Einführung in C# 8 und.NET Core

Überblick und Administratives

Martin Kropp / Yves Senn
University of Applied Sciences Northwestern Switzerland

Fachhochschule Nordwestschweiz
Hochschule für Technik

Über die Vorlesung

- Einführung C# 8 und .NET Core 3

- Voraussetzung

  Sie sind *routiniert* im Umgang mit Java und OOP

- Fokus
  - Die Sprache C#
  - .NET Grundlagen / Blick hinter die Kulissen
  - ~~Programmiergrundlagen (OO, Patterns)~~
  - ~~Frameworks (XAML, ASP.NET, EF, …)~~

Unterrichtsgestaltung

- Vorlesung
  - mit Arbeitsblättern

- Fallstudie

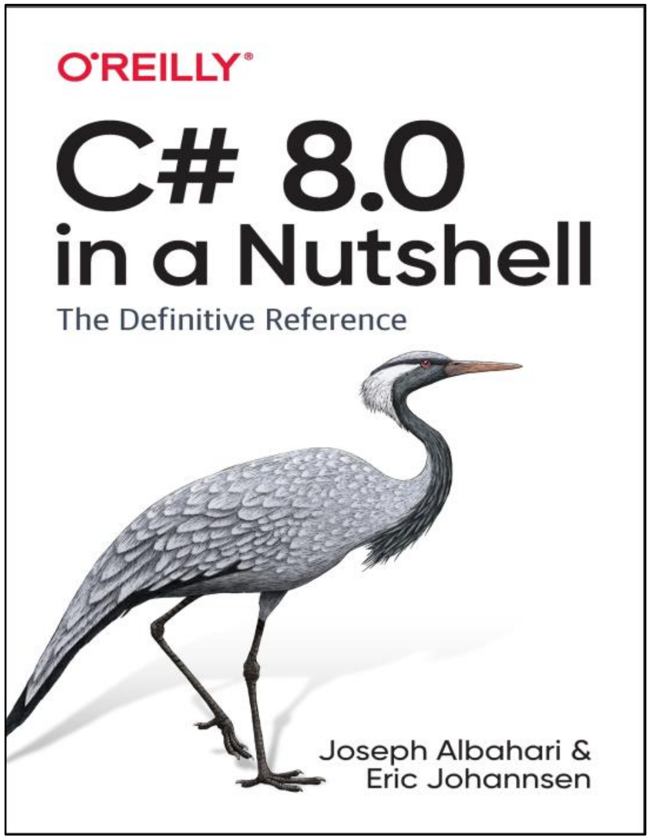
- Selbststudium

Support

1. Selbst versuchen und recherchieren
2. Mail schicken mit Screenshots und relevantem Commit oder Merge-Request

Fachhochschule Nordwestschweiz
Hochschule für Technik

Literatur



Skript zur Vorlesung

C# 8.0 in a Nutshell:

The Definitive Reference

Joseph Albahari, Eric Johannsen

O'Reilly Media *th Edition 2020*

ISBN: 978-1-492-05113-8

Für den Unterricht relevante Kapitel stehen auf dem AD zur Verfügung.

Kurs-Ressourcen

- Handouts
  - Auf AD: \\Fsemu18.edu.ds.fhnw.ch\e_18_data11$\E1811_Unterrichte_Bachelor\E1811_Unterrichte_I\3Ia\ecnf
  - Drehbuch
  - Auf Web Seite: http://web.fhnw.ch/plattformen/ecnf
- Ihr Code
  - In Ihrem Git Repo: (die Links erhalten Sie bald)

Leistungsbewertung

- Modulnote = Erfahrungsnote
  - 2 Prüfungen zu je 50%

Fallstudie – RoutePlanner

- Zweck
  - Üben, üben, üben…
  - Anwenden der erlernten C# Konzepte, Methoden und Tools
- Kontext
  - Entwicklung einer kompletten eigenen Applikation
  - Fallstudie – Routen Planung
- Aufgabe
  - Jede Gruppe entwickelt das System
  - Jede Gruppe meldet sich, sofern Sie ein GIT-Repository möchten
  - Sie bekommen dazu jede Woche neue Aufgaben gestellt

