Introduction to .NET

Florin Olariu & Andrei Arusoaie
“Alexandru Ioan Cuza”, University of Iași
Department of Computer Science

Organization

- People: Florin Olariu & Andrei Arusoaie
- 14 weeks (full activity) + 2 weeks for evaluation
- Final grade = round(40% * Exam + 60% * Lab + 10% * Kata's)
- Exam: max 40 points, midterm exam
- Lab: max 70 points = 20 lab activity + 40 semester project + 10 coding kata's
  - * bonus points!
- Criteria: min 50 (Exam + Lab)
- WEB: http://profs.info.uaic.ro/~arusoaie.andrei/lectures/NET/DotNet.html

Resources

- Books/movies:
  - https://www.asp.net/ (create your Microsoft account!)
  - Ask for books using Facebook
- Microsoft:
  - https://www.microsoft.com/net/core/platform
- Getting started:
  - https://www.microsoft.com/net/tutorials/csharp/getting-started

Questions?

- Contact:
  - andrei.arusoaie@gmail.com
  - arusoaie.andrei@info.uaic.ro
  - florin@florinolariu.ro
  - olariu@gmail.com
  - florin.olariu@centric.eu

- Facebook:
  - Introduction to .NET (click!)

Motivation

- Why C#?
  - Open source
  - The platform is new (written from scratch)
  - Is the first time when we can say: “Write once => deploy everywhere!”
  - It has been oriented to performance
  - It has a very strong middleware system built-in
  - According with the trends it is the most recommended high-level language that should be learned in 2017-2020

Goal

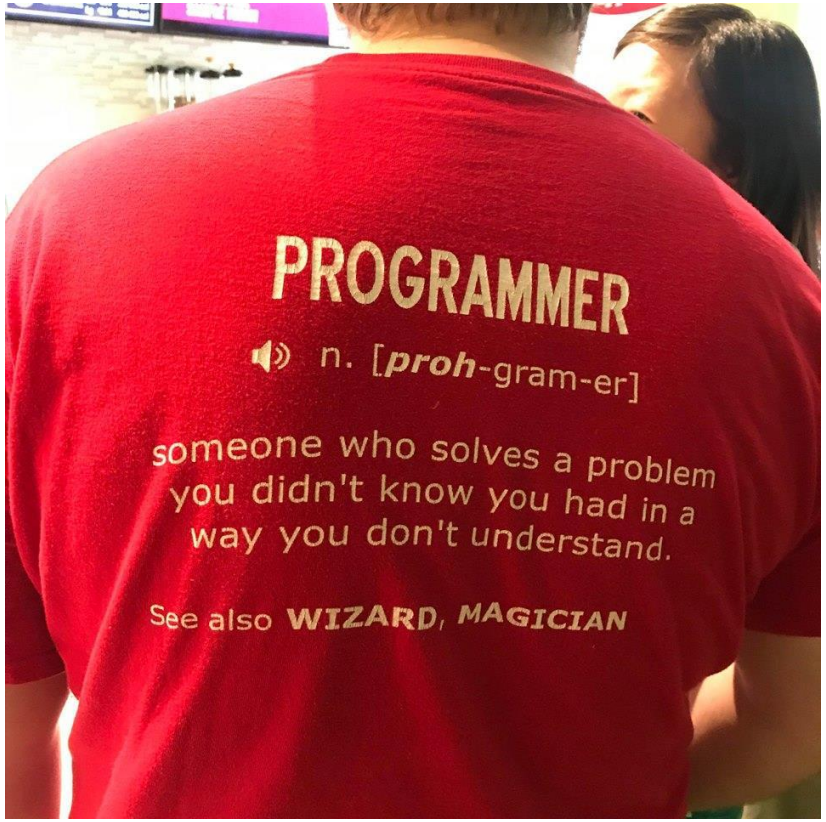
- Learn how to:
  - *Write business applications using .NET Core 2.1*
  - *Using best practices in writing maintainable software*
  - *How to use design patterns*
  - *How to use architectural patterns*
  - *How to write decoupled code*
  - *How to build responsive web applications*
  - *How to work in teams using Agile Development (SCRUM approach)*
  - *How to prepare for an interview .NET oriented*
  - *(Andrei&Florin)Open for new challenges according with your(and market) needs*

Agenda

- What is .NET Core? What is .NET Framework?
- Overview on ASP.NET Core
- What is new about this framework?
- Sample: Developing sample web application.
- OOP in .NET Core
- NET Framework – C# access modifiers and other common types

Before starting …

Before starting …



Before starting …

- **“Being a software developer is awesome. Look around you. Almost everything that is man-made today has software behind it.”**

Before starting …

- **“Every profession has a career ladder. Software development is no different: we start at the bottom and move upward as we become more experienced. Climbing up the software development ladder does not mean becoming a manager or an architect. This is not a career progression; it is a career change. The skills needed to become a great manager or architect are not necessarily the same ones needed to become a great developer (and vice versa). Developers who, for one reason or another, decide to take roles as managers or architects are not climbing up the software development ladder; they are switching ladders.”**

What is .NET Core? What is .NET Framework?

