Содержание

1) Глава – терминология (термины – слова).

2) Глава – редактор кода (текста) программы и другие нужные вещи.

3) Глава – пример программы вывода на экран текста с **подробным описанием** (вывод - показать).

4) Глава – переменные в C# (C SHARP)

1 Глава – терминология

**Термин** – слово.

В этой главе мы изучим главную терминологию языка с# (c sharp). Основу программы.

И начнём мы с главного -

namespace – основная часть программы

class – описание

переменная – значение которое может изменяться

константа – значение которое не может изменяться

массив – переменная из множества значений

public – публичный

static – статический

void – без значения

int – число (числовое значение)

string – строка (строковое значение)

метод – способ

компиляция – создание программы

интерпретация – выполнение программы самим редактором кода программы

if – если

2 Глава – редактор кода программы и другие нужные вещи

**Редактор** **кода** – редактор текста компилируемой программы. Текст программы можно писать даже в блокноте, но она не будет скомпилирована. Чтобы её скомпилировать нам потребуется редактор кода программы. Я буду всё показывать на примере sharp Develop.

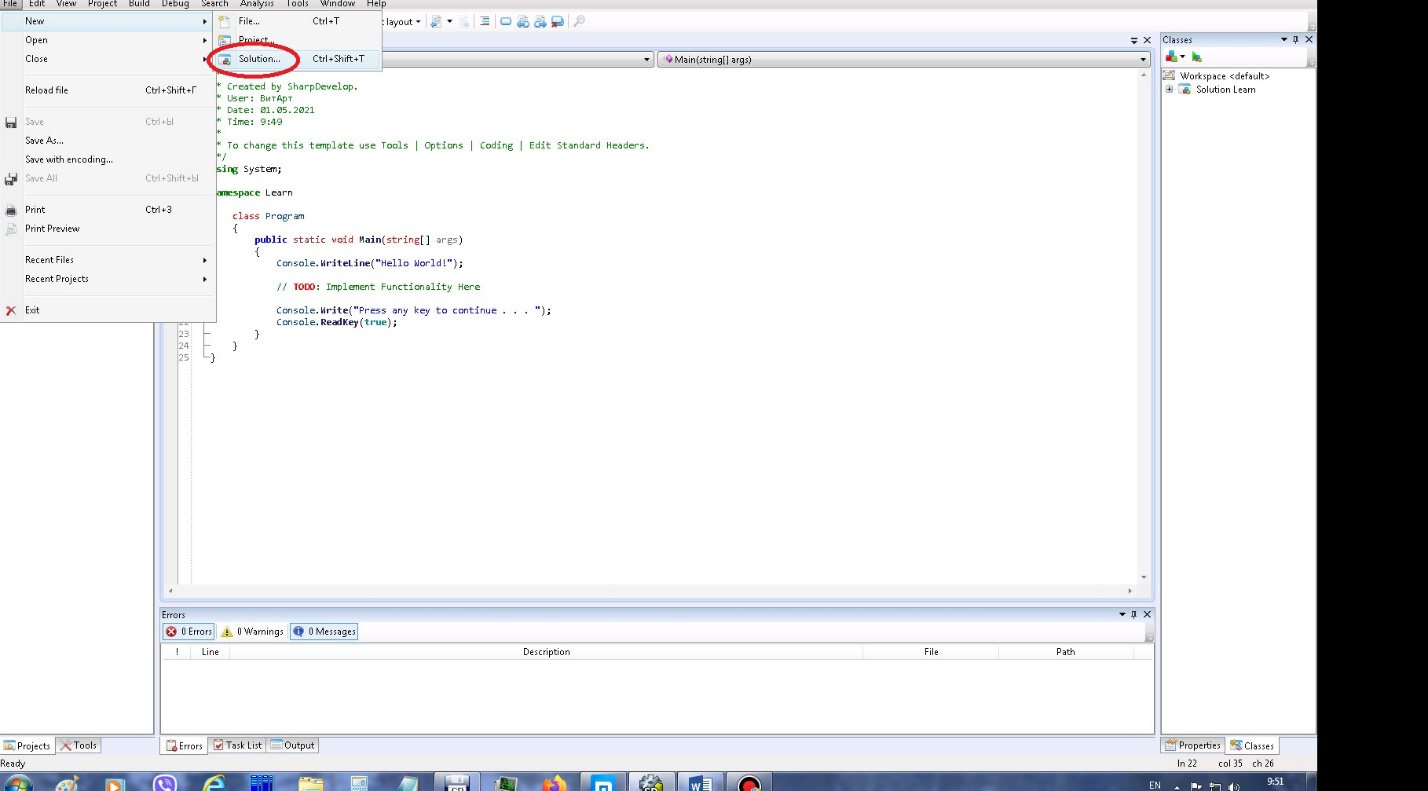
Преимущество sharp Develop в том что после фигурных скобок {} он ставит табуляцию (большой пробел). В программе при компиляции это важно! Но есть и Microsoft Visual studio ссылки на скачивание ниже.

