

Язык C# Платформа .NET Visual Studio

Выполнил: преподаватель ГАПОУ СО
«ЕКТС»
Г.В. Мирошниченко

Язык программирования

- Язык программирования – это набор правил, с помощью которых программист записывает исходную программу.
- ЯП:
 - низкоуровневые (машинно ориентированные)
 - высокоуровневые
 - Процедурные (структурные) ЯП (в основе «Алгоритм»)
 - Функции + Данные
 - Объектно-ориентированные ЯП (в основе «Модель», содержащая алгоритм)
 - Объекты = Данные + функции вокруг данных.

Цели создания C# (Си-шарп)

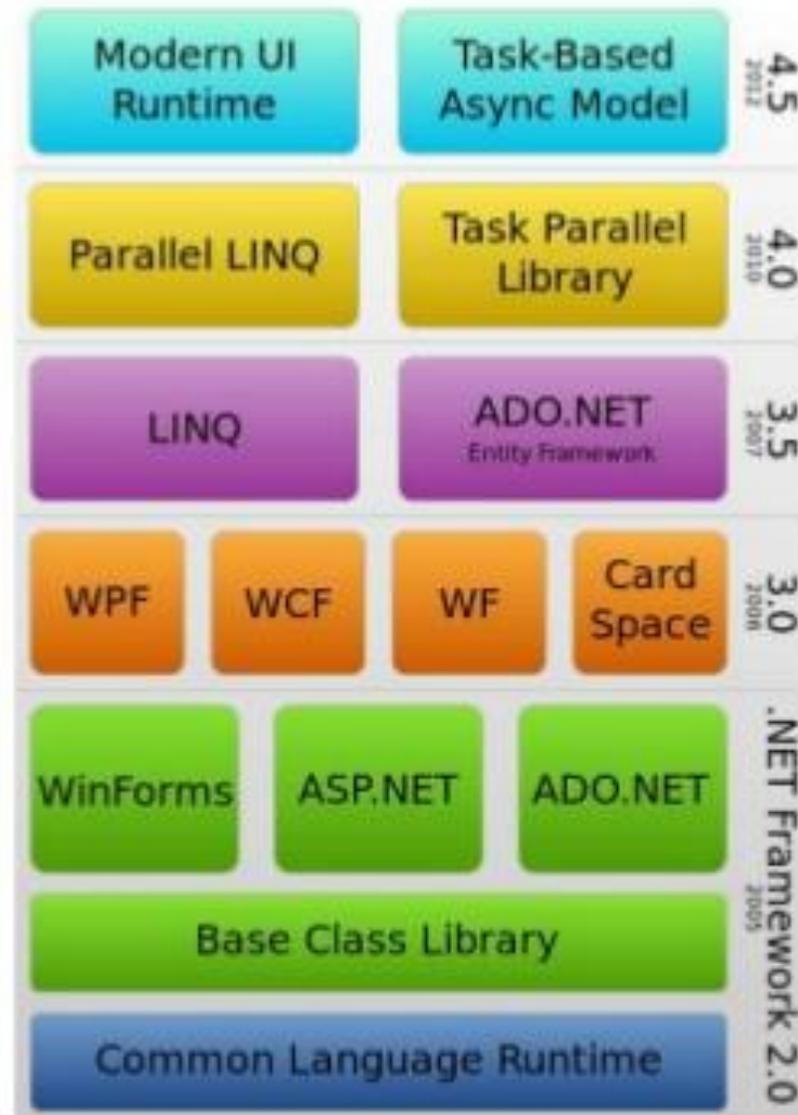
- создать объектно-ориентированный язык, в котором любая сущность представляется объектом, объединяющим как информационную (данные), так и функциональную (действия над этими данными) части;
- создать первый компонентно-ориентированный язык программирования семейства C/C++;
- упростить C++, сохранив по возможности его мощь и основные конструкции;
- повысить надёжность программного кода.



**Андерс
Хейлсберг,
разработчик
Delphi и C#**

C#
многое перенял
от C++ и Java

.NET Framework Stack



C# основной язык платформы [.NET Framework](#) (произносится как дот-нет), которая является средой, объединяющей программные технологии, для разработки Web- и Windows-приложений (отсюда и название).

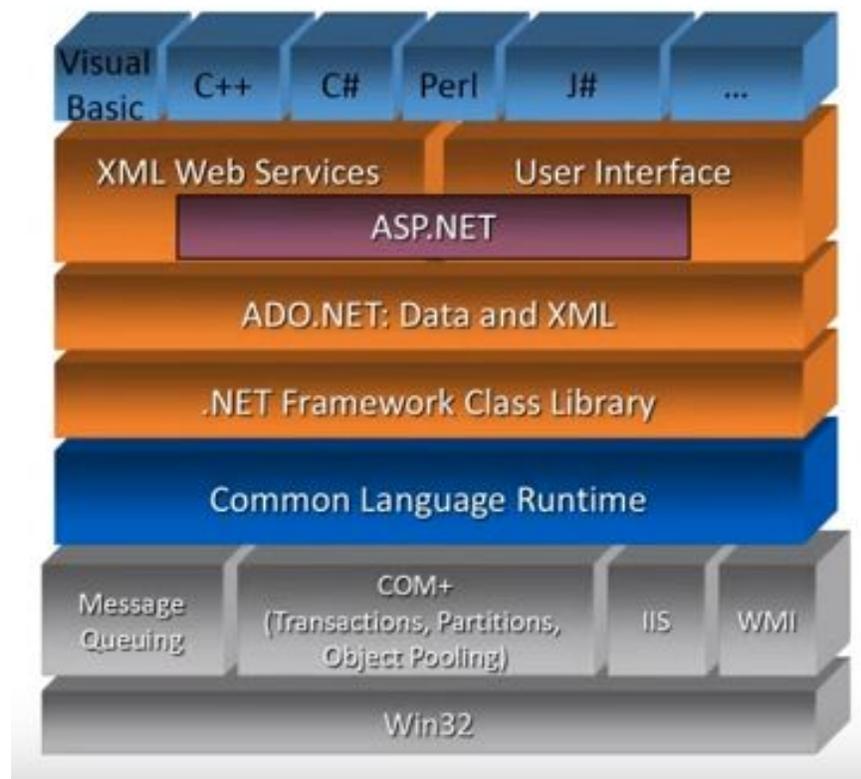
Основу среды .NET составляет CLR (Common Language Runtime) – общезыковая среда исполнения, которая состоит из двух основных частей:

- ядра (набор служб, управляющих загрузкой приложения в память, собраны в библиотеке mscorlib.dll);
- библиотеки базовых классов (главная сборка в библиотеке mscorlib.dll).

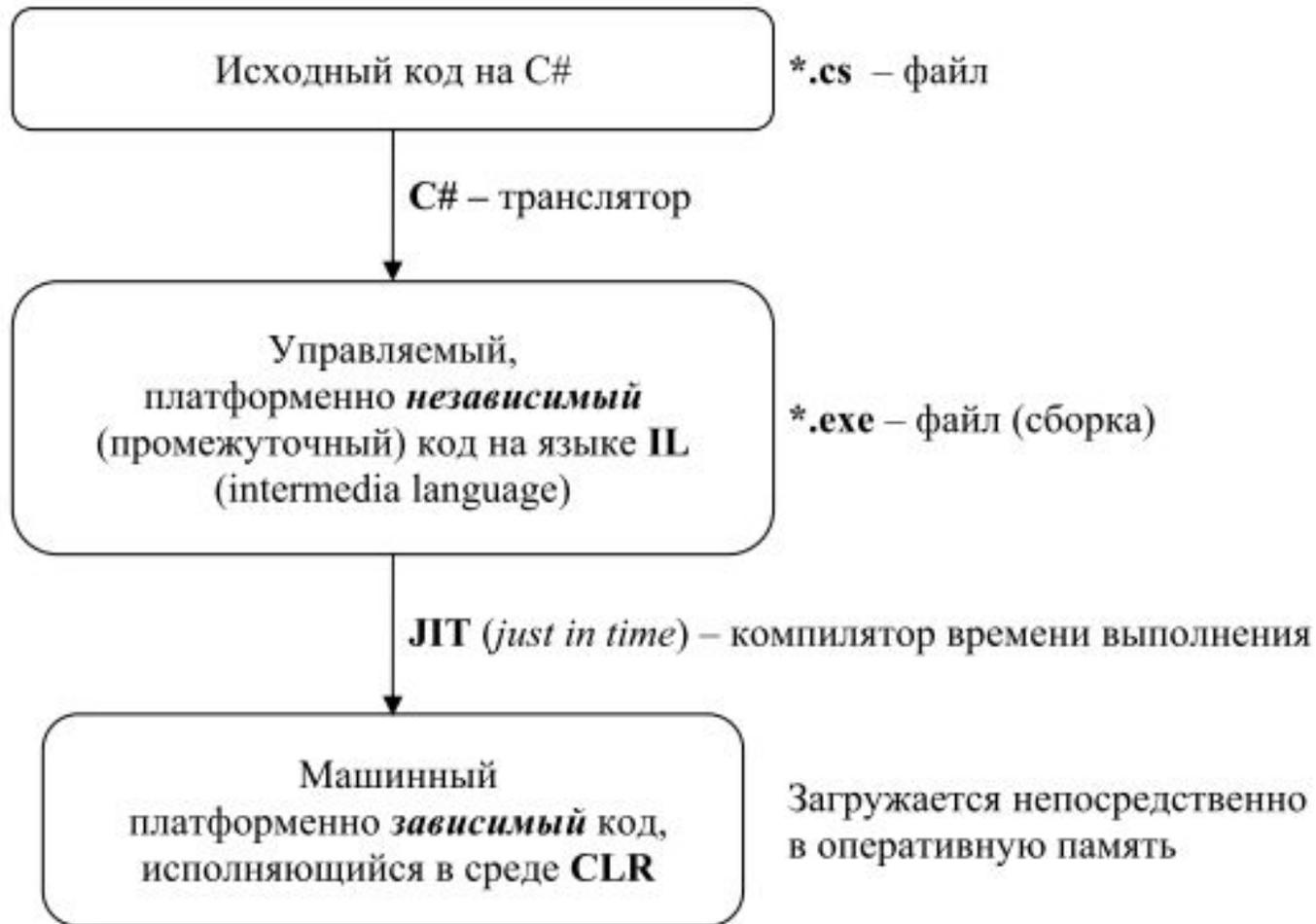
Роль платформы .NET

- **Поддержка нескольких языков.** Основой платформы является общезыковая среда исполнения Common Language Runtime (CLR), благодаря чему .NET поддерживает несколько языков: наряду с C# это также VB.NET, C++, F#, а также различные диалекты других языков, привязанные к .NET, например, Delphi.NET. При компиляции код на любом из этих языков компилируется в сборку на общем языке CIL (Common Intermediate Language) - своего рода ассемблер платформы .NET. Поэтому мы можем сделать отдельные модули одного приложения на отдельных языках.
- **Кроссплатформенность.** .NET является переносимой платформой (с некоторыми ограничениями). Например, последняя версия платформы на данный момент .NET Framework поддерживается на большинстве современных ОС Windows (Windows 10/8.1/8/7/Vista). А благодаря проекту Mono можно создавать приложения, которые будут работать и на других ОС семейства Linux, в том числе на мобильных платформах Android и iOS.
- **Мощная библиотека классов.** .NET представляет единую для всех поддерживаемых языков библиотеку классов. И какое бы приложение мы не собирались писать на C# - текстовый редактор, чат или сложный веб-сайт - так или иначе мы задействуем библиотеку классов .NET.
- **Разнообразие технологий.** Общезыковая среда исполнения CLR и базовая библиотека классов являются основой для целого стека технологий, которые разработчики могут задействовать при построении тех или иных приложений. Например, для работы с базами данных в этом стеке технологий предназначена технология ADO.NET. Для построения графических приложений с богатым насыщенным интерфейсом - технология WPF. Для создания веб-сайтов - ASP.NET и т.д.

До и после



Процесс создания приложений на C#



Первая программа на C# ([ссылка на](#)

[MSDN](#))

```
using System;
```

```
class Hello (имя программы)
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        Console.WriteLine("Hello, World");
```

```
    }
```

```
}
```

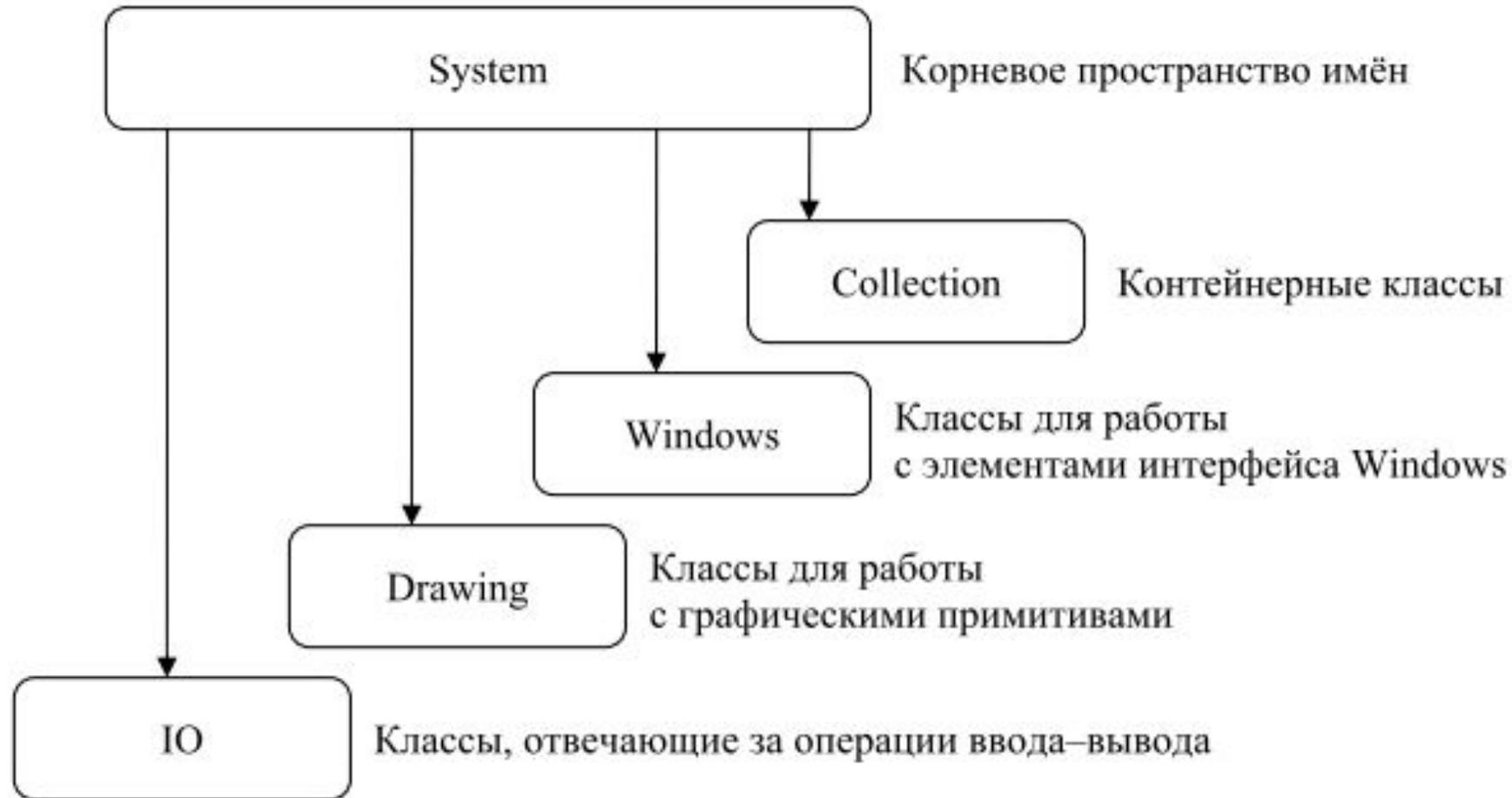
```
Здравствуй, Мир!
```

```
Для продолжения нажмите любую клавишу . . .
```

Здравствуй, мир

- Программа "Hello, World" начинается с директивы **using**, которая ссылается на пространство имен **System**.
- Пространства имен позволяют иерархически упорядочивать программы и библиотеки C#. Пространства имен содержат типы и другие пространства имен. Например, пространство имен **System** содержит несколько типов (в том числе используемый в нашей программе класс **Console**) и несколько других пространств имен, таких как **IO** и **Collections**. Директива **using**, которая ссылается на пространство имен, позволяет использовать типы из этого пространства имен без указания полного имени. Благодаря директиве **using** в коде программы можно использовать сокращенное имя [Console.WriteLine](#) вместо полного варианта [System.Console.WriteLine](#).
- Класс **Hello**, объявленный в программе "Hello, World", имеет только один член — это метод с именем **Main**. Метод **Main** объявлен с модификатором **static**. Методы экземпляра могут ссылаться на конкретный экземпляр объекта, используя ключевое слово **this**, а статические методы работают без ссылки на конкретный объект. **По стандартному соглашению точкой входа программы C# является статический метод с именем **Main**.**
- Выходные данные программы создаются в методе **WriteLine** класса **Console** из пространства имен **System**. Этот класс предоставляется библиотеками стандартных классов, ссылки на которые компилятор по умолчанию добавляет автоматически.

Пространство имён



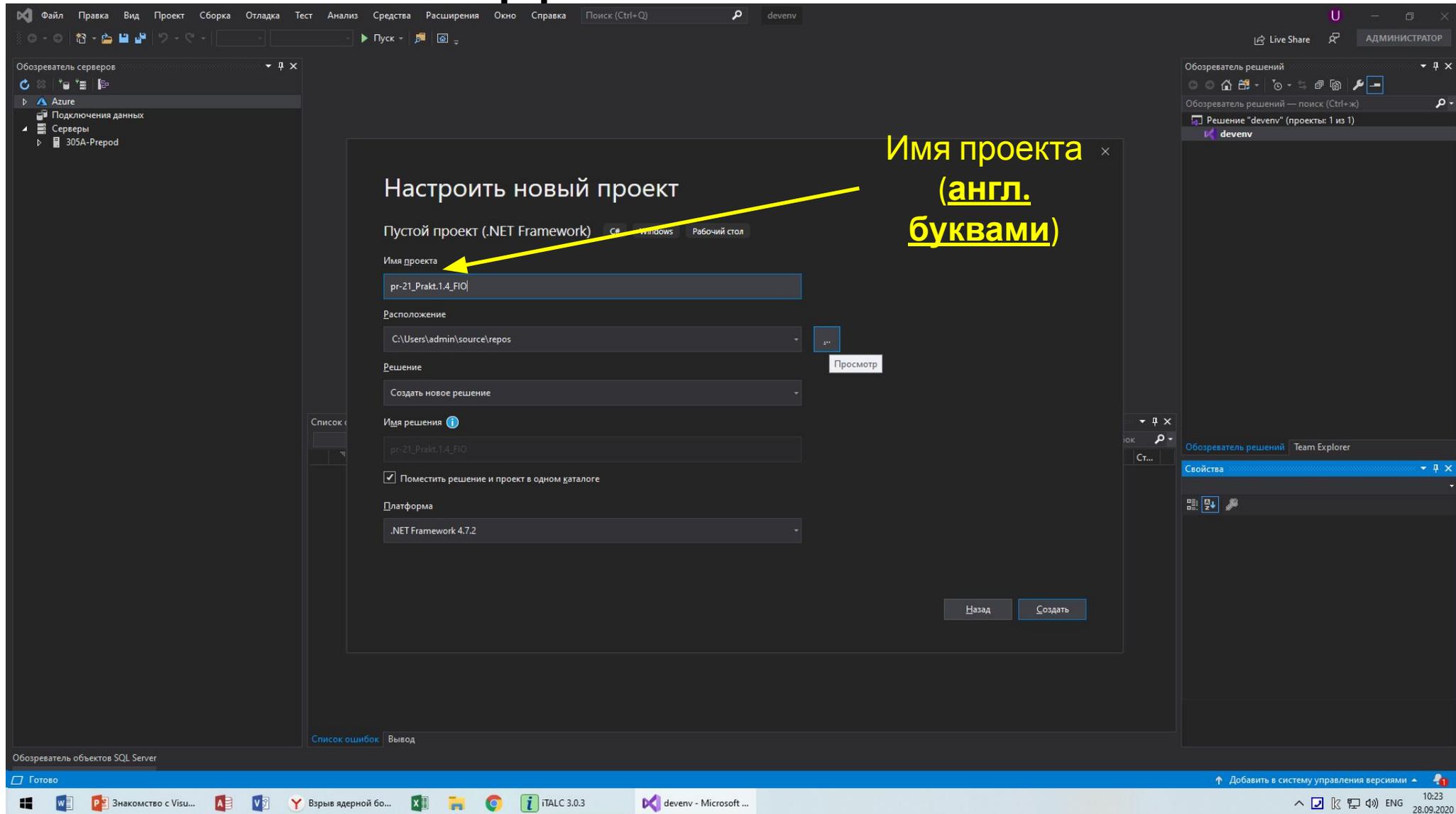
Что можно создать на C#?

- Настольные приложения
- Web-приложения
- Сервисы
- Мобильные приложения
- Игровые программы
- Решения для бизнеса

Знакомство с Visual Studio

[Ссылка на MSDN](#)

Как выглядит панель Visual Studio



План

- 1. IDE
- 2. Кратко об MS Visual Studio
- 3. Где взять?
- 4. Работа в MS Visual Studio

IDE

- IDE – Integrated Development Environment
- Это система программных средств, используемая программистами для разработки программного обеспечения (ПО)
- IDE (среда разработки) включает:
 1. текстовый редактор
 2. компилятор и/или интерпретатор
 3. средства автоматизации сборки
 4. отладчик

Visual Studio

- Microsoft Visual Studio — линейка продуктов, включающих **интегрированную среду разработки ПО** и ряд других инструментальных средств.
 - консольные приложения
 - приложения с графическим интерфейсом (с поддержкой технологии Windows Forms)
 - веб-сайты, веб-приложения, веб-службы
 - для всех платформ, поддерживаемых Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Microsoft Silverlight
- В состав Visual Studio входит:
 - Visual Basic .NET (Visual Basic чуть-чуть отличается синтаксисом)
 - Visual C++
 - Visual C#
 - Visual F# (начиная с VB2010)

Где взять Visual Studio?

- <https://www.visualstudio.com/>
- Версии ([сравнение](#)):
 - Premium
 - Professional
 - Ultimate
 - **Visual Studio Community** ([описание](#))
 - **Visual Studio Express**

Текстовый редактор (IntelliSense)

- [IntelliSense](#) - технология автодополнения Microsoft
- [Использование технологии](#)

Методы Ввод / вывод на C#

- **Ввод**

- Console.ReadLine();
- Console.Read()

- **Вывод**

- Console.WriteLine();
- Console.Write()

- [Работа с консолью, видео](#)

Задание:

- Используя, пример простейшей программы «Hello, world» и методы Ввода / вывода класса Console, откорректируйте программу, добавив ввод своей фамилии;
- Выходные данные – «Здравствуй, мир! Я ФИО (**ваша фамилия**)»

Полезные ресурсы

- [MSDN](#) – Microsoft Developer Network
- [MSDN: Основы языка C#](#)
- MSDN: Использование среды разработки

- <http://habrahabr.ru>
- <http://stackoverflow.com>

- Основы работы в VS ([youtube](#), [youtube](#))

Литература

- Visual C# ([описание](#))
 - [Руководство по программированию C#](#)
 - [Справочник по C#](#)
- Герберт Шилдт «C# 3.0, 4.0. Полное руководство»
- Совершенный код [Стив Макконнелл]
«Пишите код так, как будто сопровождать его будет склонный к насилию психопат,
который знает, где вы живете»

Стив Макконнелл