What is .NET Core? What is .NET Framework?

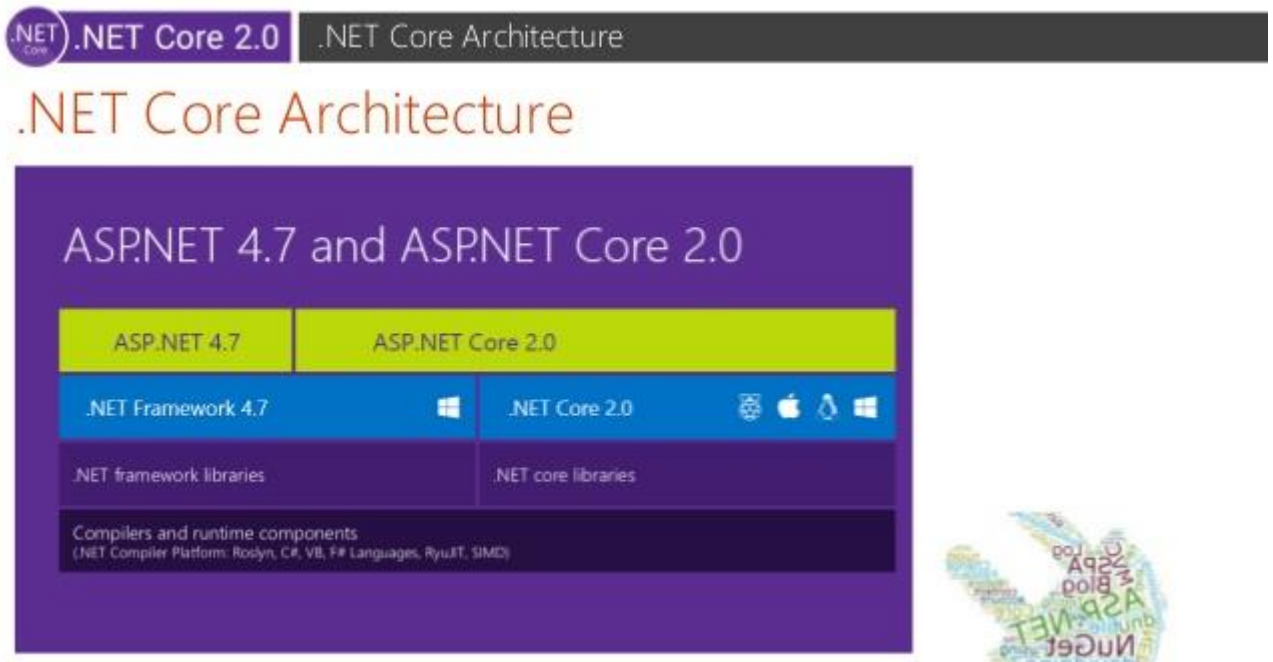
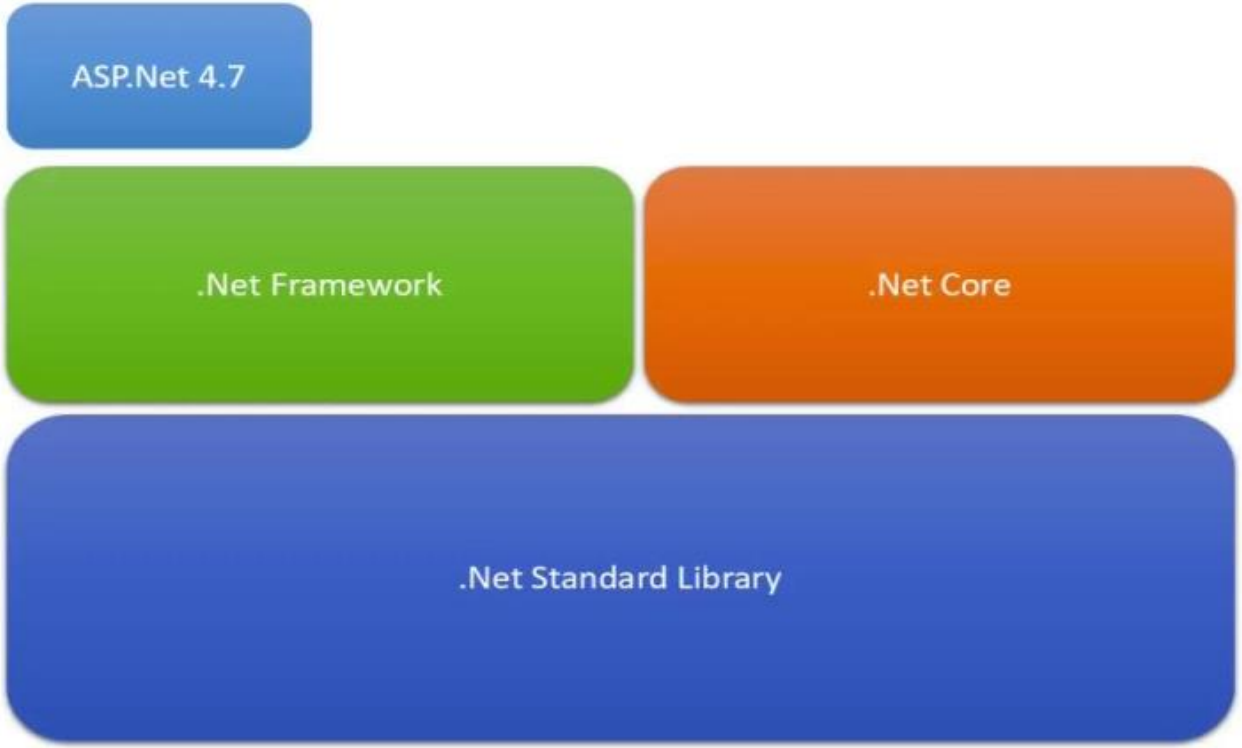