Microsoft Visual Studio - <https://visualstudio.microsoft.com/ru/downloads/> ( ТОЛЬКО WINDOWS 10)

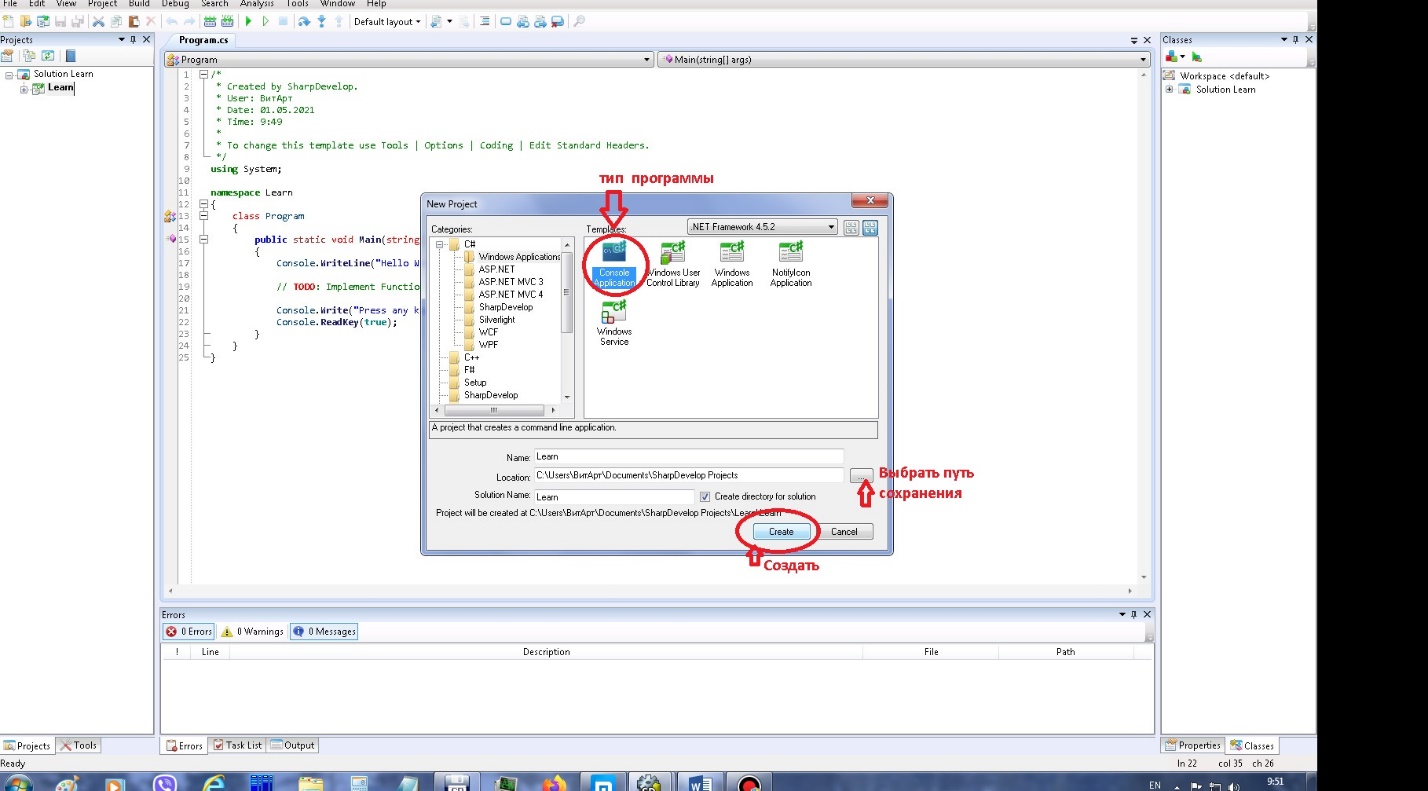
Sharp Develop - <https://sharpdevelop.ru.uptodown.com/windows/download>

