# Introduction to programming in c#

## About me

Piotr Kątny

Software Engineer and Technical Lead at SII Poland

p.katny@gmail.com

## Course Contents

- Introduction to C# and .NET
- Working with projects in Visual Studio
- Variables and expressions in c#
- Operations
- Collections introduction
- Console applications
- Classes and objects
- String operations
- Exception handling

# Plan for today

- **Introduction to C# and .NET**
- **Working with projects in Visual Studio**
- **Variables and expressions in c#**
- Operations
- Collections introduction
- Console applications
- Classes and objects
- String operations
- Exception handling

“There are only two kinds of programming languages: those people always bitch about and those nobody uses.

*Bjarne Stroustrup*

**C#** is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework.

## So what is .NET Framework?

Part of operating system that includes a virtual execution system called the common language runtime (CLR) and a unified set of class libraries

## Several mysterious, yet important terms related to .NET Framework

- **CLR** – Microsoft’s implementation of CLI
- **CLI** – Common Language Infrastructure – international standard for creating environments to develop and run apps in different languages
- **IL** – Intermediate Language – output from compilation of c# source code
- **Assembly** – executable file that consists of IL code, resources, images, etc.
- **Manifest** – contains information about the assembly's types, version, culture, and security requirements

## Execution of .NET Application

- The **assembly** is loaded into the **CLR**, which might take various actions based on the information in the **manifest**. Then, if the security requirements are met, the **CLR** performs just in time (**JIT**) compilation to convert the **IL code** to native machine instructions.

# .NET Framework vs .Net Core

**.Net Framework**

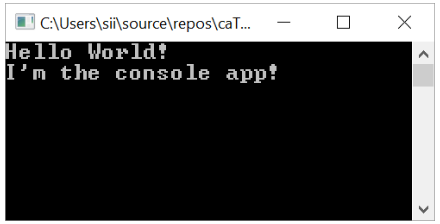
- Current version: 4.7.2
- First release of 1.0 in 2002
- „mature” and „stable”
- Many external packages available
- The only choice for WPF (is it?)
- Many existing systems

**.Net Core**

- Current version: 2.1
- First release: 27 June 2016
- Can be run on multiple platforms (Windows, Unix, MacOS)
- Higher performance
- Much smaller output
- **Open source**

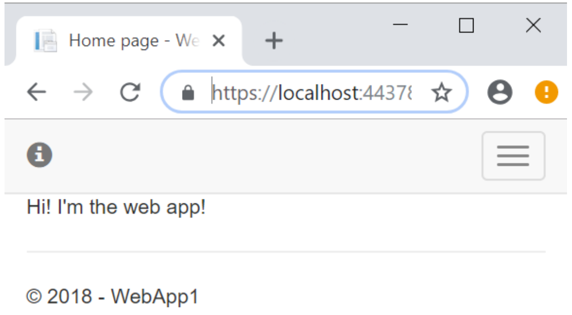
# Application types and examples

- Console



```
Hello World!
I'm the console app!
```

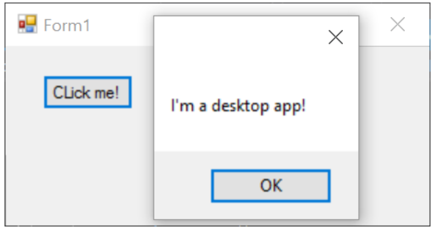
- Web



Hi! I'm the web app!

© 2018 - WebApp1

- Desktop



I'm a desktop app!

# Questions

1. What is the basic difference between c# and .NET?
2. What is the difference between console, desktop and web app?
3. When to choose NET Core and when .NET Framework?