Image source: https://blogs.msdn.microsoft.com

Overview on ASP.NET Core

Overview on ASP.NET Core



What is new about this framework?

What is new about this framework?

1. **Cross platform**

What is new about this framework?

1. **Cross platform**
2. **Open source**

What is new about this framework?

1. **Cross platform**
2. **Open source**
3. **Optimized .NET runtime & libraries**

What is new about this framework?

1. **Cross platform**
2. **Open source**
3. **Optimized .NET runtime & libraries**
4. **Completely modular**

What is new about this framework?

1. Cross platform
2. Open source
3. Optimized .NET runtime & libraries
4. Completely modular
5. Introduction to CLI

What is new about this framework?

1. Cross platform
2. Open source
3. Optimized .NET runtime & libraries
4. Completely modular
5. Introduction to CLI
6. Cloud ready environment

Sample: Developing sample web application

- DEMO

OOP in .NET Core

Inheritance

OOP in .NET Core – inheritance

```
3 references
public class Employee
{
    // ...
}

0 references
public class Architect : Employee
{
    // ...
}

0 references
public class TechLead : Employee
{
    // ..
}

0 references
public class Developer : Employee
{
    // ...
}
```

OOP in .NET Core - inheritance

```
3 references
public class Employee
{
    0 references
    public Guid Id { get; set; }

    1 reference
    public string FirstName { get; set; }

    1 reference
    public string LastName { get; set; }

    0 references
    public string FullName...

    0 references
    public virtual void Loyalty()...
}
```

OOP in .NET Core - polymorphism

OOP in .NET Core - polymorphism

```
public class Employee
{
    0 references
    public Guid Id { get; set; }

    1 reference
    public string FirstName { get; set; }

    1 reference
    public string LastName { get; set; }

    0 references
    public string FullName...

    2 references
    public virtual void Loyalty()...
}

0 references
public class Architect : Employee
{
    2 references
    public override void Loyalty()...
}
```