Fallstudie – RoutePlanner

- Bewertung
  - Die Fallstudie wird nicht bewertet; die Teilnahme ist nicht zwingend. Wir empfehlen es **jedoch dringend**. Sämtliche praktischen Übungen sind prüfungsrelevant. Für rechtzeitig abgegebene Lösungen erhalten Sie Feedback Ihres Dozierenden (Merge-Request)
- Abgabe
  - Solution ins Git Repository ihrer Gruppe.
  - *Merge-Request an den Dozenten bei jedem Meilenstein*
  - ***Hinweis**: Pushen Sie rechtzeitig und regelmässig; nicht erst zum Schluss. Merge-Request an den Dozenten bei jedem Meilenstein*
- Meilenstein MS00
  - Projektteams gebildet, jeder in Gruppe eingeschrieben
  - Einschreiben auf: https://docs.google.com/spreadsheets/d/1qadhY_riZjJQCrYTmsjffMsKWytwehug8GjbaCuiVkU/edit?usp=sharing

Fachhochschule Nordwestschweiz
Hochschule für Technik

OVERVIEW OF .NET

M. Kropp/Y. Senn
University of Applied Sciences Northwestern Switzerland

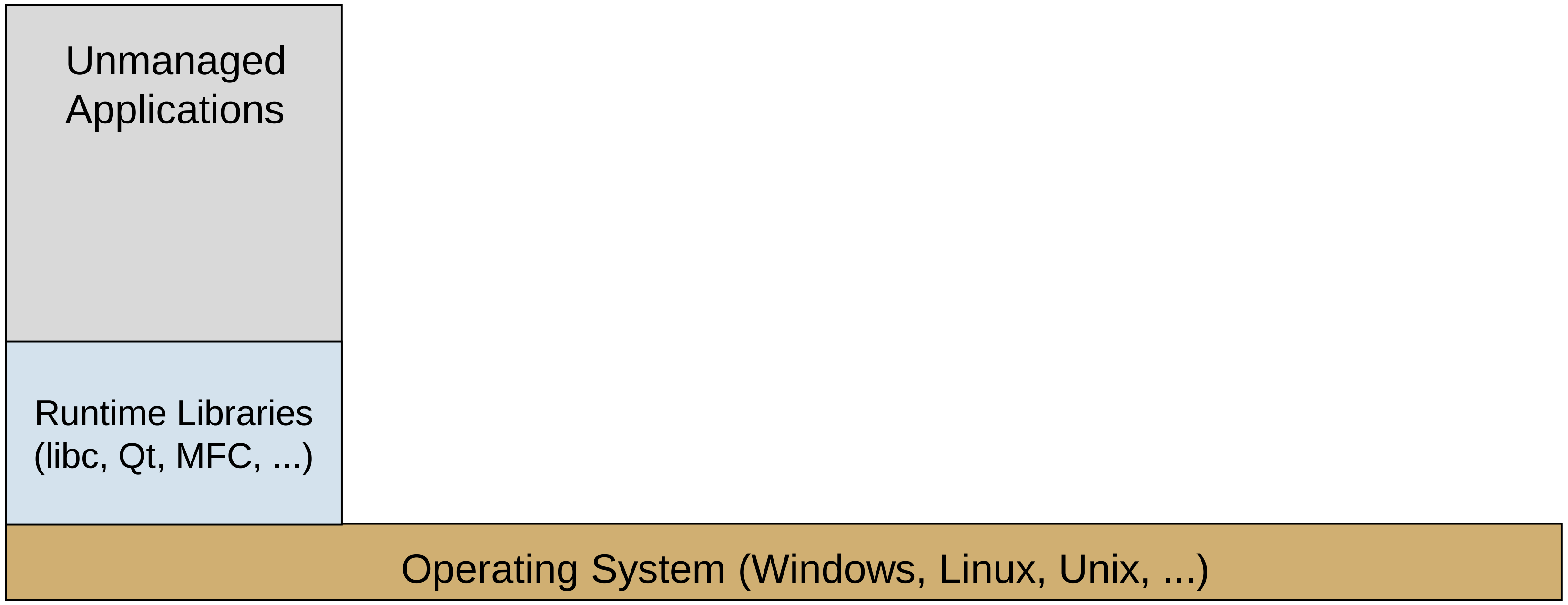
Learning Objectives

- You
  - can name the goals of the .NET Core architecture
  - can explain the terms “managed”, “unmanaged” code, assemblies and Intermediate Language.
  - know the basic building blocks of a .NET Core and C# program
  - can write your “first program” in C# and Visual Studio