## Working with projects in Visual Studio

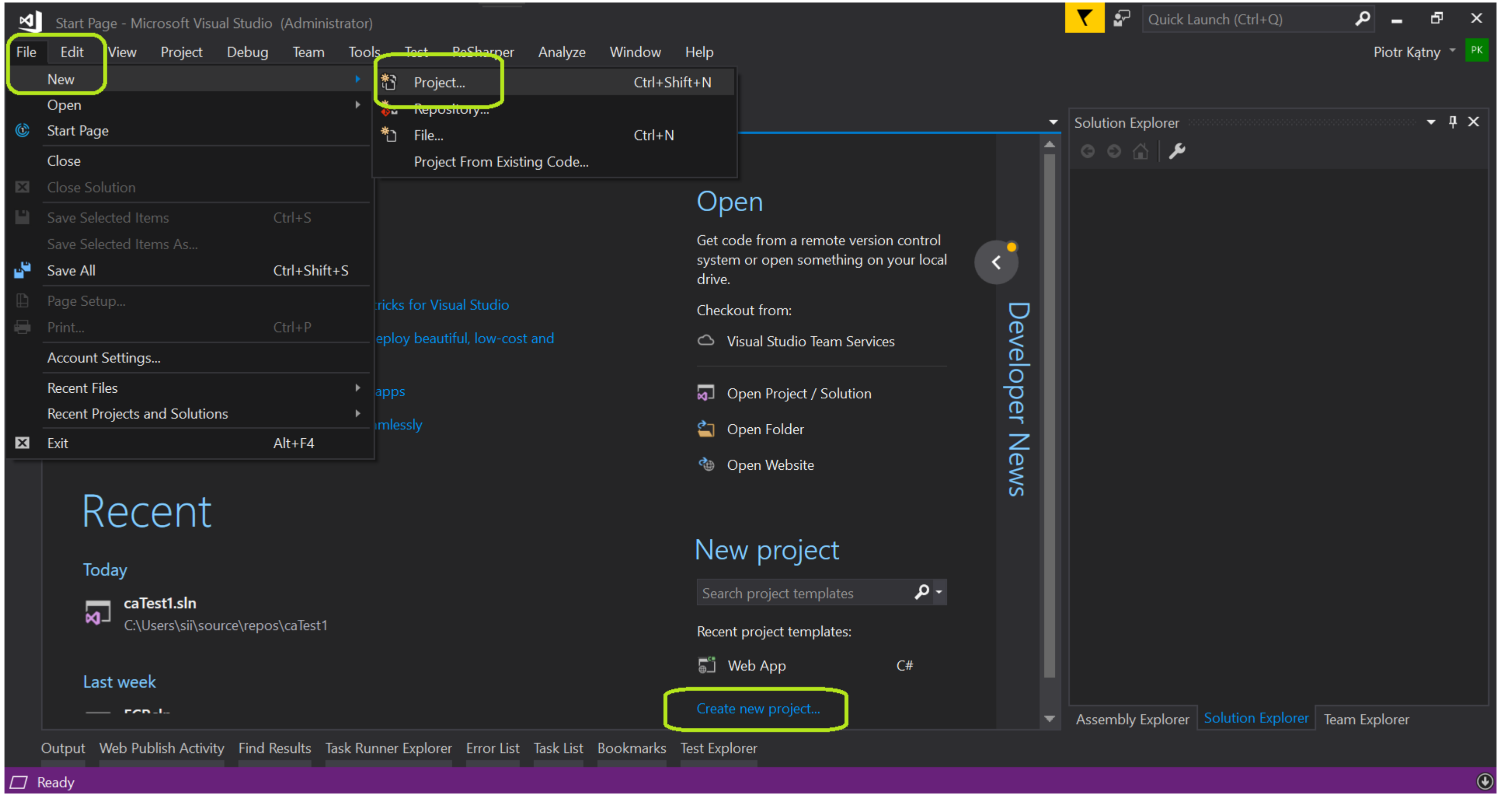
## Project organization in Visual Studio

- Solution
- Project
- File

## Creating a project

- Project templates and boilerplate code
- Project settings on creation and later modification
- Directory structure on disk
- Simplest c# program and its structure

# Creating a project



Start Page - Microsoft Visual Studio (Administrator)

Quick Launch (Ctrl+Q)

File Edit View Project Debug Team Tools Test ReSharper Analyze Window Help

Piotr Kątny

| | |
|---|---|
| New | |
| Open | |
| Start Page | |
| Close | |
| Close Solution | |
| Save Selected Items | Ctrl+S |
| Save Selected Items As... | |
| Save All | Ctrl+Shift+S |
| Page Setup... | |
| Print... | Ctrl+P |
| Account Settings... | |
| Recent Files | |
| Recent Projects and Solutions | |
| Exit | Alt+F4 |

| | |
|---|---|
| Project... | Ctrl+Shift+N |
| Repository... | |
| File... | Ctrl+N |
| Project From Existing Code... | |

Solution Explorer

Open

Get code from a remote version control system or open something on your local drive.

Checkout from:

Visual Studio Team Services

Open Project / Solution

Open Folder

Open Website

Developer News

New project

Search project templates

Recent project templates:

Web App C#

Create new project...

Recent

Today

caTest1.sln
C:\Users\sii\source\repos\caTest1

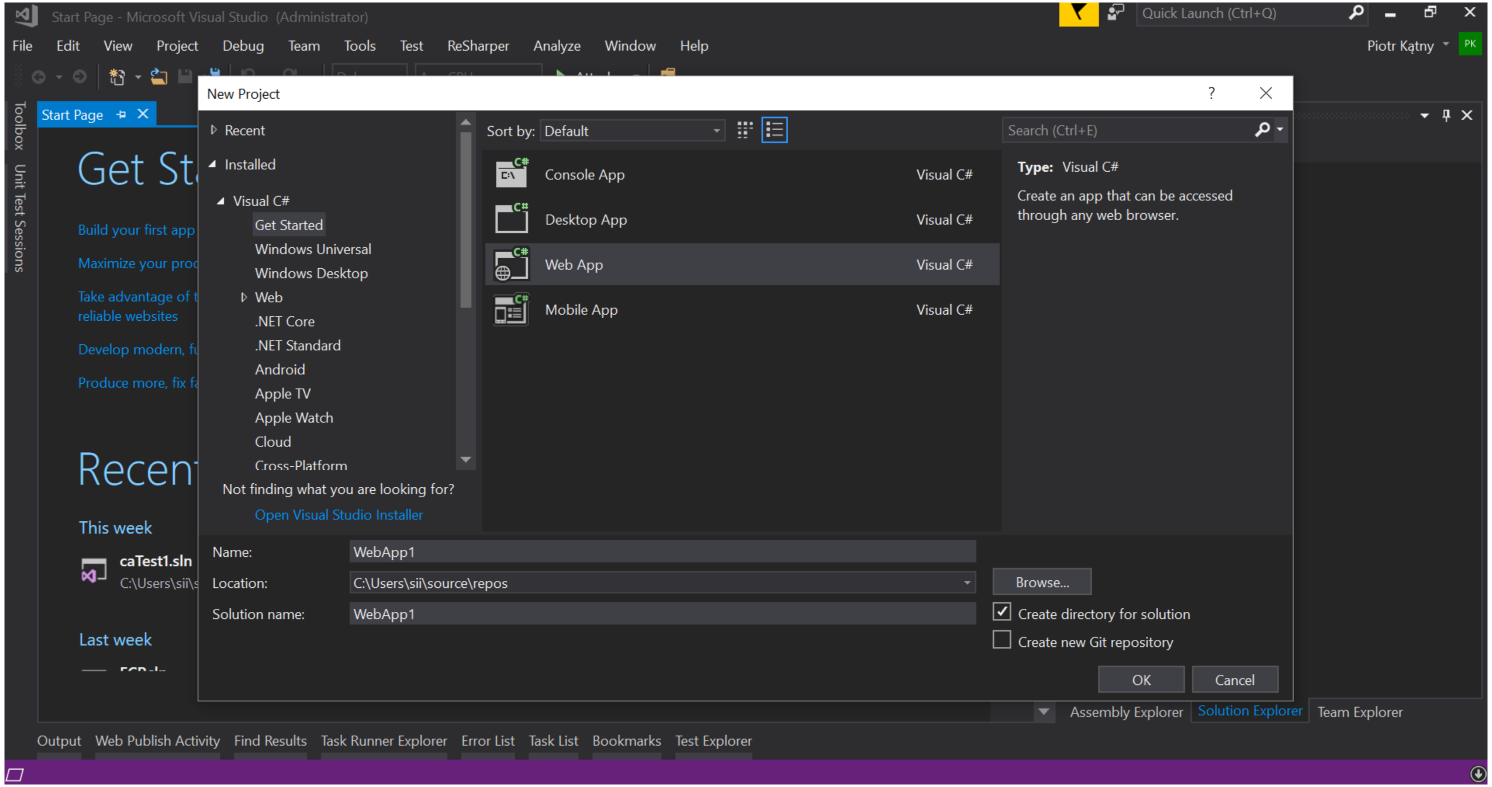
Last week

Assembly Explorer Solution Explorer Team Explorer

Output Web Publish Activity Find Results Task Runner Explorer Error List Task List Bookmarks Test Explorer

Ready

# Creating a project



Start Page - Microsoft Visual Studio (Administrator)
Quick Launch (Ctrl+Q)
File Edit View Project Debug Team Tools Test ReSharper Analyze Window Help
Piotr Kątny
Start Page
Get St
Build your first app
Maximize your prod
Take advantage of t
reliable websites
Develop modern, fu
Produce more, fix fa
Recent
This week
caTest1.sln
C:\Users\sii\s
Last week
Output Web Publish Activity Find Results Task Runner Explorer Error List Task List Bookmarks Test Explorer
Assembly Explorer Solution Explorer Team Explorer

New Project

- Recent
- Installed
  - Visual C#
    - Get Started
    - Windows Universal
    - Windows Desktop
    - Web
    - .NET Core
    - .NET Standard
    - Android
    - Apple TV
    - Apple Watch
    - Cloud
    - Cross-Platform

Not finding what you are looking for?
Open Visual Studio Installer

Sort by: Default

Search (Ctrl+E)

| | |
|---|---|
| Console App | Visual C# |
| Desktop App | Visual C# |
| Web App | Visual C# |
| Mobile App | Visual C# |

**Type:** Visual C#

Create an app that can be accessed through any web browser.

Name: WebApp1
Location: C:\Users\sii\source\repos
Solution name: WebApp1

Browse...

☑ Create directory for solution
☐ Create new Git repository

OK Cancel

## Exercise 1

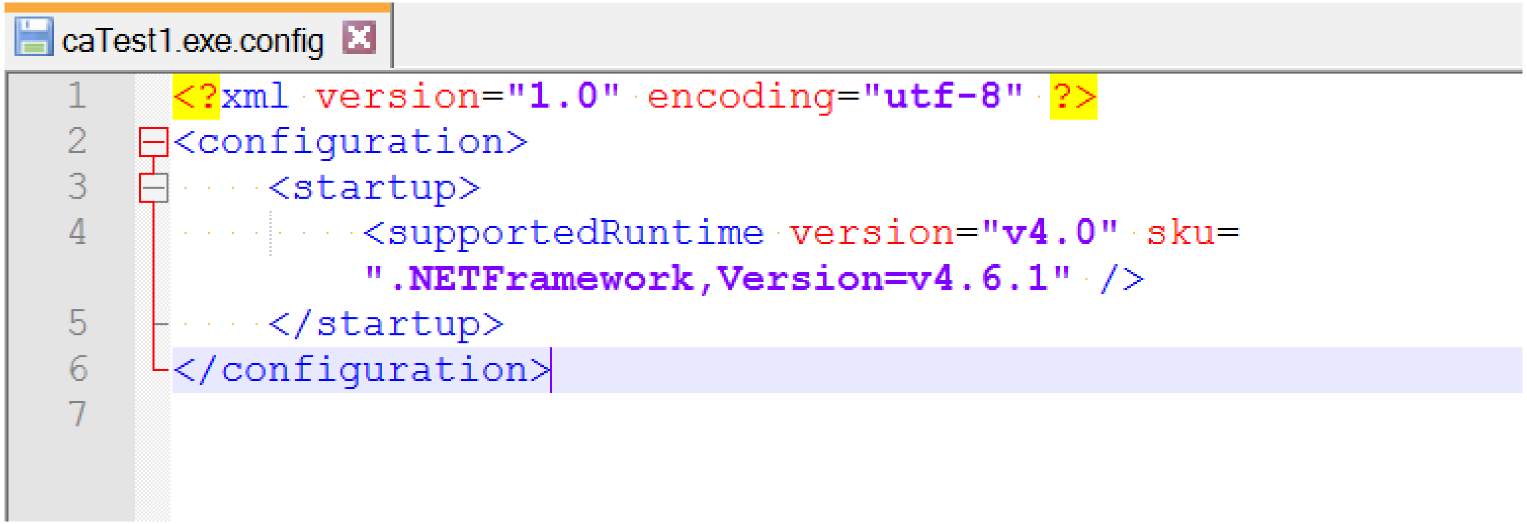
0. Open Visual Studio

1. Create a c# project of type „Console App” from Visual C# -> Get Started
2. Give it some meaningful name („MyFirstApp” is acceptable ☺ )
3. Inspect created solution

## Exercise 1 – follow-up

1. Open project location on disk
2. Open *.csproj file in text editor
3. Add a new file of type „class” to the project
4. Reload *.csproj file in text editor
5. Go to bin/debug folder of your project on disk
6. Open „*.exe.config” file in text editor
7. Inspect content of the file

## Exercise 1 (continuation)


caTest1.exe.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <startup>
        <supportedRuntime version="v4.0" sku=
        ".NETFramework,Version=v4.6.1" />
    </startup>
</configuration>
```

## Creating a project from command line

- ***dotnet.exe*** command
- *dotnet new console -o ConsoleAppFromCommandLine*
- *dotnet run ConsoleAppFromCommandLine*

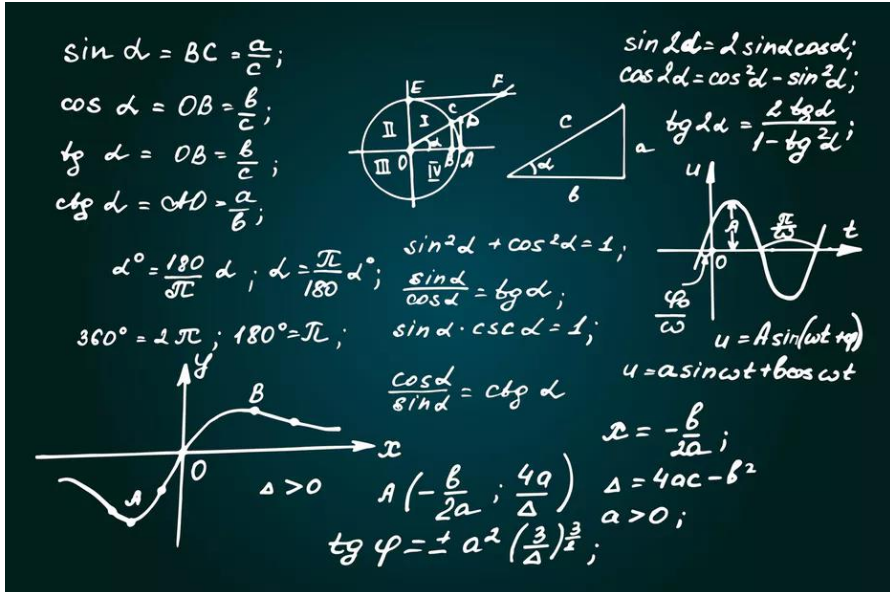
## Alternative tools to write code

- *Visual Studio Code*
- Notepad ☺

## Variables and expressions

## Variables and expressions

- „**Computer**” comes from ***computing***
- We have information and we process it
- Everything is mathematics – but we don’t have to be affraid of it !



## Variables and expressions

- JPEG mathematical formula:

$$G_{u,v} = \alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} g_{x,y}\cos\left[\frac{\pi}{8}\left(x+\frac{1}{2}\right)u\right]\cos\left[\frac{\pi}{8}\left(y+\frac{1}{2}\right)v\right]$$

$$\alpha_p(n) = \begin{cases} \sqrt{\frac{1}{8}}, & \text{if } n = 0 \\ \sqrt{\frac{2}{8}}, & \text{otherwise} \end{cases}$$

Applied on some bytes on disk becomes:

## Variables and expressions

But to display it as a programmer we may just need something like: *

```
Image carImage = Image.FromFile("c:/images/car.jpeg");
DrawImage(carImage, new RectangleF(10, 10, carImage.Width/2, MyImg.Height/2));
```

(* example)

# Variables

A ***variable*** is a „space” in memory, it has a certain type and can have a value that may change during execution of a program.

## Variables

**Examples:**

- int x;
- double y;
- string someText;
- bool isPositive;

## Variables

**Examples with initialization:**

- int a = 147;
- double b = 2.3d;
- string name = „Pan Szakal”;
- bool isWeekend = true;

## Variables

**Implictit variables:**

- var implicitVar = 2.1m;  // m => decimal number
- var implicitVar2 = "I am text"; // string

## Variable names

- Rule number 1: make it meaningful

  `int fdg;` ☹ `int numberOfStudents;` ☺

- Casing

  - `camelCase`
  - `PascalCase`
  - `snake_case`
  - `objCar` - Hungarian notation – do not use! ☺

## Variable names

- Keywords – cannot be use as variable name (with exceptions):
  - Variable types : `int, double, bool, char, string, var, dynamic`
  - Loops: `for, foreach, while` (more in next meeting)
  - Conditionals: `if, switch`
  - Misc: `null, async, await`

## Using Variables

- `y = x +7;`
- `isPositive = (x > 0);`
- `someText = "example text";`
- `otherText = someText + „ another word or four”;`

Statements on the right side of `=` produce a **value** and are called ***expressions***

A **value** from **expression** can be assigned to **variable** (but doesn’t have to be!)

## Types of operations

- Assignment, ie: x = 2;  y = z;
- Comparison, ie: x > y;
- Arithmetic operations, ie: a = b * c;
- Logical conditions, ie: if(a < 5) { … }

## Exercise 2

1. Open console project from previous exercise or create a new one
2. Create several number variables of different types and „play” with them. Put the result on the screen using

   `Console.WriteLine(result);`

   Where `result` is the value of the expression containing some arithmetical operation

## Mixed type operations

By ***default*** we can only do mathematical operations on variables of the same type… but not always

## Mixed type operations

But what if… ?

+ cat = ?

## Mixed type operations

Let’s check in VS!

# Questions

1. What is a variable?
2. What is an expression in c#?

## Mixed type operations

See you next time!

# Thanks!

You can find me at:
p.katny@gmail.com
#Piotr Kątny