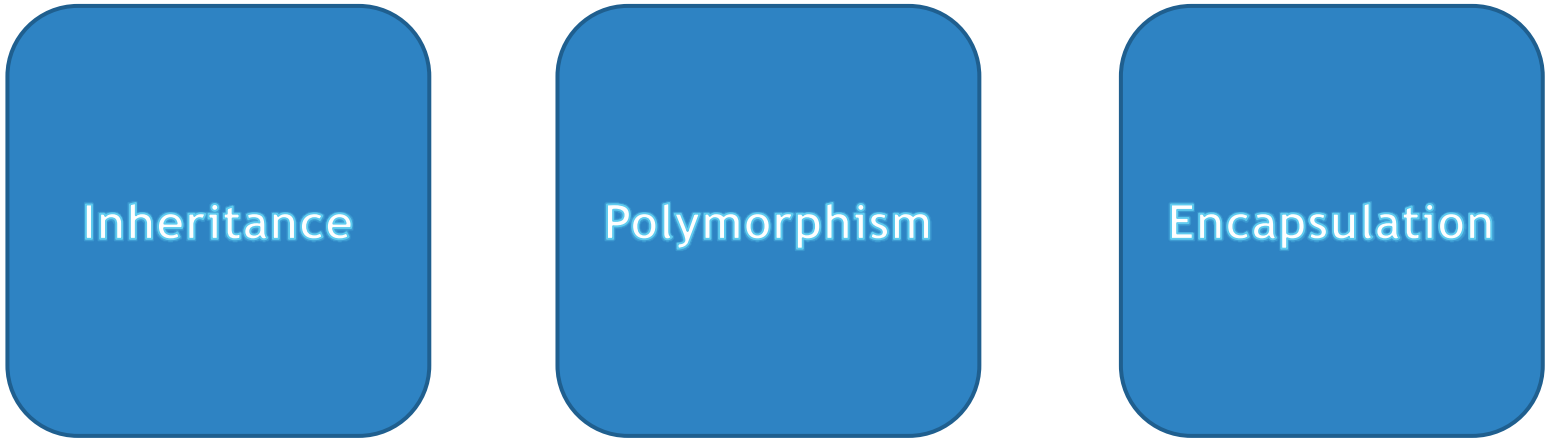
OOP in .NET Core - polymorphism

```
0 references
public double Add(double a, double b)
{
    return a + b;
}

0 references
public double Add(double a, double b, double c)
{
    return a + b + c;
}
```

OOP in .NET Core - encapsulation

OOP in .NET Core



NET Framework – C# access modifiers and other common types

NET Framework – C# access modifiers and other common types

- What is an access modifier?

NET Framework – C# access modifiers and other common types

- What is an access modifier?
  - ***An access modifier is a keyword used to specify the accessibility of a member or a type.***

NET Framework – C# access modifiers and other common types

- How many access modifiers exists in C#?

NET Framework – C# access modifiers and other common types

- How many access modifiers exists in C#?
  - In C# we have four access modifiers
    - **public**
    - **protected**
    - **internal**
    - **private**

NET Framework – C# access modifiers and other common types

- How many access modifiers exists in C#?
  - In C# we have four access modifiers
    - **public**
    - protected
    - internal
    - private

NET Framework – C# access modifiers and other common types

- How many access modifiers exists in C#?
  - In C# we have four access modifiers
    - **public**
    - protecte
    - internal
    - p

public => there is no restriction in regards with accessibility

NET Framework – C# access modifiers and other common types

- How many access modifiers exists in C#?
  - In C# we have four access modifiers
    - public
    - **protected**
    - internal
    - private

NET Framework – C# access modifiers and other common types

- How many access modifiers exists in C#?
  - In C# we have four access modifiers
    - public
    - **protected**
    - internal
    - private

protected => access is limited to the containing class or types derived from the containing class

NET Framework – C# access modifiers and other common types

- How many access modifiers exists in C#?
  - In C# we have four access modifiers
    - public
    - protected
    - **internal**
    - private

NET Framework – C# access modifiers and other common types

- How many access modifiers exists in C#?
  - In C# we have four access modifiers
    - public
    - protected
    - **internal**
    - private

internal => access is limited to the current assembly

NET Framework – C# access modifiers and other common types

- How many access modifiers exists in C#?
  - In C# we have four access modifiers
    - public
    - protected
    - internal
    - **private**

NET Framework – C# access modifiers and other common types

- How many access modifiers exists in C#?
  - In C# we have four access modifiers
    - public
    - protected
    - internal
    - **private**

private => access is limited to the containing type

NET Framework – C# access modifiers and other common types

- Other common types

NET Framework – C# access modifiers and other common types

- Other common types
  - **readonly vs const**
  - **sealed**
  - **virtual**

NET Framework – C# access modifiers and other common types

- Other common types
  - **readonly vs const**
  - sealed
  - virtual

NET Framework – C# access modifiers and other common types

- Other common types
  - **readonly vs const**
  - se
  - vi

readonly	const

NET Framework – C# access modifiers and other common types

- Other common types
  - **readonly vs const**
  - se
  - vi

readonly	const
Can be initialized: at declaration or in a constructor	

NET Framework – C# access modifiers and other common types

- Other common types
  - **readonly vs const**
  - se
  - vi

readonly	const
Can be initialized: at declaration or in a constructor	Can be initialized **only at the declaration**