3 Глава – пример программы вывода текста на экран с подробным описанием

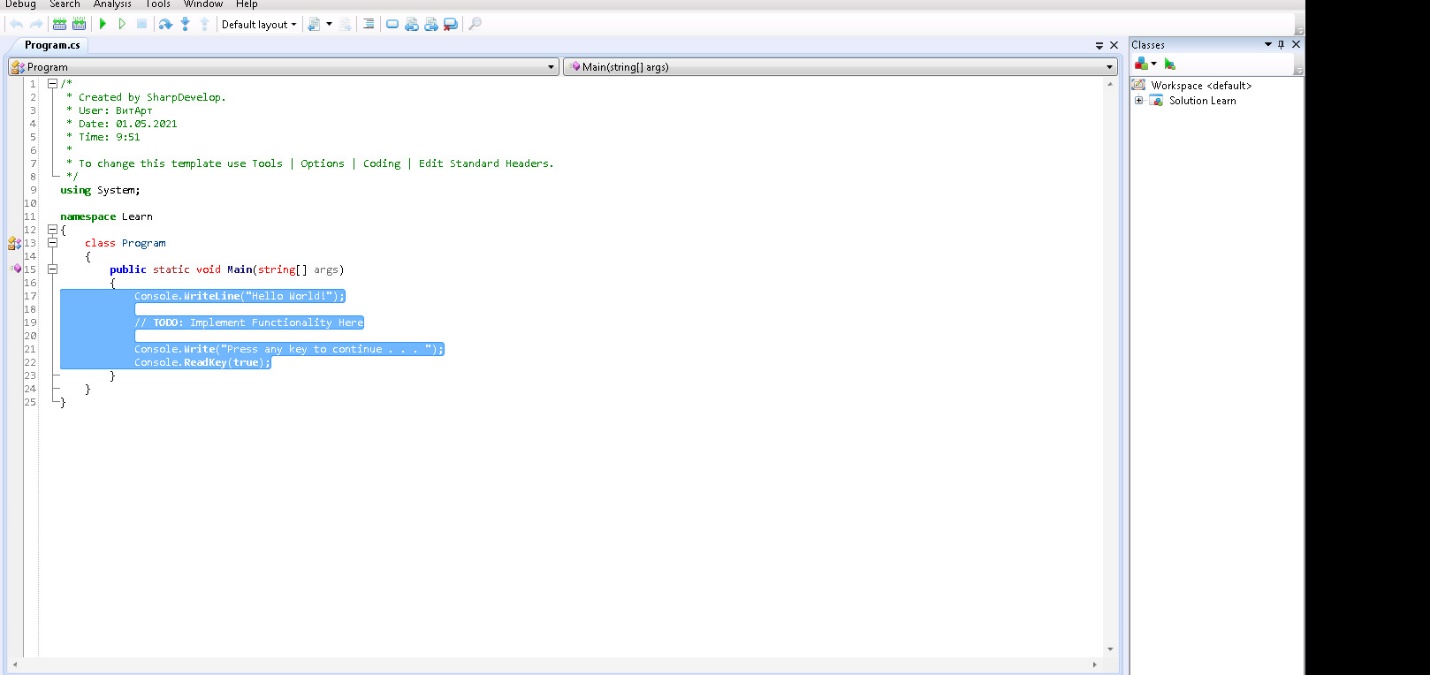
И для того чтобы начать писать программу нам надо создать **проект** создаётся он так:



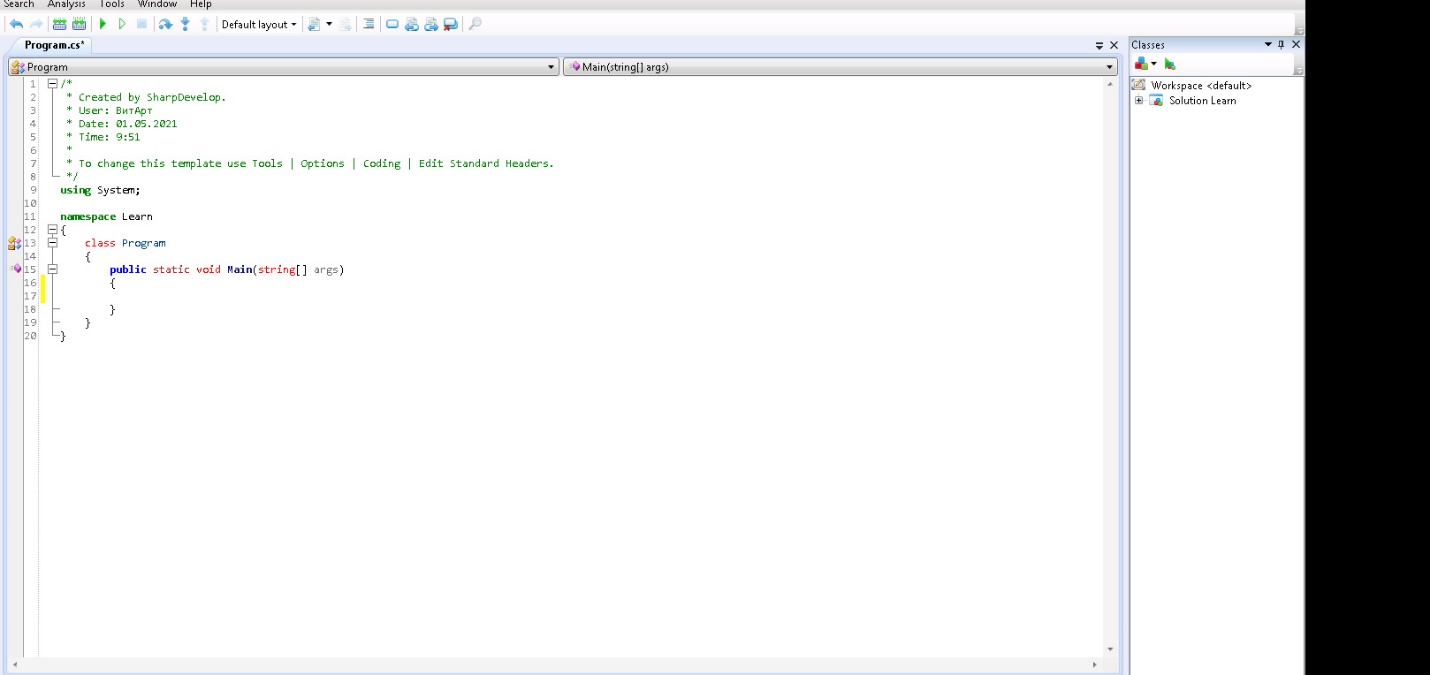
И последующие действия.



