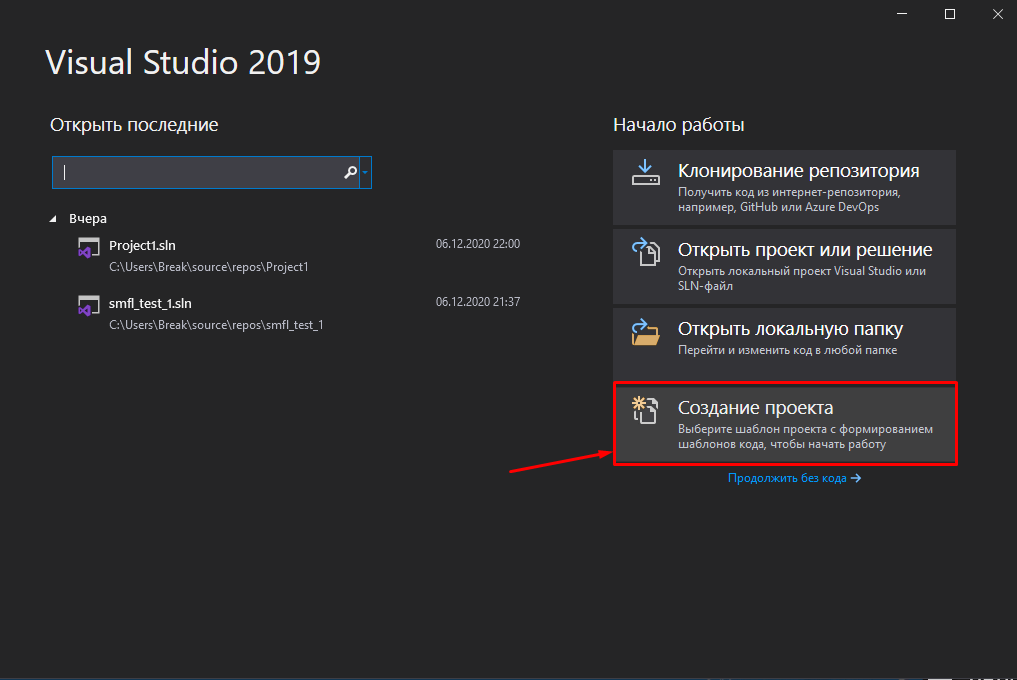
Инструкция по установке и настройке SFML. Запуск тестового кода

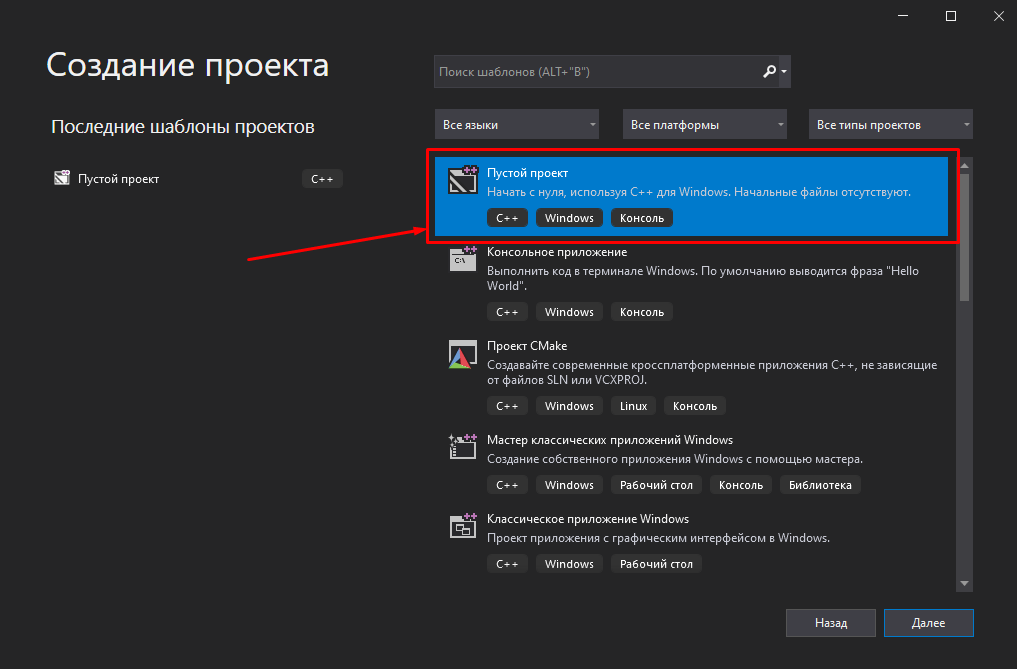
Создание проекта

Сегодня наша цель - настроить свое рабочее место, чтобы уже на следующем уроке начать разработку настоящей графической игры. Будем устанавливать библиотеку SFML, которая даст нам инструменты для создания игр: работа с окнами, мышью, клавиатурой, текстом, текстурами, временем и т.д.

