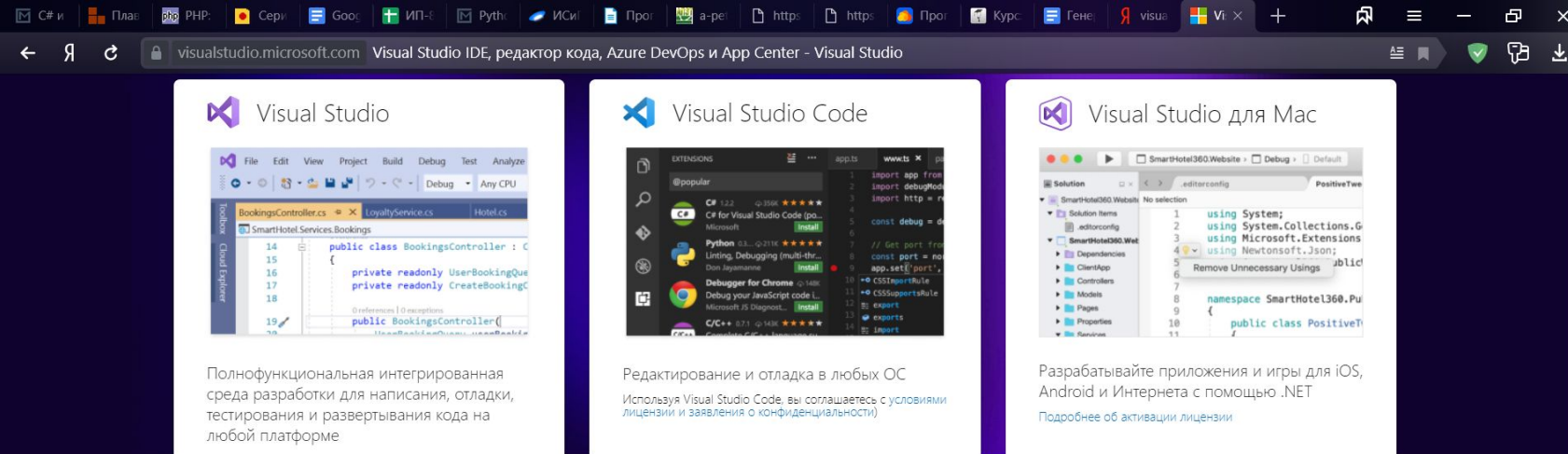


# Знакомство с Visual Studio

Ваша первая программа

Примечание: в презентации присутствуют гиперссылки для подробного изучения.

# Для создания приложения мы с вами будем использовать бесплатную среду разработки – Visual Studio 2019



The screenshot shows the Visual Studio website with three main sections:

- Visual Studio**: Полнофункциональная интегрированная среда разработки для написания, отладки, тестирования и развертывания кода на любой платформе. Includes a dropdown menu for "Скачайте Visual Studio" with options: Community 2019, Professional 2019, and Enterprise 2019.
- Visual Studio Code**: Редактирование и отладка в любых ОС. Includes a "Download Visual Studio Code" button and a "Подробнее..." link.
- Visual Studio для Mac**: Разрабатывайте приложения и игры для iOS, Android и Интернета с помощью .NET. Includes a "Скачайте Visual Studio для Mac" button and a "Подробнее..." link.

Below the main sections, there are three service cards:

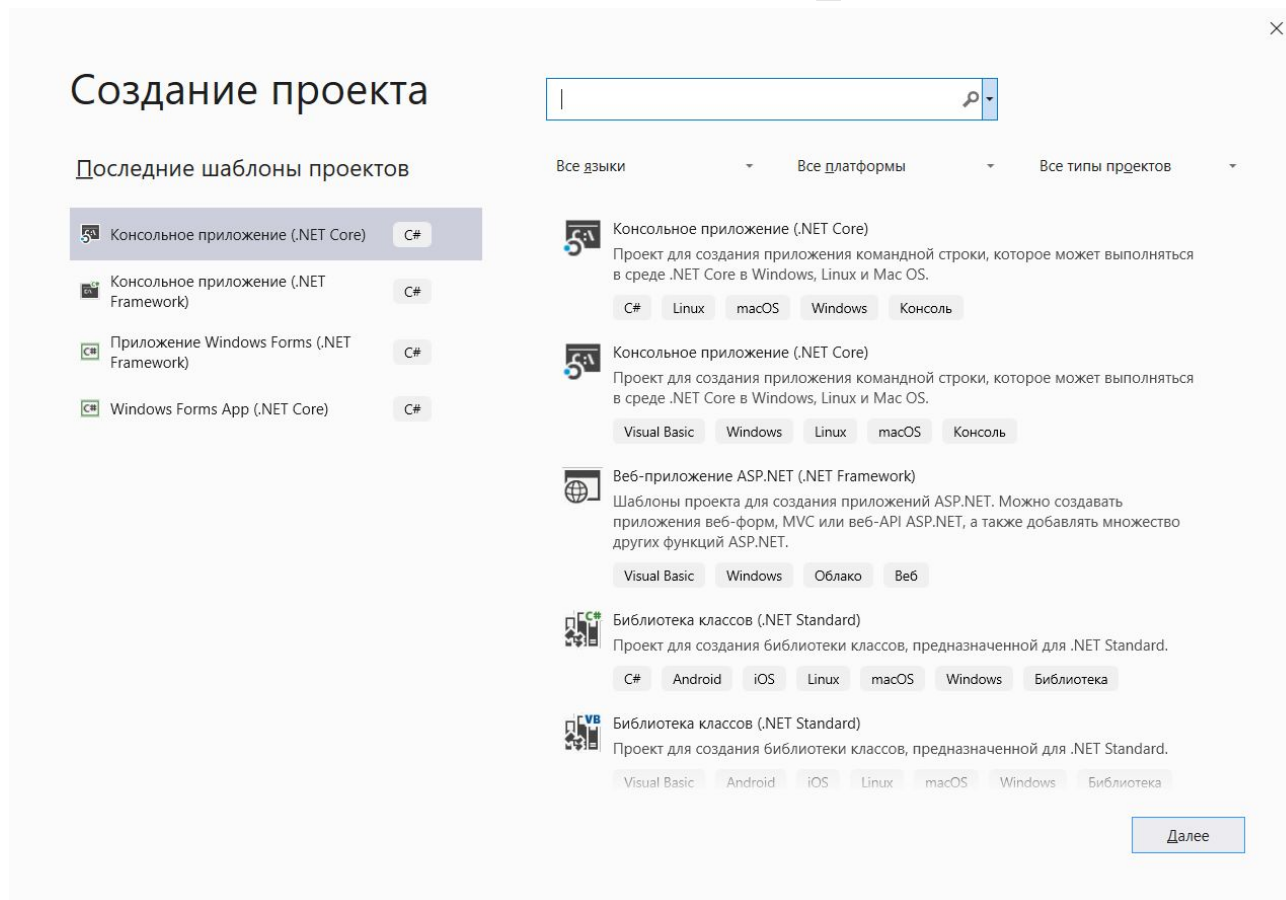
- Microsoft Azure**: Эффективные облачные решения, которые
- Azure DevOps**: Инструменты гибкой разработки, Git и непрерывная интеграция
- Visual Studio App Center**: Комплексные службы для разработчиков мобильных и классических приложений

A "Feedback" button is located on the right side of the page.

<https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community&rel=16>

# Создаем новый проект в данном окне

## Выбираем “Консольное приложение(.NetCore)”



# Следующее окно у нас будет настройки проекта

×

## Настроить новый проект

Консольное приложение (.NET Core) C# Linux macOS Windows Консоль

Имя проекта

ConsoleApp19

Расположение

D:\VisualStudioProject

...