NET Framework – C# access modifiers and other common types

- Other common types
  - **readonly vs const**
  - se
  - vi

readonly	const
Can be initialized: at declaration or in a constructor	Can be initialized **only at the declaration**
Can be used only by fields	

NET Framework – C# access modifiers and other common types

- Other common types
  - **readonly vs const**
  - se
  - vi

readonly	const
Can be initialized: at declaration or in a constructor	Can be initialized **only at the declaration**
Can be used only by fields	Can be used only by fields

NET Framework – C# access modifiers and other common types

- Other common types
  - **readonly vs const**
  - se
  - vi

readonly	const
Can be initialized: at declaration or in a constructor	Can be initialized **only at the declaration**
Can be used only by fields	Can be used only by fields
If is initialized via constructor can have different values	

NET Framework – C# access modifiers and other common types

- Other common types
  - **readonly vs const**
  - se
  - vi

readonly	const
Can be initialized: at declaration or in a constructor	Can be initialized **only at the declaration**
Can be used only by fields	Can be used only by fields
If is initialized via constructor can have different values	Is a compile-time constant

NET Framework – C# access modifiers and other common types

- Other common types
  - readonly v
  - se
  - vi

readon	
Can be in a co	
Can be	
If is ini have d	

```
class Age
{
    readonly int _year;
    Age(int year)
    {
        _year = year;
    }
    void ChangeYear()
    {
        //_year = 1967; // Compile error if uncommented.
    }
}
```

NET Framework – C# access modifiers and other common types

- Other common types
  - readonly vs const
  - **sealed**
  - virtual

NET Framework – C# access modifiers and other common types

- Other common types
  - readonly vs const
  - **sealed**
  - virtual

Can be used for classes and methods

NET Framework – C# access modifiers and other common types

- Other common types
  - readonly vs const
  - sealed
  - virtual

Classes => prevents other classes to inherit from it.

Methods => prevents from overriding virtual methods.

NET Framework – C# access modifiers and other common types

- Other co
- Clas
  - seale
  - virtu

Clas lasses

Metho erriding

```
class X
{
    protected virtual void F() { Console.WriteLine("X.F"); }
    protected virtual void F2() { Console.WriteLine("X.F2"); }
}
class Y : X
{
    sealed protected override void F() { Console.WriteLine("Y.F"); }
    protected override void F2() { Console.WriteLine("Y.F2"); }
}
class Z : Y
{
    // Attempting to override F causes compiler error CS0239.
    // protected override void F() { Console.WriteLine("C.F"); }

    // Overriding F2 is allowed.
    protected override void F2() { Console.WriteLine("Z.F2"); }
}
```

NET Framework – C# access modifiers and other common types

- Other common types
  - readonly vs const
  - sealed
  - **virtual**

NET Framework – C# access modifiers and other common types

- Other common types
  - readonly vs const
  - sealed
  - virtual

It is used for: methods, properties indexer and events.

NET Framework – C# access modifiers and other common types

- Other common types
  - readonly vs const
  - sealed
  - virtual

It is used for: methods, properties indexer and events.

This allows us to override the behavior in a derived class.

One more thing…

One more thing…

- “Talk is cheap. Show me the code.”

One more thing…

- “Talk is cheap. Show me the code.” — Linus Torvalds
- Write comments only when strictly necessary and keep them in sync with the code.

One more thing…

- “Talk is cheap. Show me the code.” — Linus Torvalds
- Write comments only when strictly necessary and keep them in sync with the code.
- Establish a set of shared coding standards

One more thing…

- “Talk is cheap. Show me the code.” — Linus Torvalds
- Write comments only when strictly necessary and keep them in sync with the code.
- Establish a set of shared coding standards.
- “Any fool can write code that a computer can understand. Good programmers write code that humans can understand.”

One more thing…

- “Talk is cheap. Show me the code.” — Linus Torvalds
- Write comments only when strictly necessary and keep them in sync with the code.
- Establish a set of shared coding standards.
- “Any fool can write code that a computer can understand. Good programmers write code that humans can understand.” – Martin Fowler

Questions

- Do you have any other questions?

Thanks!

See you next time! ☺