После того как мы создали проект мы можем писать наш код.



То, что выделено на рисунке то мы можем стереть.



И можем начать писать код. Теперь мы напишем простую программу. Её код прилагается ниже с объяснением ↓.

Using System; //Используем самую главную библиотеку. Сотрите весь код и впишите этот.

namespace Hello // Вместо Hello напишите название проекта

{ //Начало namespace

class Programm //Создаём класс

{ //Начало класса

public static void Main() //Создаём метод

{ //Начало метода

Console.WriteLine(“Hello world”); //Выводим на экран Hello world

} //Конец метода

} //Конец класса

} //Конец namespace

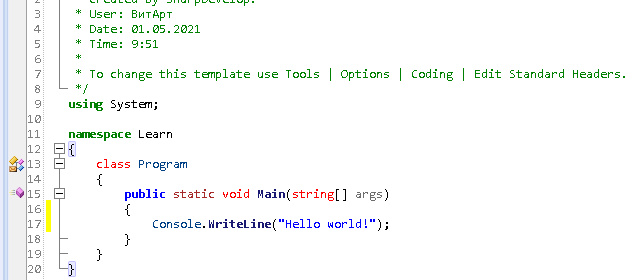
Давайте во всём разберёмся. Что за оператор using? Оператор **using** – нужен для использования библиотек языка c# (c sharp) .

Оператор namespace – имя тела вашей программы.

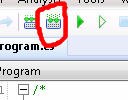
Оператор class – основа программы.

А дальше у нас не оператор а главный метод Main().

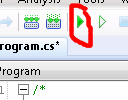
В этом методе у нас идёт главный оператор **Console.WriteLine()** – **для вывода информации на экран!**

****

После того как мы написали этот код мы должны скомпилировать программу (превратить в приложение) – для этого мы нажимаем на этот значок обведённый красным Это для того, чтобы скомпилировать программу.



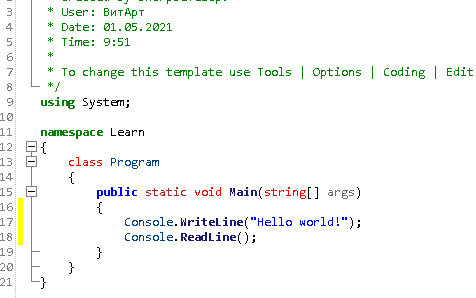
А потом нажимаем на этот значок обведённый красным. Чтобы запустить программу.



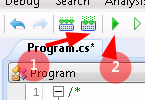
И видим что программа закрылась сразу же и мы ничего не увидели. Для исправления этой ошибки нам надо дописать вот этот кусок кода↓.

Console.ReadLine();

И тогда пока мы что нибудь не напишем и не нажмём enter у нас не закроется программа. Выглядеть это будет так ↓.



И теперь повторяем те же самые действия ↓.



И видим такой результат ↓.



Результат: у нас открылась консоль где написанно Hello world.

Скажем так для первого раза нормально.

4 Глава – переменные

Переменная состоит из четырёх частей. Тип переменной(число, строка или символ) название переменной, присваивание значения (знак =) знак равенства.

Вот так выглядит переменная выглядит в теории.

А в программном коде это выглядит вот так int name = 1; или string name = “Hello world” а вот ещё

сhar name = “H”; именно так переменные выглядят в коде программы. Они нужны для ввода информации от пользователя это в зависимости от заданного типа переменной или вывода текущего значения переменной. Пример:

**using** System;  
  
**namespace** Learn  
{  
    class Program  
    {  
        **public** static void **Main**()  
        {  
            string name;  
            name = Console.**ReadLine**();  
            Console.**WriteLine**(name);  
            Console.**ReadKey**();  
  
        }  
    }  
}

Это простая программа ввода от пользователя строки и вывод её на экран результат будет такой: