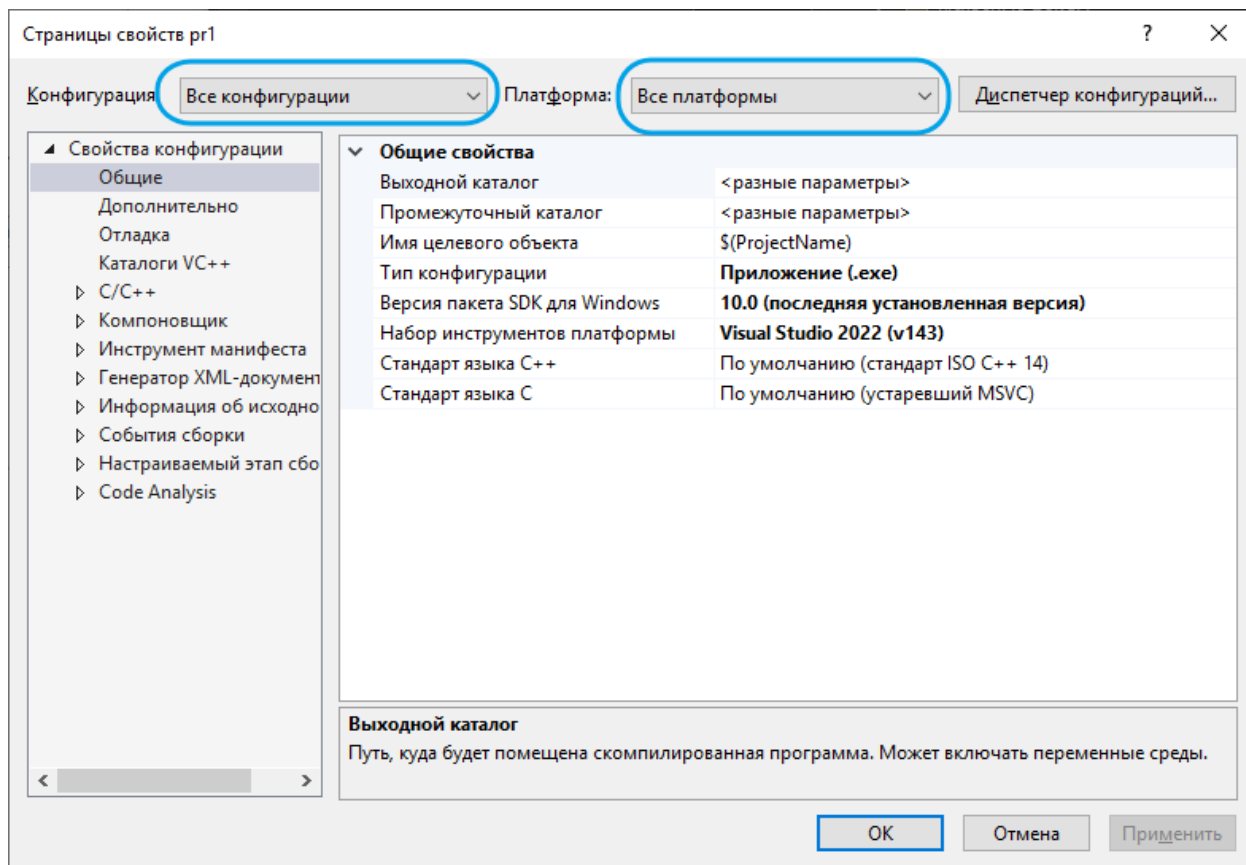
- Папку **SFML-2.6.1** теперь можно закрыть, она нам больше не нужна, ведь все файлы мы перенесли в папку с проектом.

Теперь перейдём к настройкам **взаимодействия проекта с библиотекой SFML**.

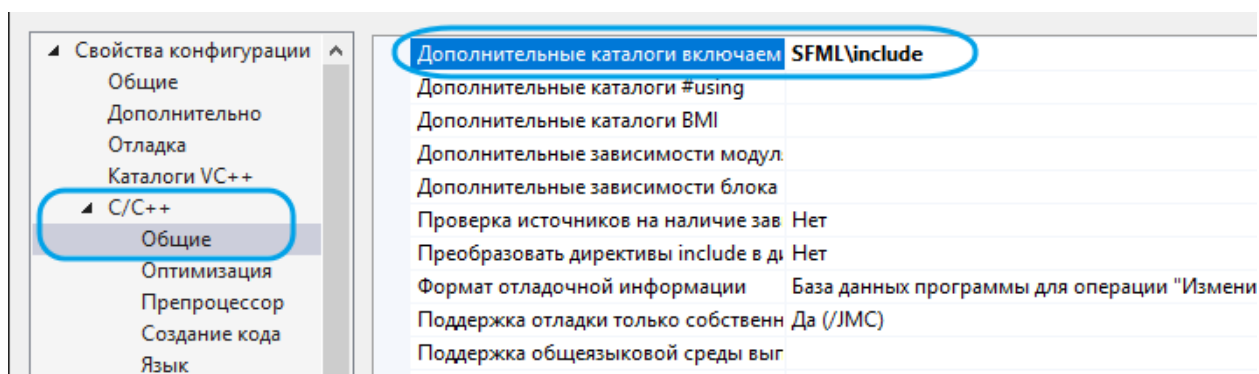
- В Visual Studio зайти в **свойства своего проекта**, которые находятся в разделе **Проект**:



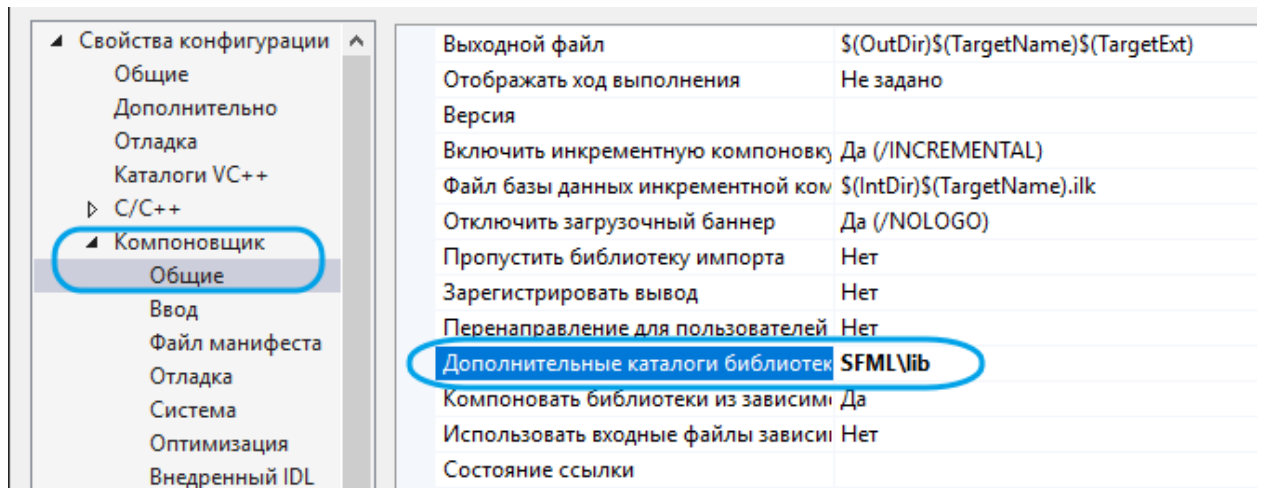
- Сверху в выпадающих меню выбери **Все конфигурации** и **Все платформы**. Это означает, что сделанные сейчас нами изменения будут распространяться как на **Debug**-версию, так и на **Release**:



- Укажи относительный путь к папке **include** (то есть путь к папке относительно файла **main.cpp**). Для этого добавь путь **SFML\include** в первую строку раздела свойств **C++**, **Общие** и нажми клавишу **Enter**:



- Таким же образом укажи путь к папке **lib** и вставь его в **Дополнительные каталоги библиотек** в строке **Компоновщик**, **Общие**:



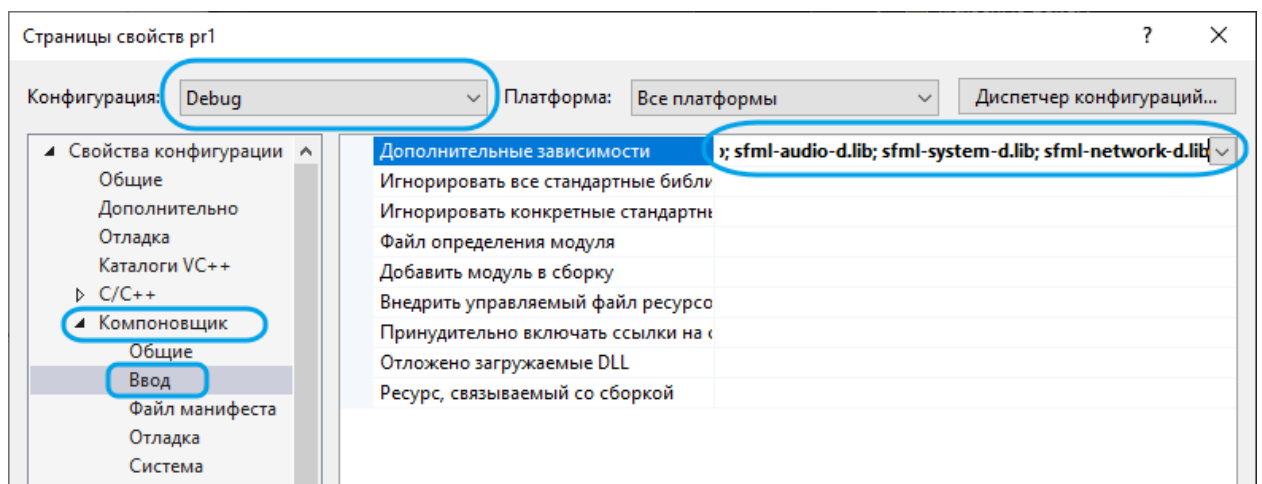
- Нажми на кнопку **Применить**.

Теперь мы укажем, какие библиотеки будут использоваться для работы в режиме отладки **Debug**.

- Сверху в выпадающем списке укажи конфигурацию **Debug**.
- В разделе **Компоновщик**, **Ввод** в строку **Дополнительные зависимости** вставь используемые файлы библиотеки (они размещены в папке **lib** и имеют такое же расширение) через **знак ;**. Удалять тот текст, который там уже есть, не нужно, просто вставь новый с начала строки.

В нашем проекте тебе понадобятся всего пять библиотек:

sfml-graphics-d.lib; sfml-window-d.lib; sfml-audio-d.lib; sfml-system-d.lib; sfml-network-d.lib

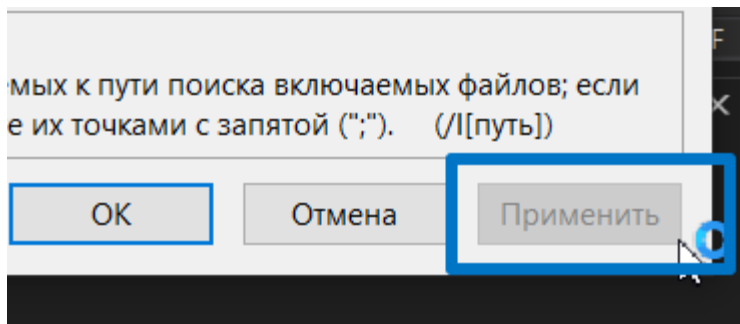


- Нажми **Применить**.
- Теперь в разделе **Конфигурации** выбери **Release**.
- В разделе **Компоновщик**, **Ввод**, в строке **Дополнительные зависимости** укажи те же самые библиотеки, но без приписки **-d**.

Вот список библиотек:

sfml-graphics.lib; sfml-window.lib; sfml-audio.lib; sfml-system.lib; sfml-network.lib

- Нажми **Применить**, а затем **ОК**:



Осталось загрузить тестовый код и посмотреть, работает ли всё как надо.

- В файле **main.cpp** вместо функции **main** загрузи тестовый код:

```
// main.cpp

#include <SFML/Graphics.hpp>

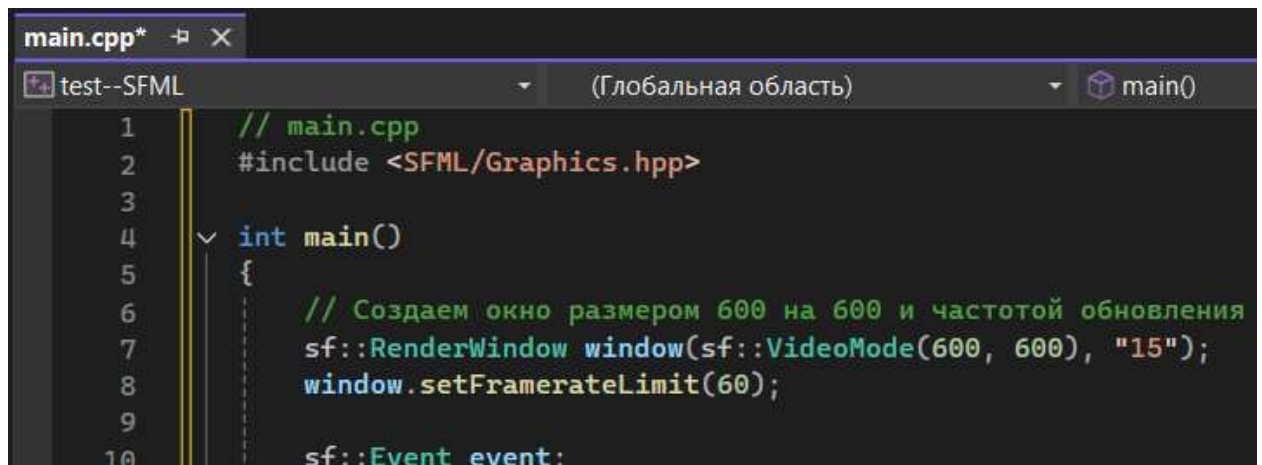
int main()
{
    sf::RenderWindow window(sf::VideoMode(600, 600), " test-SFML");
    window.setFramerateLimit(60); sf::Event event;

    while (window.isOpen())
    {
        while (window.pollEvent(event))
        {
            if (event.type == sf::Event::Closed) window.close();
            if (event.type == sf::Event::KeyPressed)
            {
                if (event.key.code == sf::Keyboard::Escape) window.close();
            }
        }

        window.clear();

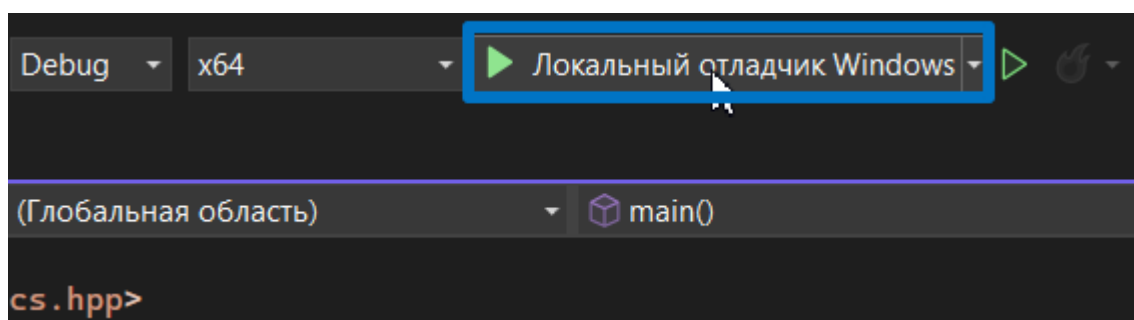
        window.display();
    }
}
```

```
}  
  
return 0;  
  
}
```



```
main.cpp*  test--SFML  (Глобальная область)  main()  
1  // main.cpp  
2  #include <SFML/Graphics.hpp>  
3  
4  int main()  
5  {  
6      // Создаем окно размером 600 на 600 и частотой обновления  
7      sf::RenderWindow window(sf::VideoMode(600, 600), "15");  
8      window.setFramerateLimit(60);  
9  
10     sf::Event event;
```

- Запусти проект в режиме Debug, чтобы проверить ошибки в коде:



Если у тебя появится небольшое окошко с названием test-SFML, поздравляем, ты всё сделал верно. Если нет, попробуй снова:

