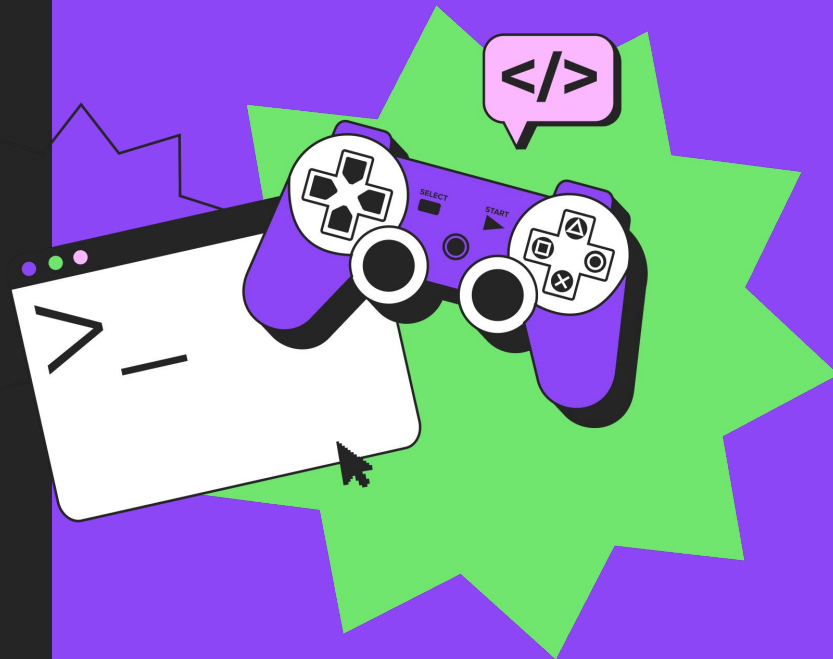


# Углубленный курс C#

## Урок 1

Виды .Net, среда разработки, основные файлы проекта,  
сборщик мусора, debug и release сборка





## Кирилл Орлов

Руководитель направления разработки эквайринговых решений

Пишу на C#, C++ и Kotlin

- 💥 ПО для работы с банковскими терминалами, ПО для сертификации банков в МПС, ПО для виртуализации
- 💥 ТрансКредитБанк, Совкомбанк, Abbrevia, Holistic Technologies, КейтМедиа



# План курса

1

Начало работы: .Net и Visual Studio

2

Основные конструкции языка

3

Классы и ООП

4

Интерфейсы и обобщения

5

Работа с датой и временем, пространство имен System.Collections и коллекции LINQ

6

Реализация паттерна Publisher/Subscriber

7

Управляемый и неуправляемый код: обработка ошибок

8

Получение информации об объекте во время выполнения приложения

9

Сериализация: все о сериализации (Байт, XML, JSON)

10

Работа с файловой системой: потоки и буферизация

11

Работа с сетью: чтение и запись данных в сеть. Клиентские и серверные приложения

12

Синхронизации : многопоточность, создание и завершение потоков



# Почему C#



## Преимущества С#

- Современный



## Преимущества C#

- Современный
- Мультиплатформенный



## Преимущества C#

- Современный
- Мультиплатформенный
- Вобрал все лучшее из C++ и Java



## Преимущества C#

- Современный
- Мультиплатформенный
- Вобрал все лучшее из C++ и Java
- Тысячи вакансий





## Преимущества C#

- Современный
- Мультиплатформенный
- Вобрал все лучшее из C++ и Java
- Тысячи вакансий
- Desktop, web и мобильная разработка



## Преимущества C#

- Современный
- Мультиплатформенный
- Вобрал все лучшее из C++ и Java
- Тысячи вакансий
- Desktop, web и мобильная разработка
- Топ-3 в мировом рейтинге языков программирования



# Грейды разработчиков



# Junior



# Mid-level



# Senior



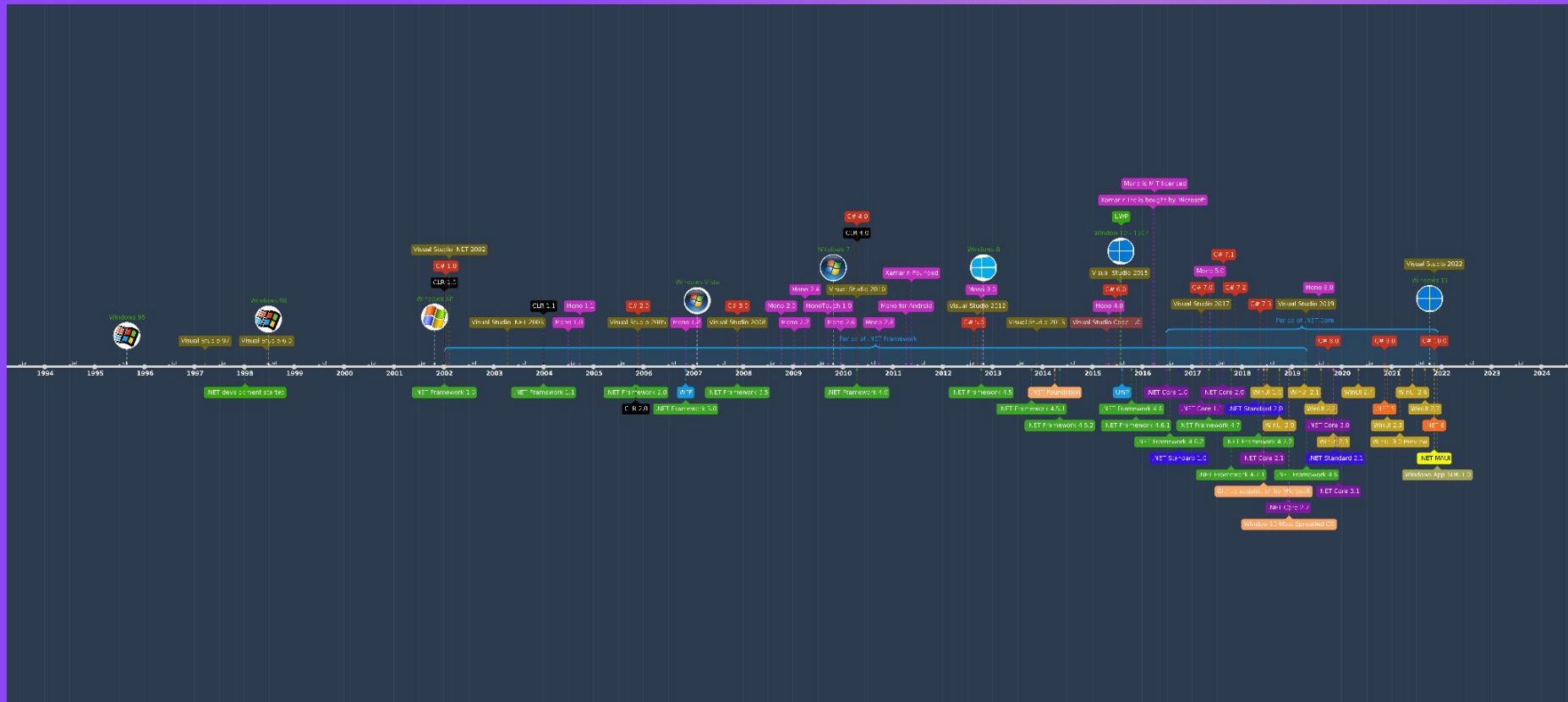
# Платформа .Net



Язык C# — это язык разработки под программную среду .Net.

Программа, написанная на этом языке, может работать там, где установлен .Net. Таким образом, целесообразно начинать погружаться в обучение именно с информации о .Net .







## На каких языках ведется разработка под .Net

- Visual Basic .Net



## На каких языках ведется разработка под .Net

- Visual Basic .Net
- C++/CLI



## На каких языках ведется разработка под .Net

- Visual Basic .Net
- C++/CLI
- F#



## На каких языках ведется разработка под .Net

- Visual Basic .Net
- C++/CLI
- F#
- C#



## Компоненты .Net

- CLR - Common Language Runtime



## Компоненты .Net

- CLR - Common Language Runtime
- FCL- Framework Classes Library



## Из чего состоит FCL

Классы для создания:

- Консольные





## Из чего состоит FCL

Классы для создания:

- Консольные
- Графические



## Из чего состоит FCL

Классы для создания:

- Консольные
- Графические
- Веб-приложения



## Ключевые особенности CLR

- умеет оптимизировать “на лету” приложение под архитектуру конкретной платформы;



## Ключевые особенности CLR

- умеет оптимизировать “на лету” приложение под архитектуру конкретной платформы;
- включает в свой состав “сборщик мусора” — механизм, упрощающий работу с памятью для программистов путем автоматического освобождения памяти переменных, вышедших из зоны видимости;



## Ключевые особенности CLR

- умеет оптимизировать “на лету” приложение под архитектуру конкретной платформы;
- включает в свой состав “сборщик мусора” — механизм, упрощающий работу с памятью для программистов путем автоматического освобождения памяти переменных, вышедших из зоны видимости;
- гарантирует “безопасное выполнение” кода, полученного из сторонних источников: платформа поддерживает строгую типизацию и безопасный доступ к памяти;



## Ключевые особенности CLR

- умеет оптимизировать “на лету” приложение под архитектуру конкретной платформы;
- включает в свой состав “сборщик мусора” — механизм, упрощающий работу с памятью для программистов путем автоматического освобождения памяти переменных, вышедших из зоны видимости;
- гарантирует “безопасное выполнение” кода, полученного из сторонних источников: платформа поддерживает строгую типизацию и безопасный доступ к памяти;
- предоставляет различные службы диагностики, отладки, получения дампов памяти, трассировки и наблюдаемости выполняемого кода (Совокупность этих служб обеспечивает гибкий мониторинг приложений. Тут есть все инструменты позволяющие оптимизировать выполнение вашего кода под конкретную платформу, а также настроить его гибкое взаимодействие с компонентами системы);



## Ключевые особенности CLR

- умеет оптимизировать “на лету” приложение под архитектуру конкретной платформы;
- включает в свой состав “сборщик мусора” — механизм, упрощающий работу с памятью для программистов путем автоматического освобождения памяти переменных, вышедших из зоны видимости;
- гарантирует “безопасное выполнение” кода, полученного из сторонних источников: платформа поддерживает строгую типизацию и безопасный доступ к памяти;
- предоставляет различные службы диагностики, отладки, получения дампов памяти, трассировки и наблюдаемости выполняемого кода (Совокупность этих служб обеспечивает гибкий мониторинг приложений. Тут есть все инструменты позволяющие оптимизировать выполнение вашего кода под конкретную платформу, а также настроить его гибкое взаимодействие с компонентами системы);
- умеет эффективно взаимодействовать с “неуправляемым кодом”.



Управляемый код — код,  
выполняемый под управлением CLR





Неуправляемый код — выполняется  
за пределами CLR



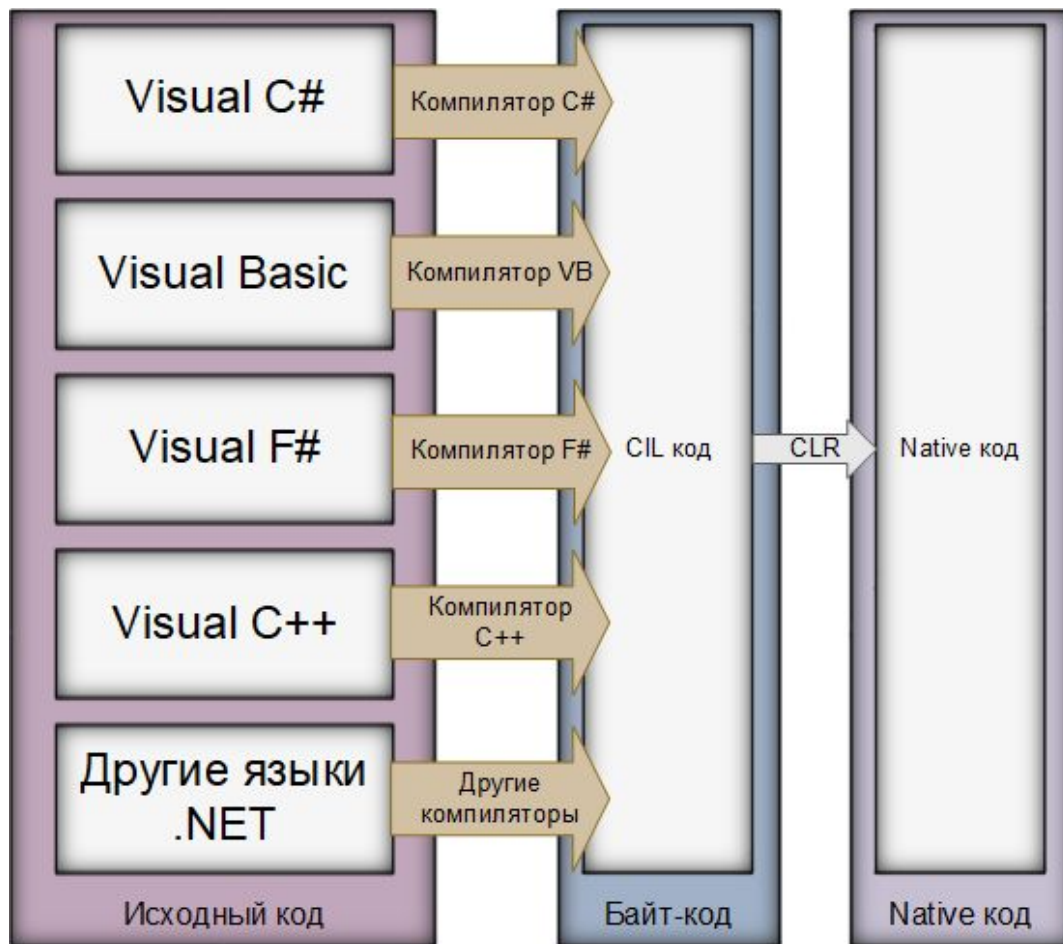
## Сборщик мусора

- Инструмент автоматического освобождения памяти
- Имеет 3 поколения



## IL или CIL

- IL - это Intermediate Language, специальный байт-код, понятный CLR
- Компилятор - преобразует исходных код в код IL





# Инструменты разработки



# Visual studio

Скачать Visual Studio 2022 можно по ссылке

<https://visualstudio.microsoft.com/>

# Visual studio

## Meet the Visual Studio family



### Visual Studio |

The most comprehensive IDE for .NET and C++ developers on Windows. Fully packed with a sweet array of tools and features to elevate and enhance every stage of software development.

[Learn more →](#)

[Download Visual Studio ▾](#)

Community 2022

Professional 2022

Enterprise 2022

Free for individual developers, academic uses, and open source



### Visual Studio for Mac |

A comprehensive IDE for .NET developers that's native to macOS. Includes top-notch support for web, cloud, mobile, and game development.

[Learn more →](#)

[Read more about activating your license](#)

[Download Visual Studio for Mac](#)



### Visual Studio Code |

A standalone source code editor that runs on Windows, macOS, and Linux. The top pick for JavaScript and web developers, with extensions to support just about any programming language.

[Learn more →](#)

By using Visual Studio Code you agree to its [license & privacy statement](#)

[Download Visual Studio Code ▾](#)



## Visual studio

При установке Visual Studio обязательно выберите:

- ASP.Net & web development — разработка web-приложений;
- .Net desktop development — разработка настольных приложений или сервисов.







# Visual studio


Modifying — Visual Studio Community 2022 — 17.0.0


Workloads Individual components Language packs Installation locations

Web & Cloud (4)


**ASP.NET and web development**  
Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...


**Python development**  
Editing, debugging, interactive development and source control for Python.


**Azure development**  
Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework....


**Node.js development**  
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

**Mobile development with .NET**  
Build cross-platform applications for iOS, Android or Windows using Xamarin.

**.NET desktop development**  
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

**Desktop development with C++**  
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

**Universal Windows Platform development**  
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

Location  
C:\Program Files\Microsoft Visual Studio\2022\Community

By continuing, you agree to the [license](#) for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the [3rd Party Notices](#) or in its accompanying license. By continuing, you also agree to those licenses.

Installation details

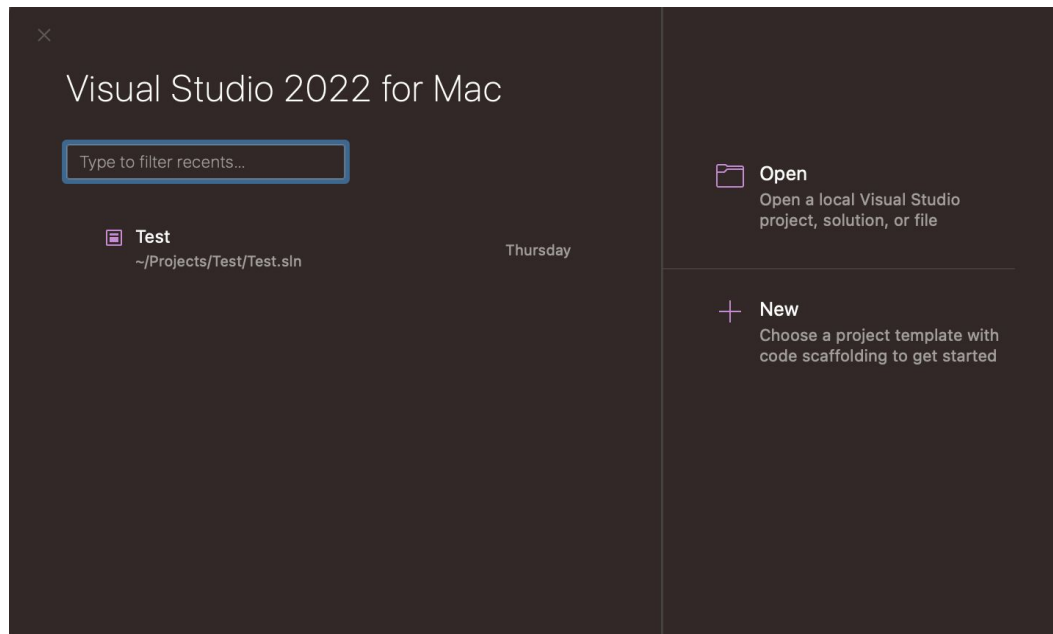
- Visual Studio core editor
- .NET desktop development \*
- ASP.NET and web development \*
  - Included
    - .NET Framework 4.7.2 development tools
    - ASP.NET and web development prerequisi...
  - Optional
    - .NET Framework 4.8 development tools
    - Cloud tools for web development
    - .NET profiling tools
    - Entity Framework 6 tools
    - Live Share
    - .NET Debugging with WSL
    - IntelliCode
    - .NET Framework project and item templates
    - Windows Communication Foundation
    - .NET Framework 4.6.2-4.7.1 development t...
    - Additional project templates (previous ver...
    - .NET WebAssembly build tools

Total space required 1.17 GB

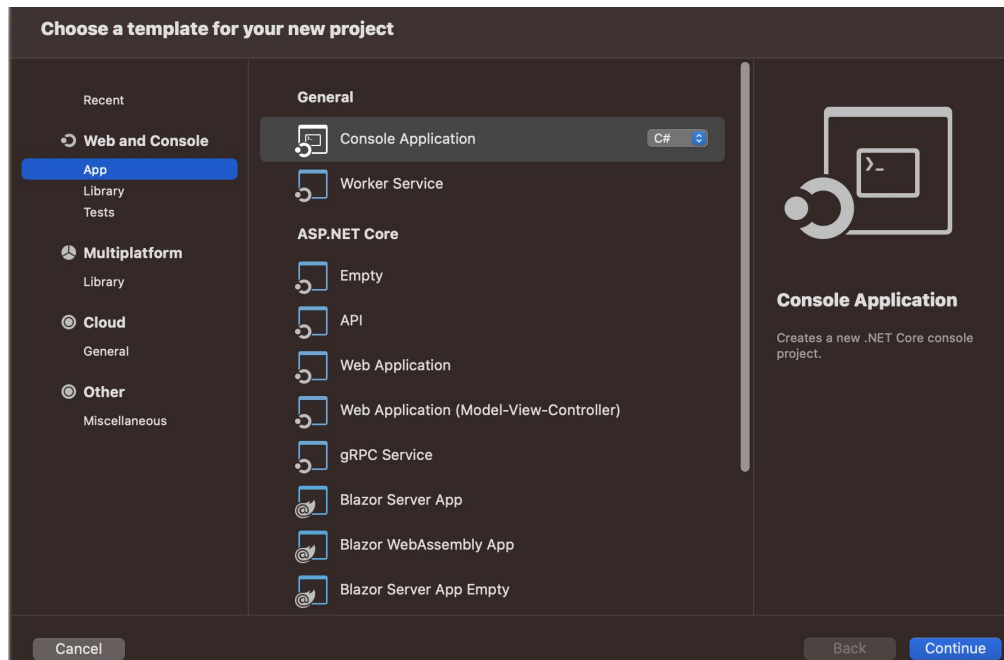
Install while downloading Modify



## Visual studio. Окно запуска



## Visual studio. Шаблоны проектов





# Visual studio

Шаблоны проектов удобно структурированы в группы:

- Web and Console - группа позволяющая выбрать шаблоны проектов для web и desktop разработки
  - App - приложения
  - Library - библиотеки классов
  - Tests - проекты unit-тестирования



# Visual studio

Шаблоны проектов удобно структурированы в группы:

- Multiplatform - шаблоны кроссплатформенных проектов
  - Library - библиотеки классов



# Visual studio

Шаблоны проектов удобно структурированы в группы:

- Cloud - облачные приложения
  - General - проекты для работы с облачной платформой Microsoft Azure

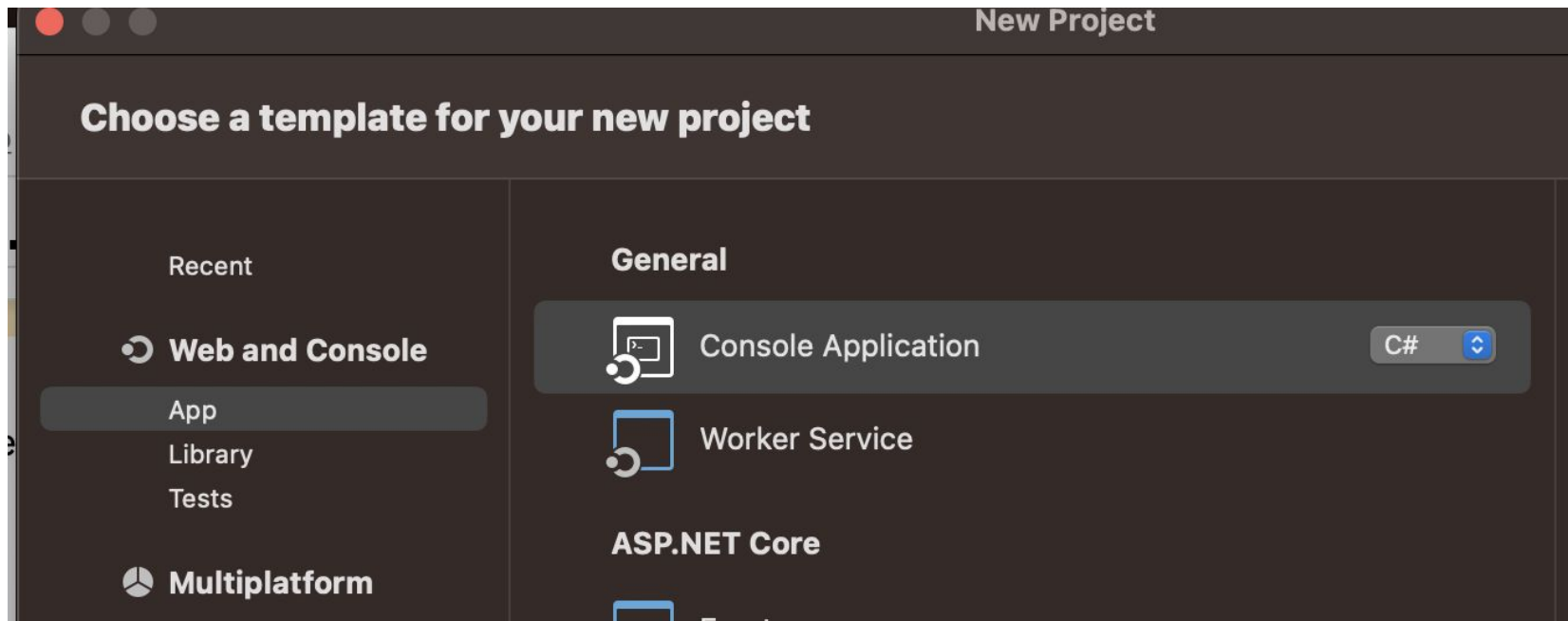


# Visual studio

Шаблоны проектов удобно структурированы в группы:

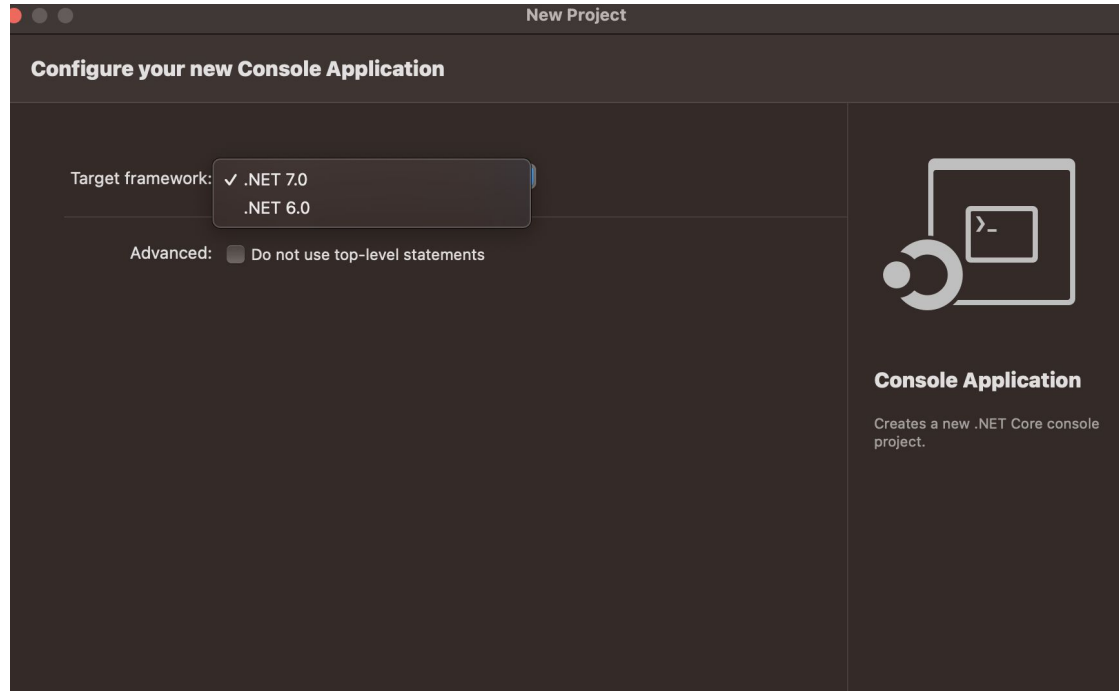
- Other - прочее
  - Miscellaneous - все то, что не вошло в предыдущие группы (если при установке вы выбрали рекомендованные выше опции, то в этой подгруппе будет находиться только проект пустого решения)

## Visual studio - Создание нового проекта





## Visual studio - Создание нового проекта (выбор версии .Net)





## Visual studio - Создание нового проекта (настройка путей и имен)

**Configure your new Console Application**

Project name:

Solution name:

Location:

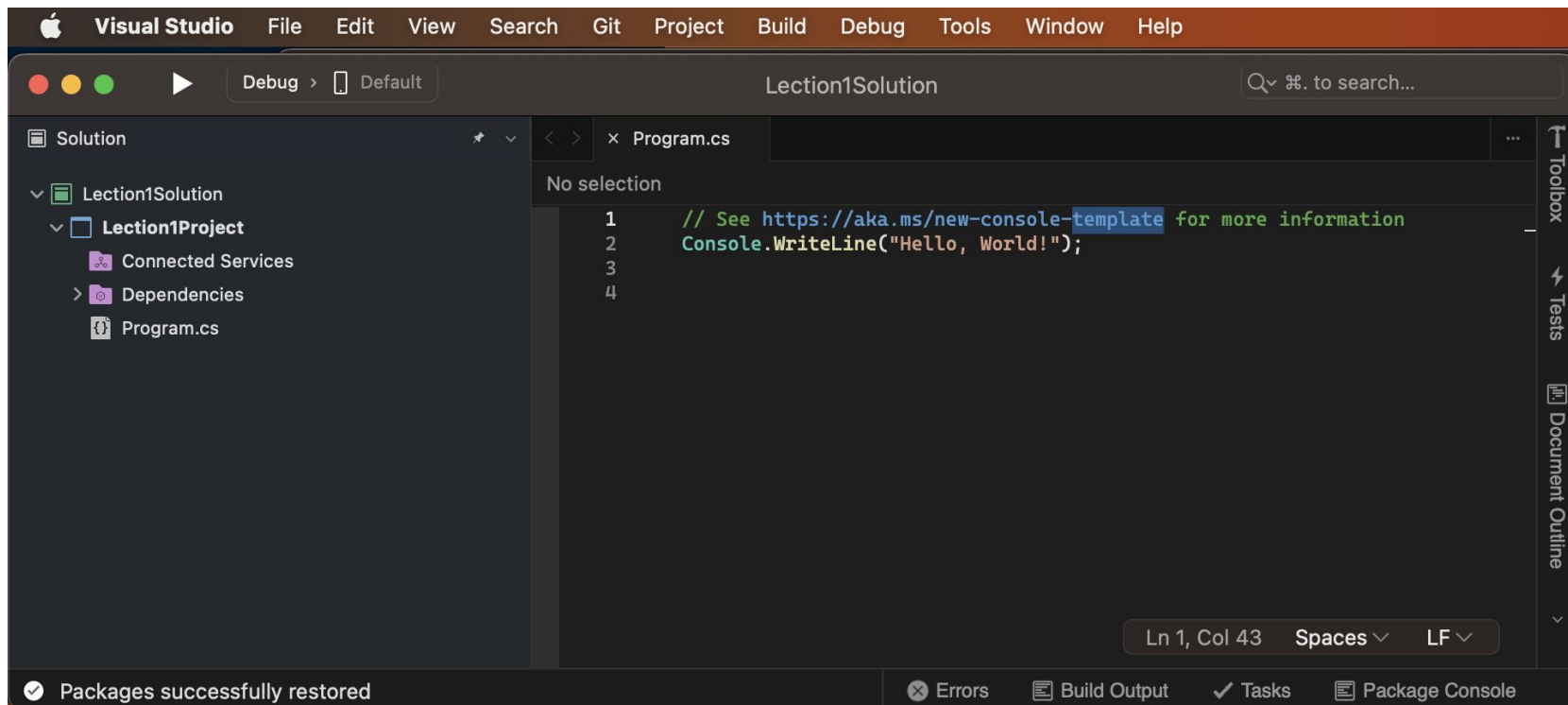
☒ Put project file in a subfolder

Version control: ☐ Use git for version control  
☒ Add .gitignore file for inessential files

**Preview**

- /Users/korlov/Projects
  - Lesson1Solution
    - Lesson1Solution.sln
    - Lesson1Project
      - Lesson1Project.csproj

# Visual studio - Главное окно среды разработки



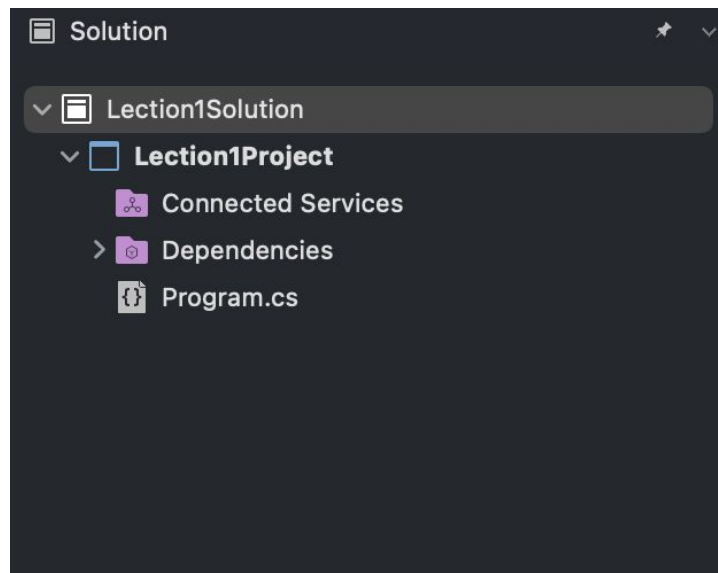


## Visual studio - Редактор кода

```
< > • Program.cs  
No selection  
1 // See https://aka.ms/new-console-template for more information  
2 Console.WriteLine("Hello, World!");  
3  
4  
5
```

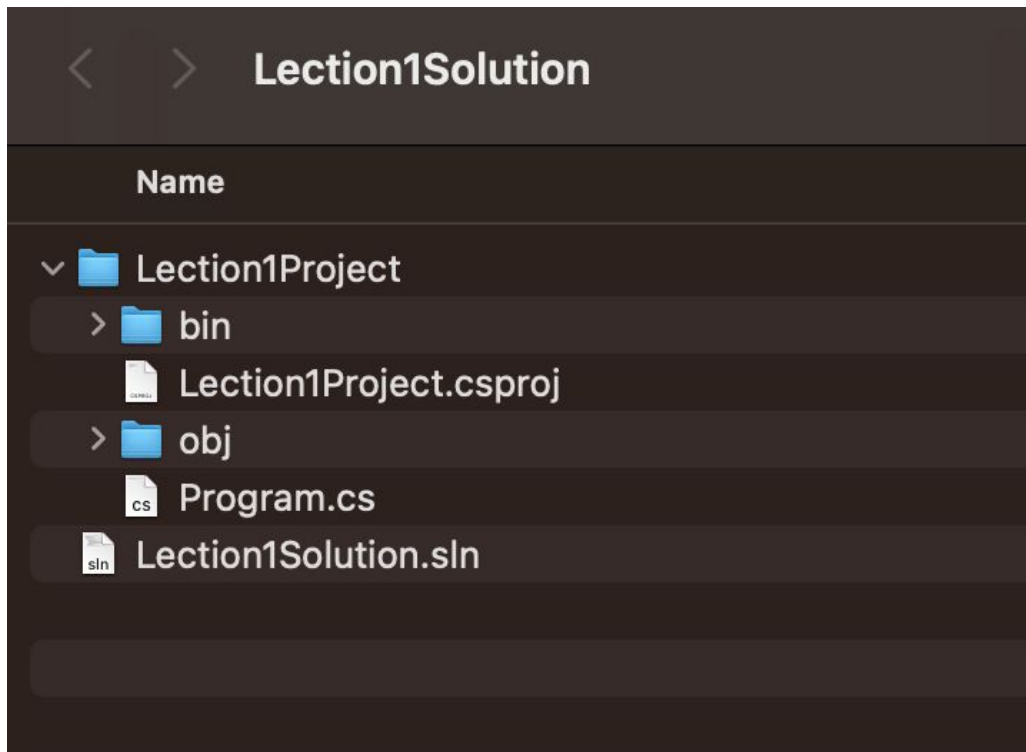


## Visual studio - просмотр структуры проекта





## Visual studio - вид каталога проекта на диске ОС





## Visual studio - полный шаблон консольного приложения

```
Program > Main(string[] args)
1  using System;
2
3  namespace Lection1
4  {
5      internal class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.WriteLine("Hello World!");
10         }
11     }
12
13 }
```



## Visual studio - приложение “приветствие”

```
No selection

1  namespace Lection1
2  {
3      internal class Program
4      {
5          static void Main(string[] args)
6          {
7              if (args.Length == 1)
8              {
9                  string name = args[0];
10                 string message = $"Привет {name}";
11                 Console.WriteLine(message);
12             }
13             else
14             {
15                 Console.WriteLine("Hello World!");
16             }
17         }
18     }
19 }
20 }
```





## Visual studio - запуск и отладка приложения

A screenshot of a terminal window within the Visual Studio IDE. The window title is 'Terminal - Lection1Project'. The terminal displays the text 'Hello, World!' in a monospaced font, with a green cursor block positioned at the start of the line below it.

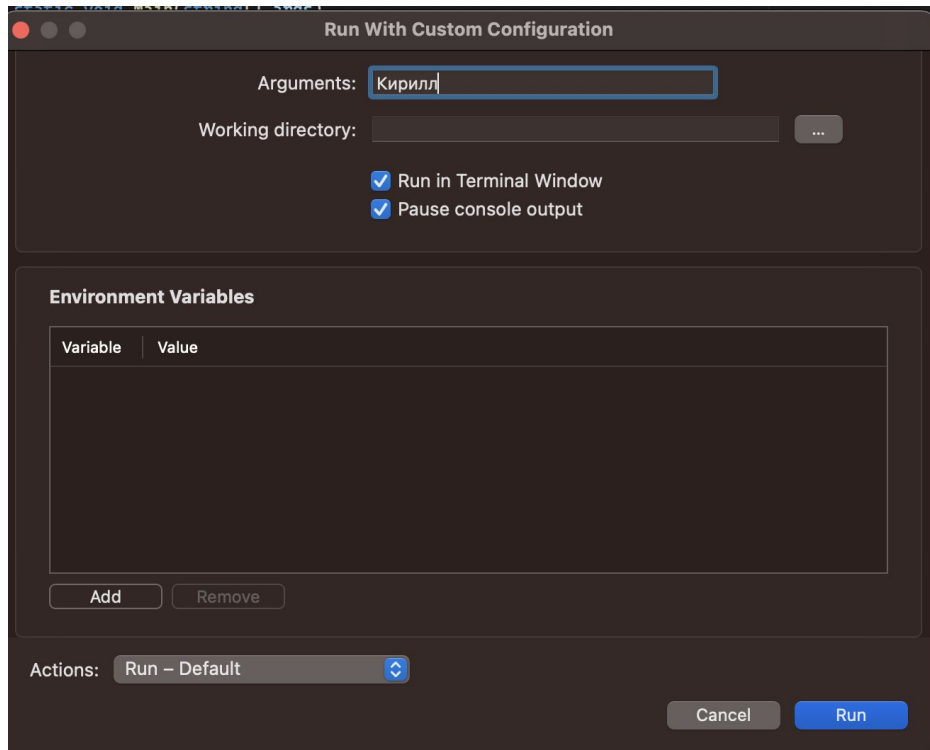
```
>- Terminal - Lection1Project  
  
Hello, World!  
█
```



## Visual studio - запуск приложения с аргументами из командной строки

```
[korlov@Kirills-MacBook-Pro net6.0 % ./Lec1ion1Project  
Hello, World!  
[korlov@Kirills-MacBook-Pro net6.0 % ./Lec1ion1Project Кирилл  
Привет Кирилл!  
korlov@Kirills-MacBook-Pro net6.0 % █
```

## Visual studio - запуск приложения с аргументами из среды





## Visual studio - запуск приложения - результат

 Terminal – Lection1Project

П р и в е т   К и р и л л !

## Visual studio - отладка приложения - точка останова



```
1 namespace MyProject;
2 class Program
3 {
4     static void Main(string[] args)
5     {
6         if (args.Length < 1)
7         {
8             string name = args[0];
9             string message = $"Привет {name}!";
10            Console.WriteLine(message);
11        }
12        else
13            Console.WriteLine("Hello, World!");
14    }
15 }
```

The image shows a Visual Studio code editor with a C# program. A breakpoint is set at line 6, which is the start of the `if` statement. A tooltip is visible over the breakpoint, showing the variable `args.Length` with a value of `0`. The program is a simple `Main` method that checks if `args.Length` is less than 1. If true, it prints a greeting with the first argument; otherwise, it prints "Hello, World!".



## Visual studio - отладка приложения - приложение с ошибкой

```
1 namespace Lektion1
2 {
3     internal class Program
4     {
5         static int DivideTwoNumbers(int n1, int n2)
6         {
7             int res = n1 / n2;
8             return res;
9         }
10
11         static void Main(string[] args)
12         {
13             int resultOfDivision = DivideTwoNumbers(1, 0);
14
15             Console.WriteLine($"Result = ${resultOfDivision}");
16         }
17     }
18 }
19 }
```

The screenshot shows the Visual Studio code editor with a file named Program.cs. The code defines a namespace Lektion1 containing an internal class Program. Inside Program, there is a static method DivideTwoNumbers that takes two integers and returns their division result. The Main method calls DivideTwoNumbers(1, 0), which causes a runtime error (division by zero). A yellow lightbulb icon is visible next to line 11, indicating a warning or error.



Вопрос

Попробуйте самостоятельно найти ошибку!





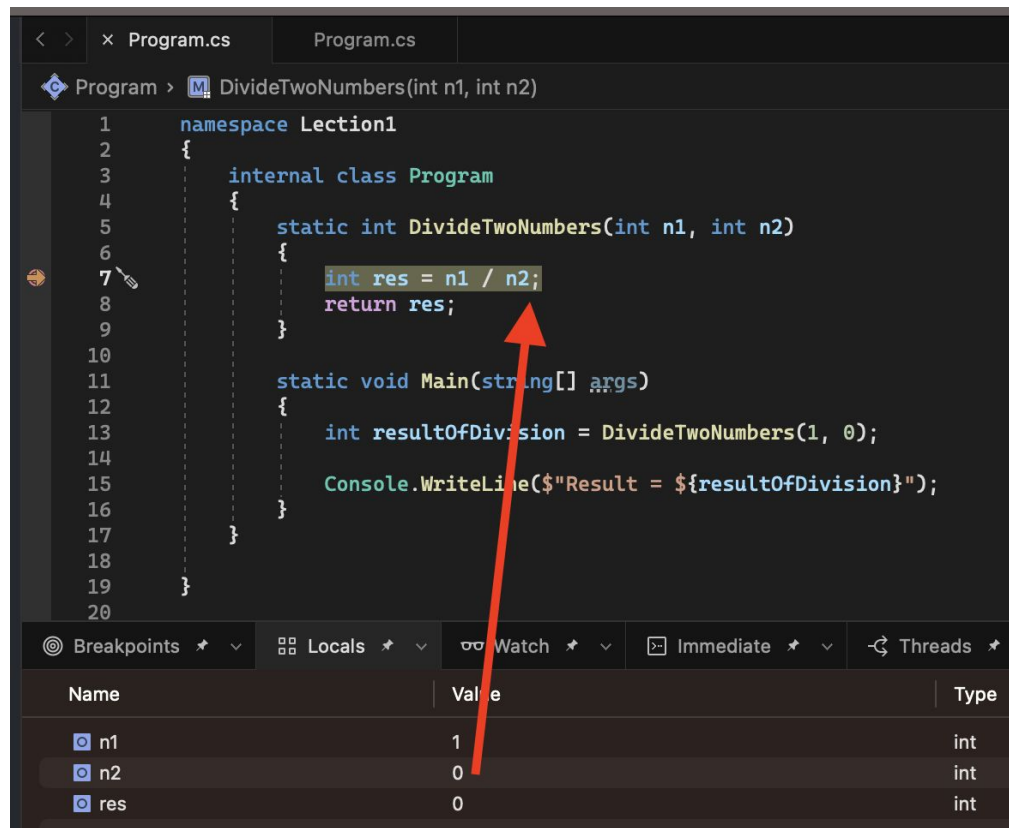
## Visual studio - отладка приложения - запуск с ошибкой

Terminal – Lektion1Program2

```
Unhandled exception. System.DivideByZeroException: Attempted to divide by zero.  
    at Lektion1.Program.DivideTwoNumbers(Int32 n1, Int32 n2) in /Users/korlov/csac/lessons/Lektion1/Lektion1Program2/Program.cs:line 7  
    at Lektion1.Program.Main(String[] args) in /Users/korlov/csac/lessons/Lektion1/Lektion1Program2/Program.cs:line 13  
/bin/bash: line 1: 90708 Abort trap: 6          "/usr/local/share/dotnet/dotnet" "/Users/korlov/csac/lessons/Lektion1/Lektion1Program2/bin/Debug/net7.0/Lektion1Program2.dll"
```



## Visual studio - отладка приложения - запуск с отладкой



Program > M DivideTwoNumbers(int n1, int n2)

```
1 namespace Lecion1
2 {
3     internal class Program
4     {
5         static int DivideTwoNumbers(int n1, int n2)
6         {
7             int res = n1 / n2;
8             return res;
9         }
10
11        static void Main(string[] args)
12        {
13            int resultOfDivision = DivideTwoNumbers(1, 0);
14            Console.WriteLine($"Result = ${resultOfDivision}");
15        }
16    }
17 }
18
19
20
```

Breakpoints Locals Watch Immediate Threads

Name	Value	Type
n1	1	int
n2	0	int
res	0	int



# Стиль написания кода



## Стиль написания кода

```
namespace MyProject;
class Program{
static int MY_function(int n1, int n2){//МОЯ ФУНКЦИЯ
int res = n1 /n2;return res;}
static void Main(string[] args){
int my_____VARIABLE = MY_function(1, 1);//ВЫЗОВ МОЕЙ ФУНКЦИИ И ЗАПИСЬ ЕЕ ЗНАЧЕНИЯ В ПЕРЕМЕННУЮ
Console.WriteLine($"Result = {my_____VARIABLE}");}}
```



## Стиль написания кода

### 1 ошибка — отступы

```
namespace MyProject;
class Program{
static int MY_function(int n1, int n2){//МОЯ ФУНКЦИЯ
int res = n1 /n2;return res;}
static void Main(string[] args){
int my_____VARIABLE = MY_function(1, 1);//ВЫЗОВ МОЕЙ ФУНКЦИИ И ЗАПИСЬ ЕЕ ЗНАЧЕНИЯ В ПЕРЕМЕННУЮ
Console.WriteLine($"Result = {my_____VARIABLE}");}}
```



## Стиль написания кода

2 ошибка — одна строка - одна команда

```
namespace MyProject;
class Program{
static int MY_function(int n1, int n2){//МОЯ ФУНКЦИЯ
int res = n1 /n2;return res;}
static void Main(string[] args){
int my_____VARIABLE = MY_function(1, 1);//ВЫЗОВ МОЕЙ ФУНКЦИИ И ЗАПИСЬ ЕЕ ЗНАЧЕНИЯ В ПЕРЕМЕННУЮ
Console.WriteLine($"Result = {my_____VARIABLE}");}}
```



## Стиль написания кода

**3 ошибка — имена функций и переменных должны отражать их назначение**

```
namespace MyProject;
class Program{
static int MY_function(int n1, int n2){//МОЯ ФУНКЦИЯ
int res = n1 /n2;return res;}
static void Main(string[] args){
int my_____VARIABLE = MY_function(1, 1);//ВЫЗОВ МОЕЙ ФУНКЦИИ И ЗАПИСЬ ЕЕ ЗНАЧЕНИЯ В ПЕРЕМЕННУЮ
Console.WriteLine($"Result = {my_____VARIABLE}");}}
```



## Стиль написания кода

### 4 ошибка — комментарии

```
namespace MyProject;
class Program{
static int MY_function(int n1, int n2){//МОЯ ФУНКЦИЯ
int res = n1 /n2;return res;}
static void Main(string[] args){
int my_____VARIABLE = MY_function(1, 1);//ВЫЗОВ МОЕЙ ФУНКЦИИ И ЗАПИСЬ ЕЕ ЗНАЧЕНИЯ В ПЕРЕМЕННУЮ
Console.WriteLine($"Result = {my_____VARIABLE}");}}
```



## Стиль написания кода

Гайд: [Имена идентификаторов C# | Microsoft Learn](#)





## Стиль написания кода

✓ Стиль кода

имена идентификаторов C#

Соглашения о написании кода на C#

- Име
- Типь
- Типь
- явля


На этой лекции вы:

- узнали о платформе .Net, её истории и версиях;
- разобрались с установкой и настройкой среды разработки Visual Studio;
- научились создавать собственные проекты из представленных средой шаблонов и изучили, из чего они состоят;
- написали первое приложение и освоили инструмент “отладчик”;
- получили понимание о стиле написания кода.



## Домашнее задание

- Установите Visual Studio.
- Попробуйте написать примеры приложений из нашего урока (Приветствие).
- Изучите стиль написания кода на официальном сайте Microsoft по адресу:  
[Имена идентификаторов C# | Microsoft Learn](#)

**Спасибо**   
**за внимание**