Решение

Создать новое решение

Имя решения ⓘ

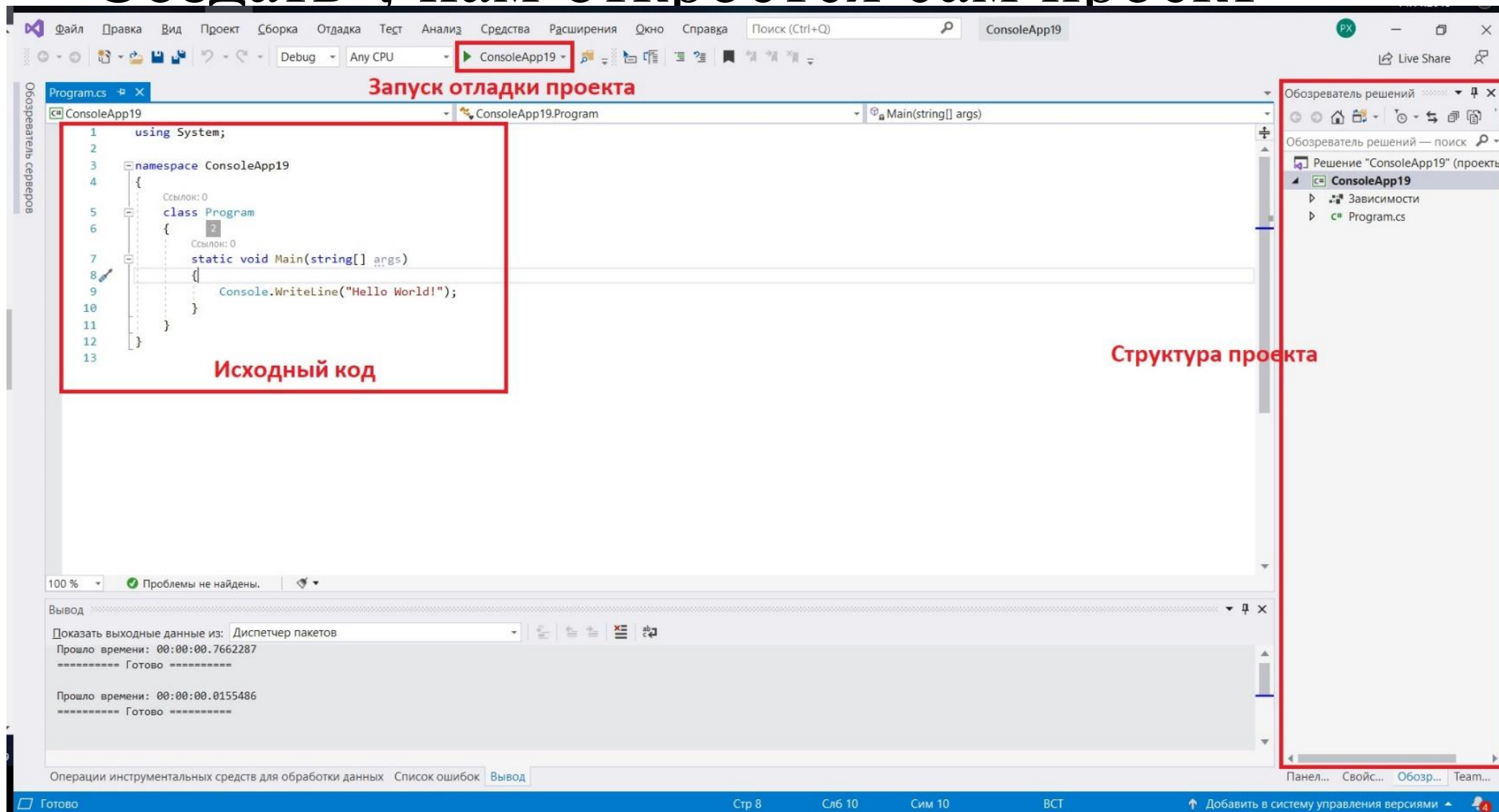
ConsoleApp19

☐ Поместить решение и проект в одном каталоге

Назад

Создать

# После того как мы нажмем кнопку “Создать”, нам откроется сам проект



```
1  using System; //Подключение пространства имен
2
3  namespace ConsoleApp19 //объявление нового пространства имен
4  {
5      class Program // объявление нового класса
6      {
7          static void Main(string[] args) //объявление нового метода
8          {
9              Console.WriteLine("Hello World!"); // действие метода
10         }
11     }
12 }
13
```

Ссылки в коде:

- Ссылка на `using System;` (строка 1): Ссылка: 0
- Ссылка на `namespace ConsoleApp19` (строка 3): Ссылка: 0
- Ссылка на `class Program` (строка 5): Ссылка: 0
- Ссылка на `static void Main` (строка 7): Ссылка: 0
- Ссылка на `Console.WriteLine` (строка 9): Ссылка: 0

- Ключевое слово [namespace](#) используется для объявления области действия, которая содержит набор связанных объектов. Пространство имен можно использовать для организации элементов кода и для создания глобально уникальных типов.
- [Классы](#) являются основным типом в языке C#. Класс представляет собой структуру данных, которая объединяет в себе значения (поля) и действия (методы и другие функции-члены). Класс предоставляет определение для динамически создаваемых экземпляров класса, которые также именуются объектами. Классы поддерживают механизмы наследования и полиморфизма, которые позволяют создавать производные классы, расширяющие и уточняющие определения базовых классов. Новые классы создаются с помощью объявлений классов. Объявление класса начинается с заголовка, в котором указаны атрибуты и модификаторы класса, имя класса, базовый класс (если есть) и интерфейсы, реализуемые этим классом. За заголовком между разделителями { и } следует тело класса, в котором последовательно объявляются все члены класса.

Директива using используется в следующих трех целях:

- Для разрешения использования типов в пространстве имен, чтобы не нужно было квалифицировать использование типа в этом пространстве имен
- Для разрешения доступа к статическим членам и вложенным типам без необходимости квалифицировать доступ с помощью имени типа.
- Чтобы создать псевдоним для пространства имен или типа. Это называется директивой `using static`.

Ключевое слово `using` также используется для создания операторов `using`, которые помогают обеспечить правильную обработку объектов `IDisposable`, таких как файлы и шрифты.



# Напишем самостоятельно свое первое приложение

- Для этого нам понадобится:
- `Console.WriteLine()` – Вывод данных в консоль
- `Console.ReadLine()` – Ввод данных в консоль

Так же мы будем использовать объявление переменных, рассмотрим это в будущем подробно.

```
1  using System; //Подключение пространства имен
2
3  namespace ConsoleApp19 //объявление нового пространства имен
4  {
5      class Program // объявление нового класса
6      {
7          static void Main(string[] args) //объявление нового метода
8          {
9              Console.WriteLine("Привет создатель, как тебя зовут?");
10             string answeruser = Console.ReadLine(); //string - это строковый тип данных, answeruser - имя переменной,
11                                                         //в которой сохранится наш ответ
12             Console.WriteLine("А как ты назовешь меня?");
13             string namebot = Console.ReadLine();
14             Console.WriteLine($"Приятно познакомится {answeruser}. Меня зовут {namebot} - это ты меня так назвал.");
15         }
16     }
17 }
18
```

 Консоль отладки Microsoft Visual Studio

Привет создатель, как тебя зовут?

Руслан

А как ты назовешь меня?

Жорик

Приятно познакомится Руслан. Меня зовут Жорик - это ты меня так назвал.

**Поздравляю, теперь вы**  
**погружаетесь в мир разработки!**

**Примечание: Сделаю приложение к этой лекции, для тех у кого не получится сделать свой проект(для разборки ошибок)**