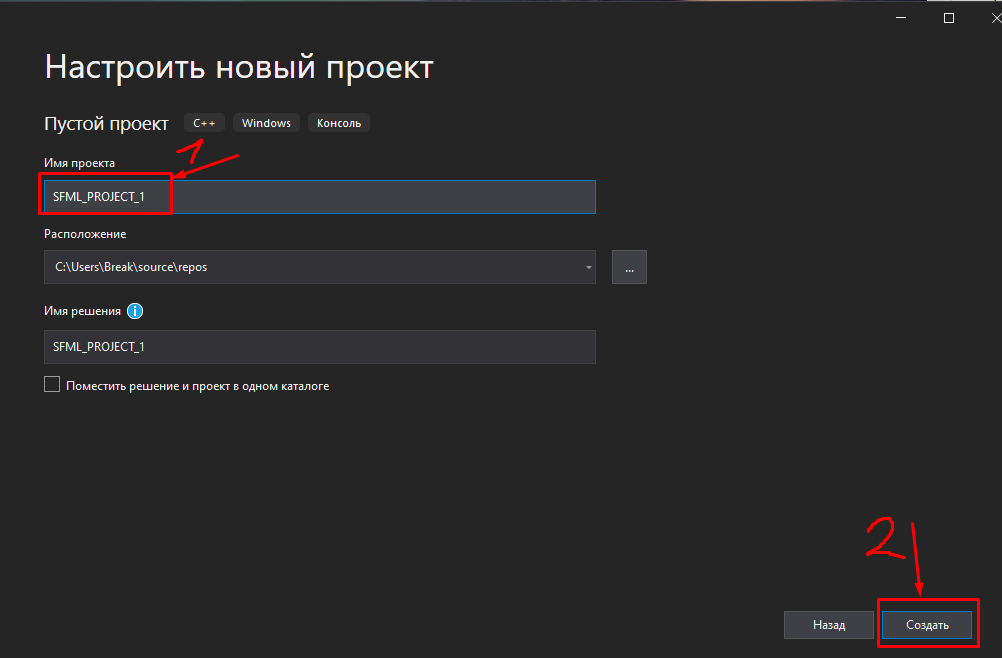
Заходим в Visual Studio и выбираем создание проекта.



Далее создаем пустой проект C++.



Называем своей проект и нажимаем создать.

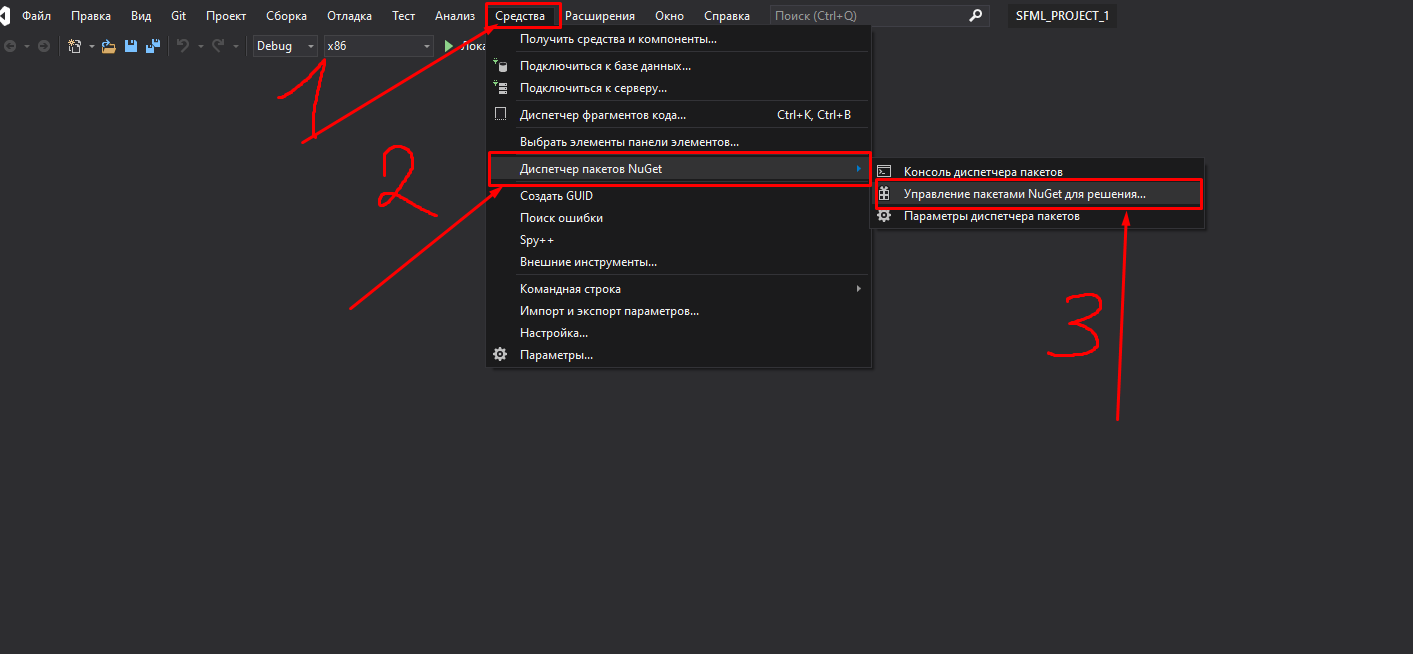


Установка библиотеки SFML

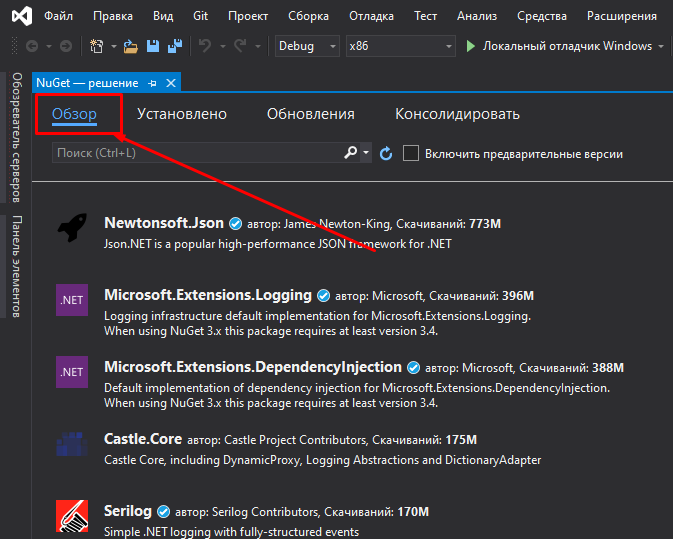
Установить SFML нам поможет NuGet Manager. Это инструмент, который помогает разработчику управлять сторонними библиотеками (пакетами) в своем проекте.

Под управлением подразумевается поиск, скачивание, установка, настройка, обновление и удаление файлов библиотек у себя в приложении. Воспользуемся им.

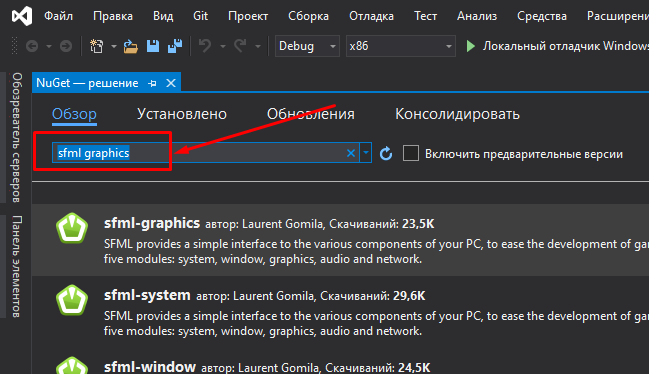
В открывшемся окне выбираем ->средства -> диспетчер пакетов NuGet -> управление пакетами NuGet для решения.



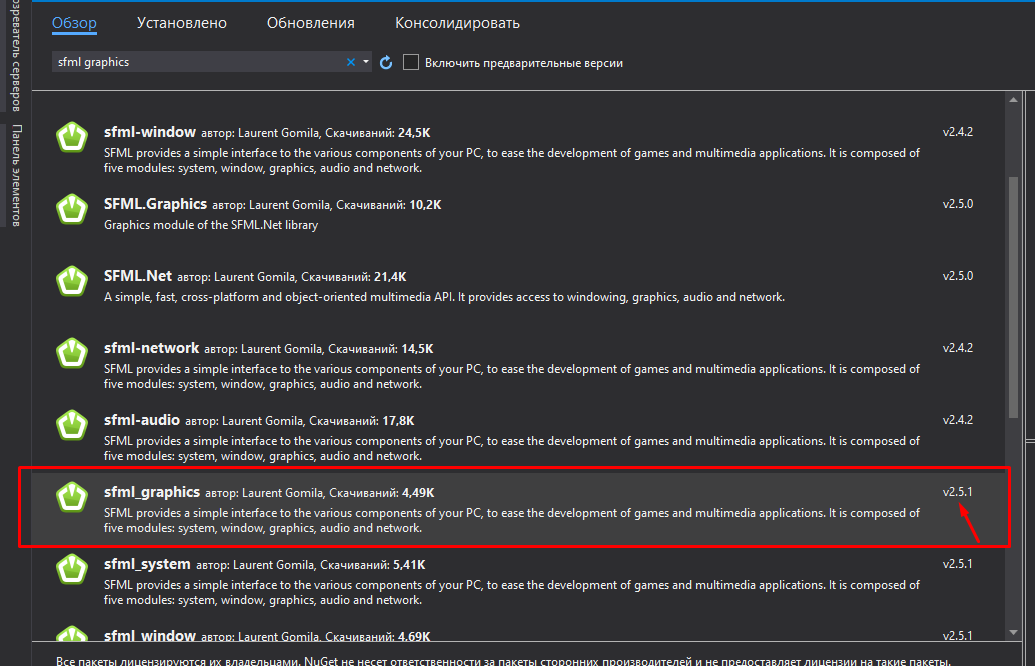
Далее переключаемся во вкладку Обзор.



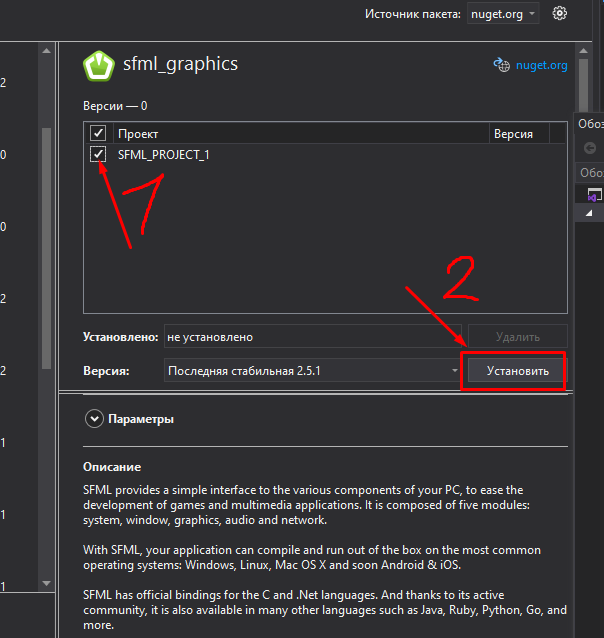
В строке поиска пишем - sfml graphics



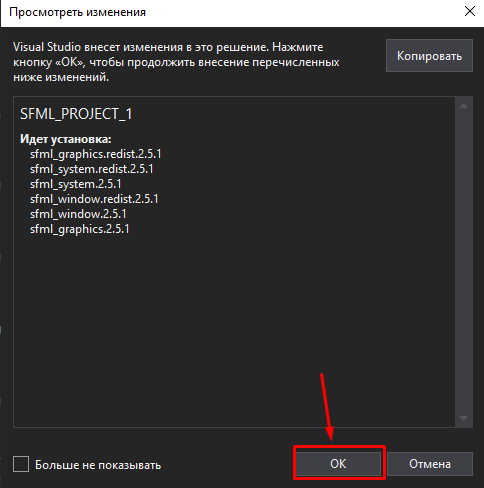
Находим версию 2.5.1



Затем ставим галочку и нажимаем “Установить”. Устанавливать будем графический пакет, он подтянет за собой все остальное.

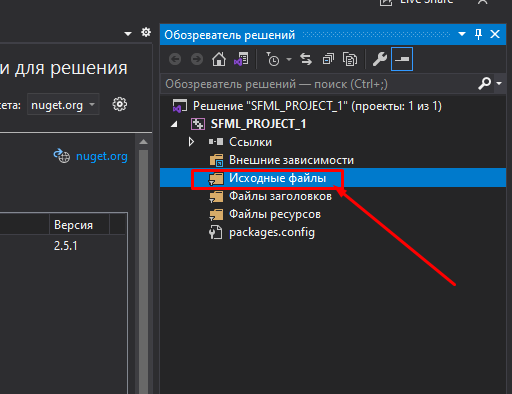


В открывшемся окне нажимаем ОК.

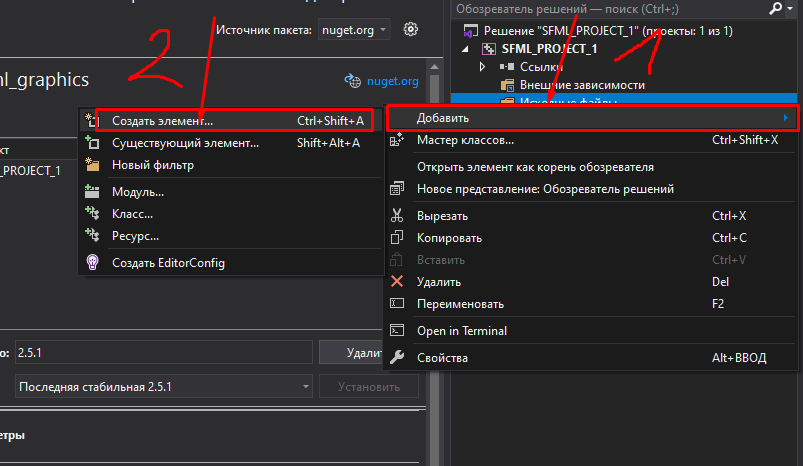


Создание главного файла

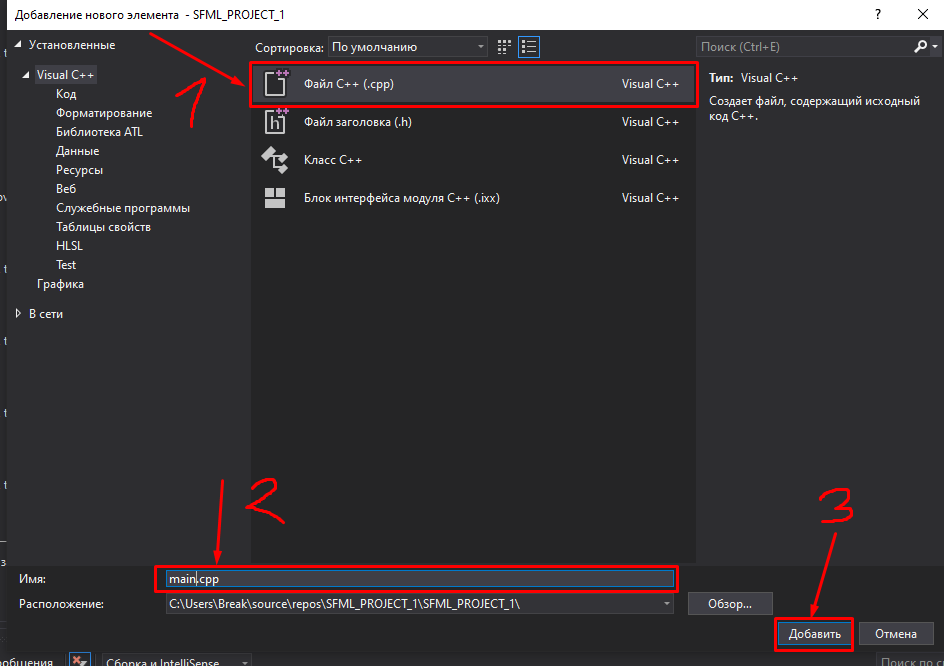
Библиотека установилась, нужно проверить доступна ли она. Для этого создадим файл с исходным кодом. В обозревателе решений, правой кнопкой мыши нажимаем на папку - исходные файлы.



Выбираем добавить и создать элемент.



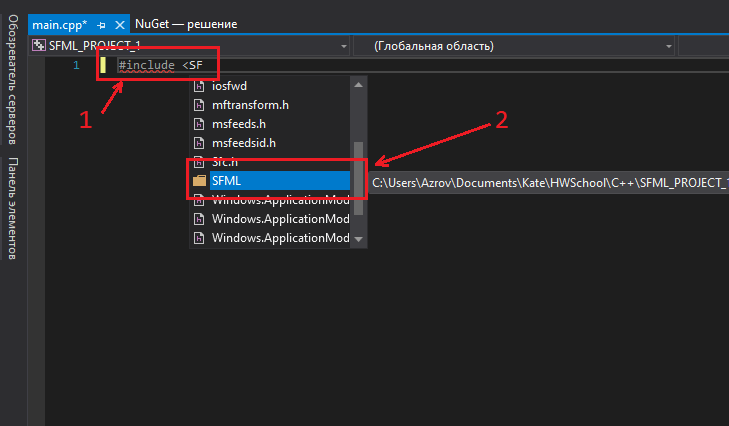
Затем выбираем файл С++. Меняем имя файла на main.cpp. И нажимаем добавить.



В файле начинаем печатать:

**#include <SFML/Graphics.hpp>**

Если подсказка VS находит библиотеку SFML, значит она успешно подключилась к нашему проекту.



Для того чтобы убедиться, что все работает, скидываем студенту тестовый код и пытаемся его запустить (зеленый треугольник на верхней панели). Разбираться с тем, что тут написано будем на следующем уроке.

**#include <SFML/Graphics.hpp>**

**int main()**

**{**

**sf::RenderWindow window(sf::VideoMode(200, 200), "SFML works!");**

**sf::CircleShape shape(100.f);**

**shape.setFillColor(sf::Color::Green);**

**while (window.isOpen())**

**{**

**sf::Event event;**

**while (window.pollEvent(event))**

**{**

**if (event.type == sf::Event::Closed)**

**window.close();**

**}**

**window.clear();**

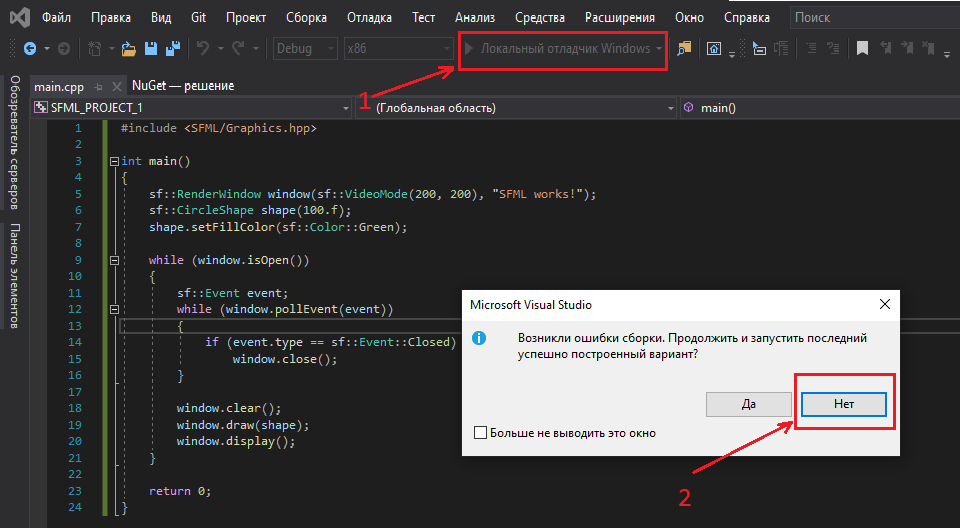
**window.draw(shape);**

**window.display();**

**}**

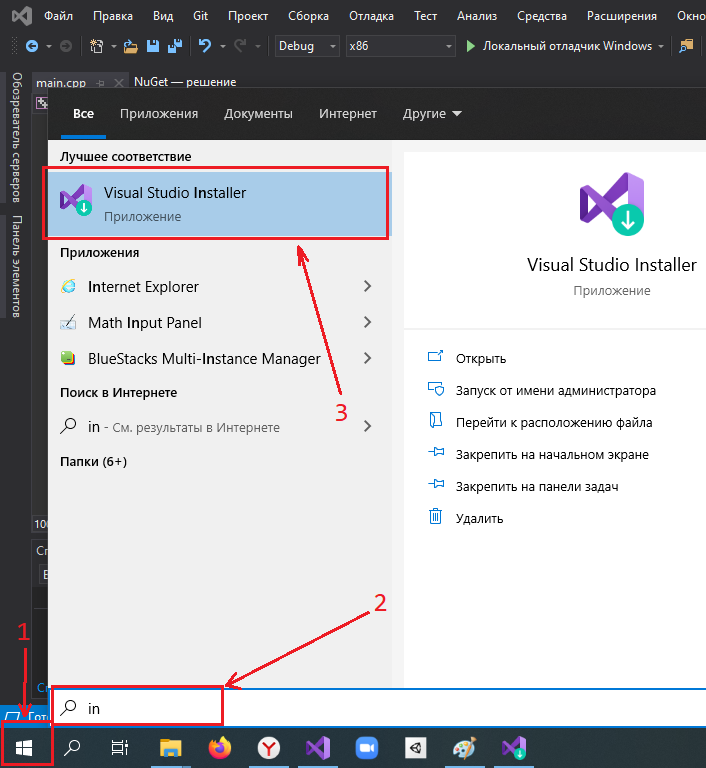
**return 0;**

**}**

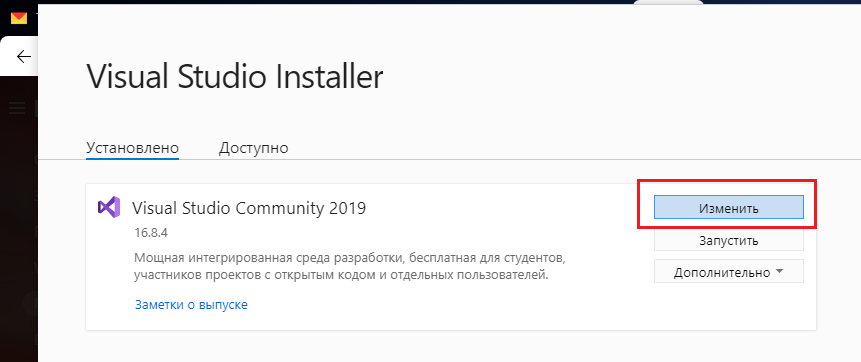
****

Ловим 18 ошибок. Для того чтобы их исправить нам нужно настроить Visual Studio.