Content

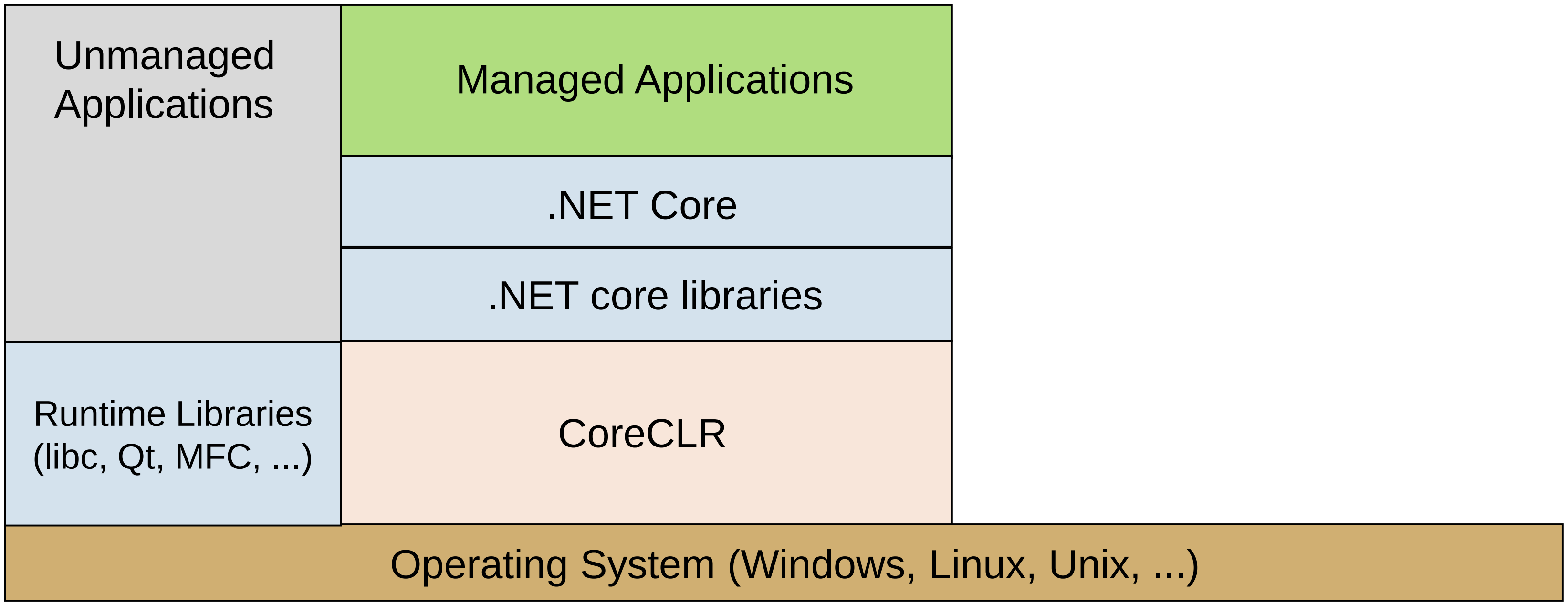
What is .NET?

- Before .NET


Unmanaged Applications
Runtime Libraries (libc, Qt, MFC, ...)
Operating System (Windows, Linux, Unix, ...)

What is .NET?

