

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

Visual Studio
Полнофункциональная интегрированная среда разработки для написания, отладки, тестирования и развертывания кода на любой платформе.
[Скачайте Visual Studio](#)
Community 2019
Professional 2019
Enterprise 2019

Visual Studio Code
Редактирование и отладка в любых ОС
Используя Visual Studio Code, вы соглашаетесь с условиями лицензии и заявления о конфиденциальности.
[Download Visual Studio Code](#)
[Подробнее...](#)

Visual Studio для Mac
Разрабатывайте приложения и игры для iOS, Android и Интернета с помощью .NET
[Скачайте Visual Studio для Mac](#)
[Подробнее...](#)

СЛУЖБЫ ДЛЯ РАЗРАБОТЧИКОВ



Microsoft Azure

Эффективные облачные решения, которые



Azure DevOps

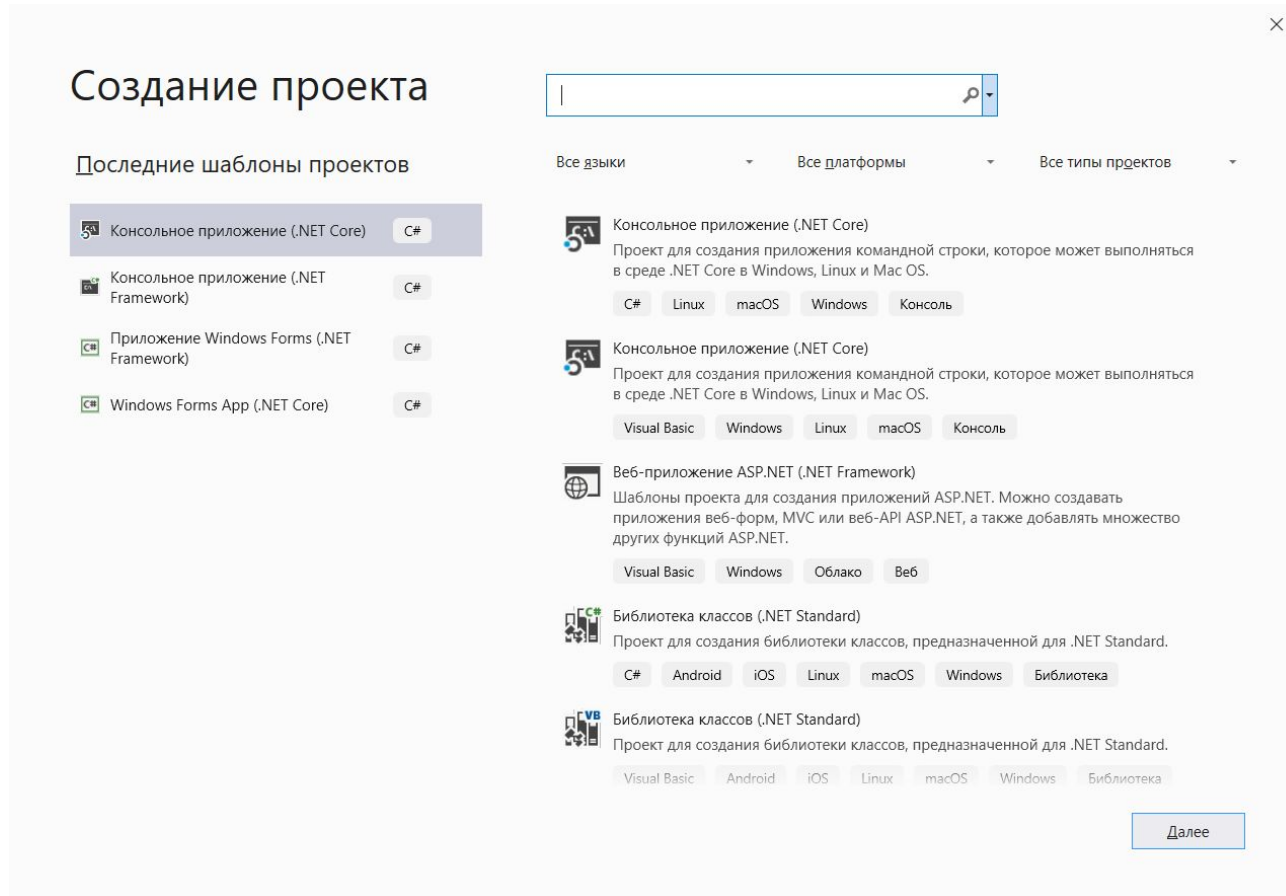
Инструменты гибкой разработки, Git и непрерывная интеграция



Visual Studio App Center

Комплексные службы для разработчиков мобильных и классических приложений

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

Запуск отладки проекта

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

Исходный код

Структура проекта

Решение "ConsoleApp19" (проект...)
ConsoleApp19
Зависимости
Program.cs

Вывод
Показать выходные данные из: Диспетчер пакетов
Прошло времени: 00:00:00.7662287
----- Готово -----

Прошло времени: 00:00:00.0155486
----- Готово -----

Операции инструментальных средств для обработки данных Список ошибок Вывод

Готово Стр 8 Слб 10 Сим 10 ВСТ Добавить в систему управления версиями

```
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
```

Ссылок: 0

2

Ссылок: 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
```

CA 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)