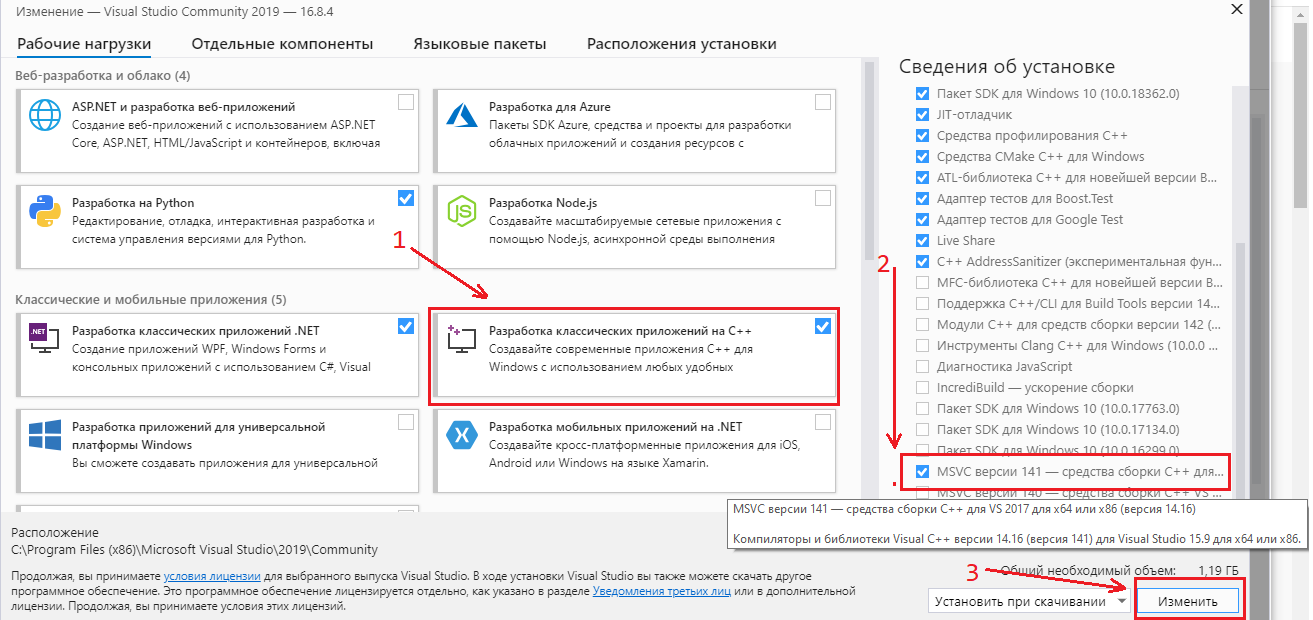
Настройка Visual Studio

Открываем меню пуск, начинаем набирать “installer”, находим программу Visual Studio Installer и запускаем ее.

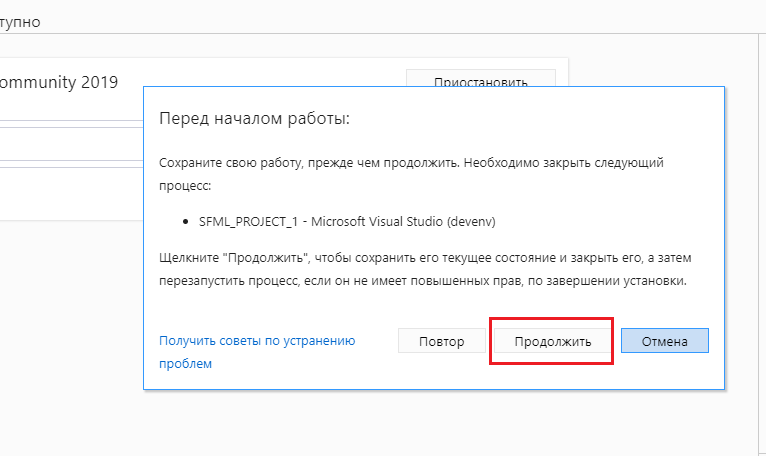
На вкладке Visual Studio 2019 Community нажимаем “Изменить”



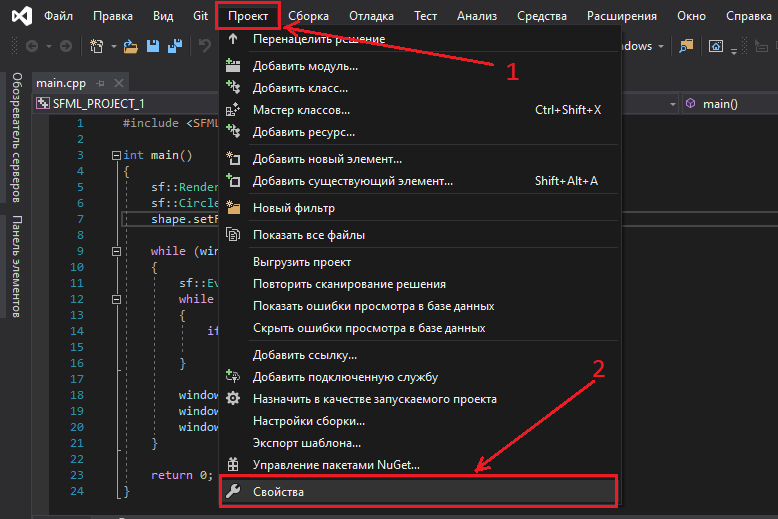
В открывшемся окне выбираем “Разработка классических приложений С++”, прокручиваем вниз и ставим галочку “MSVC версии 141 - средства разработки С++...”, затем нажимаем “Изменить”.



Появится окно, которое требует закрытия Visual Studio, закрываем Visual Studio и нажимаем кнопку “Продолжить”



Когда обновление установилось, запускаем Visual Studio. Теперь необходимо настроить проект. В верхнем меню находим пункт “Проект”, выбираем “Свойства”.

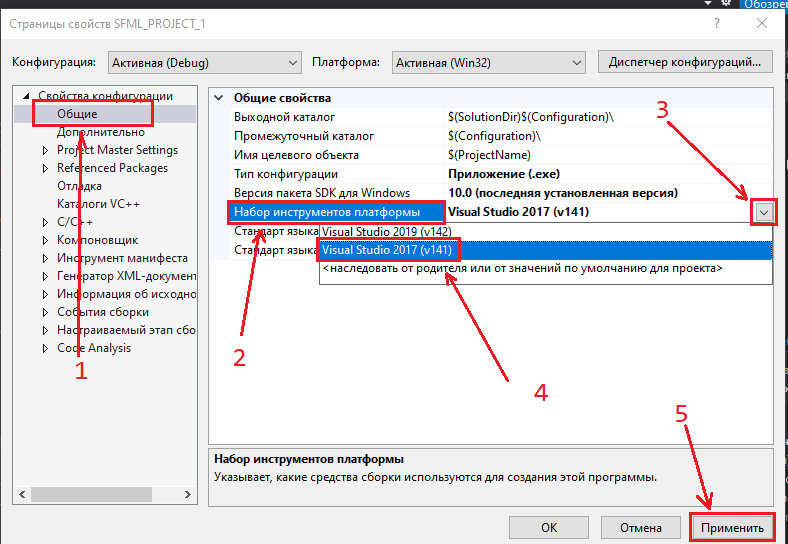


Меняем “Набор инструментов платформы” версию 141 и нажимаем “Применить”

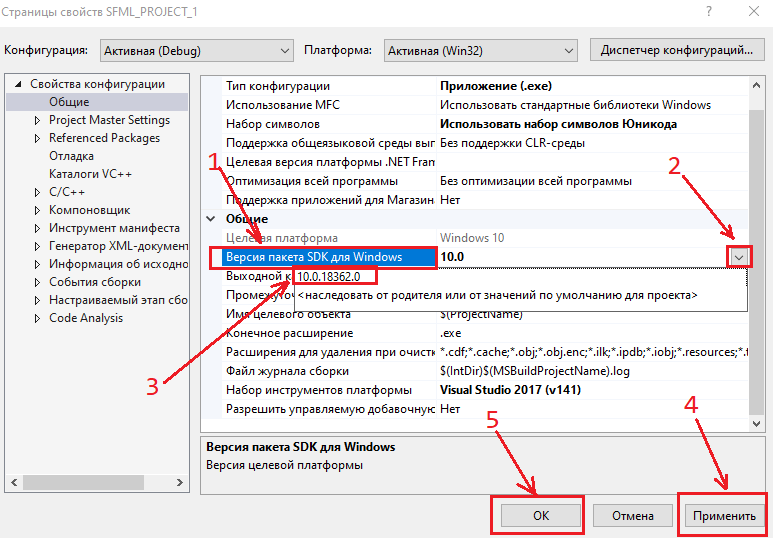
Для любопытных: почему не используем более свежую версию инструментов набора платформы. Для каждой версии VS свой набор:

* Visual Studio 2015: v140
* Visual Studio 2017: v141
* Visual Studio 2019: v142

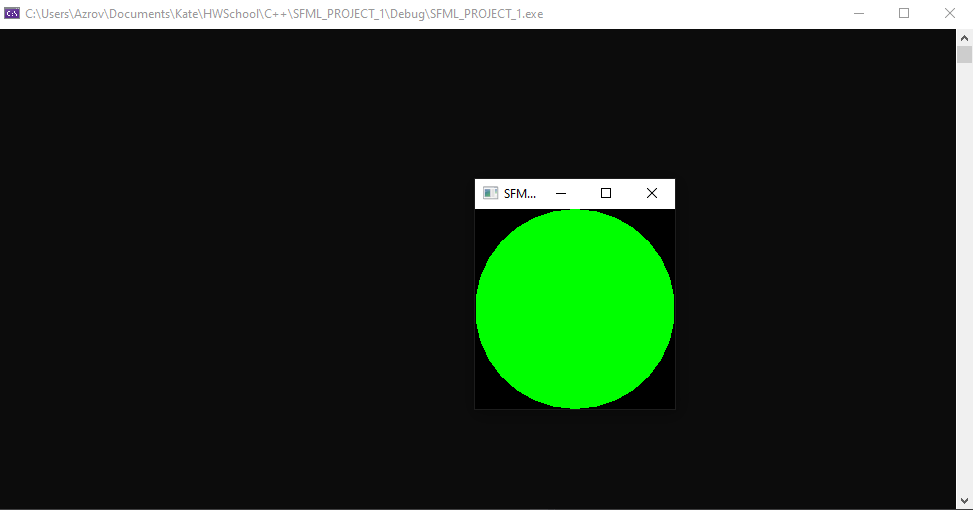
А библиотека SFML последний раз обновлялась в 2018 году, это значит, что она не совместима с 142 версией, так как та выпущена в 2019 году.



Меняем “Версия пакета SDK для Windows”, выбираем доступную, нажимаем “Применить”, потом “ОК”.



Запускаем, теперь ошибок нет, это означет, что библиотеку SFML мы успешно установили. Любуемся на зеленый круг.



На следующем уроке мы разберем подробно каждую строчку тестового кода и начнем писать свою игру.