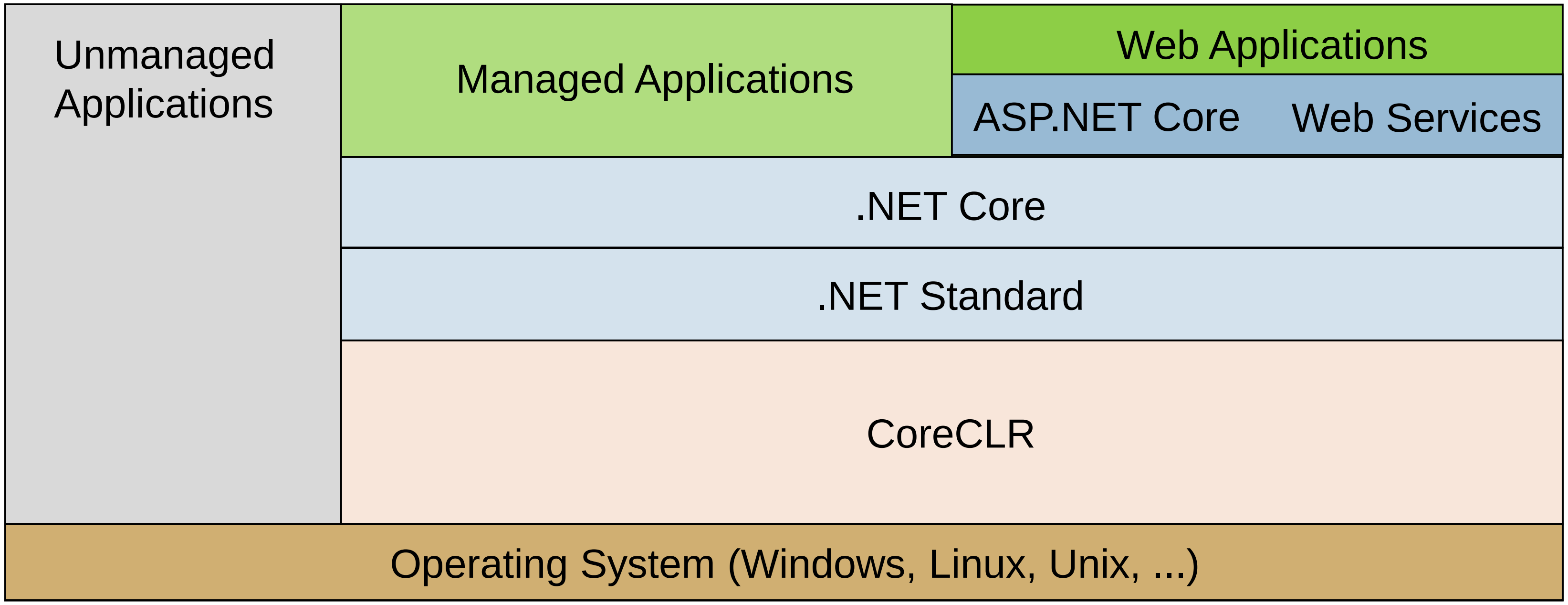
- A VM based development and runtime software platform


Unmanaged Applications
Managed Applications
.NET Core
.NET core libraries
Runtime Libraries (libc, Qt, MFC, ...)
CoreCLR
Operating System (Windows, Linux, Unix, ...)

CoreCLR (Common Language Runtime)	interoperability, security, garbage collection, versioning, ...
Base Class Library	GUI, collections, threads, networking, reflection, XML, ...

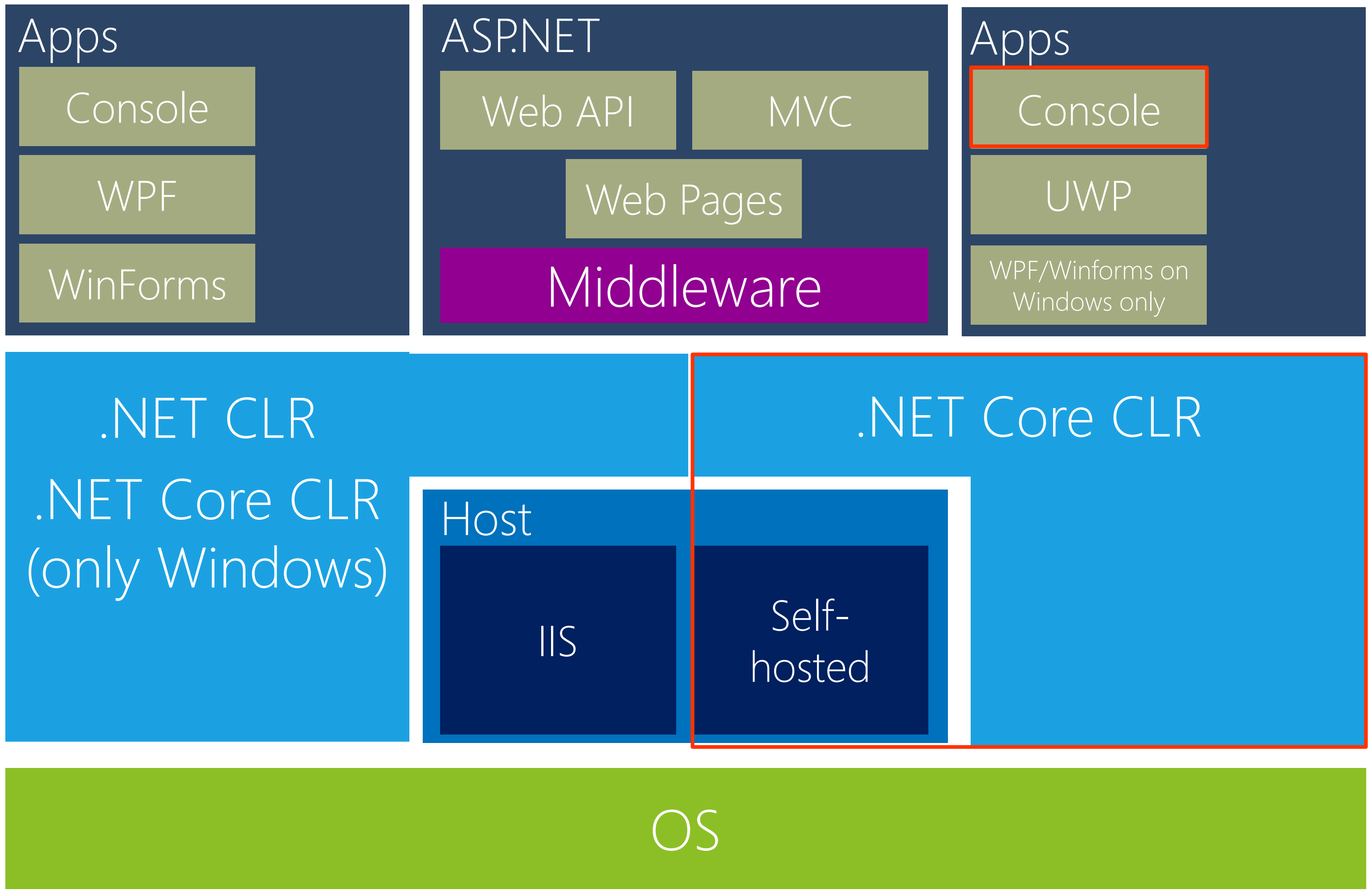
What is .NET?

- A uniform platform for the desktop, the Web, and more…



.NET Architecture (current)

Fachhochschule Nordwestschweiz
Hochschule für Technik



Our Focus

Goals of .NET

Uniform model for development

Before

iOS apps

Compiled (Objective C, Swift)

Extensive class libraries

Android apps

AOT compiled (Java)

Extensive class libraries

Data science

R/Python/IDL

Interpreted, slow

Games

Compiled (C++, assembly, ...)

Extensive class libraries

RAD/LOB desktop apps

Access/Delphi/VB6/…

Varying execution models

Web

ASP/PHP/JavaScript...

Interpreted, slow

Specialized libraries

Under .NET

All above scenarios

Object-oriented

JIT- or native compiled (C#, C++, VB.NET, ...)

Uniform class library (.NET Core)

Goals of .NET

Interoperability of programming languages

Before

- millions of lines of code in C++, Fortran, Visual Basic, ...
- very limited interoperability

Under .NET

- one **Common Intermediate Language** (CIL)
- binary compatibility between more than 20 languages (C#, C++, VB.NET, Java, Eiffel, Fortran, Cobol, ML, Haskell, Pascal, Oberon, Perl, Python, ...)

class in VB.NET

```
Public Class A
  Public x As Integer
  Public Sub Foo() ...
End Class
```

subclass in C#

```
class B : A
{
  public string s;
  public void Bar() {...}
}
```

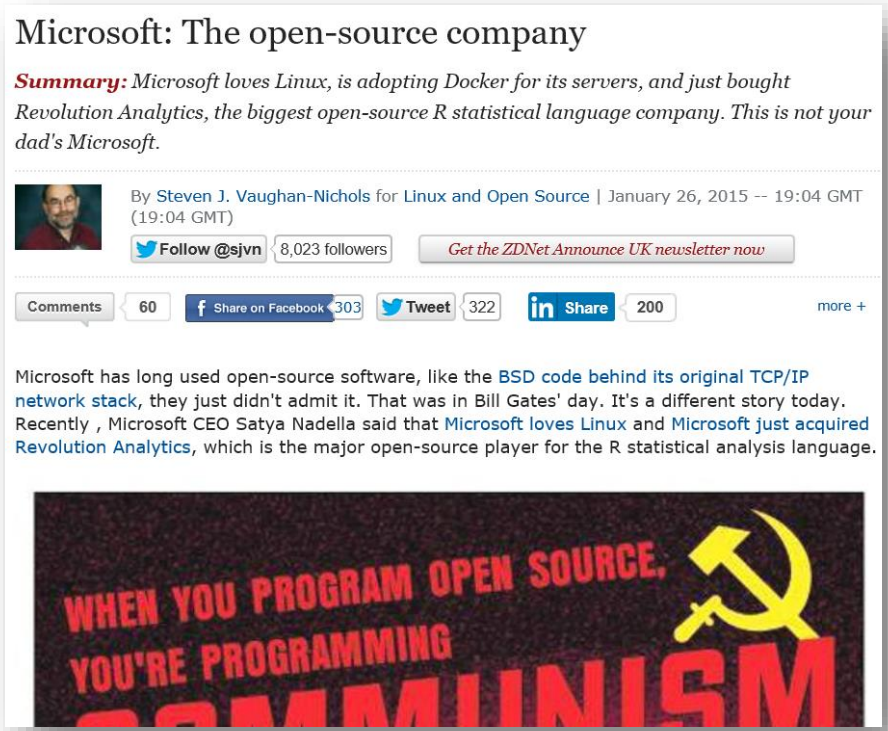
used in Eiffel

```
class Client feature
  obj: B;
  ...
  create obj;
  obj.Bar;
  ...
end
```

.NET is Open Source

https://github.com/dotnet

http://sourceof.net


Microsoft: The open-source company

Summary: *Microsoft loves Linux, is adopting Docker for its servers, and just bought Revolution Analytics, the biggest open-source R statistical language company. This is not your dad's Microsoft.*

By Steven J. Vaughan-Nichols for Linux and Open Source | January 26, 2015 -- 19:04 GMT (19:04 GMT)

Microsoft has long used open-source software, like the BSD code behind its original TCP/IP network stack, they just didn't admit it. That was in Bill Gates' day. It's a different story today. Recently , Microsoft CEO Satya Nadella said that Microsoft loves Linux and Microsoft just acquired Revolution Analytics, which is the major open-source player for the R statistical analysis language.

- Compilers (Roslyn)
- Runtime (JIT & GC)
- Foundational libraries
- ASP.NET, .NET Core
- Package mgmt.

http://www.dotnetfoundation.org/

State in 2019

- Free IDEs
  - Visual Studio IDE Community Edition
  - Visual Studio Code
  - Visual Studio for Mac
  - MonoDevelop

- Development of new features as Open Source
  - Language design (C# & VB.NET)
  - API reviews
  - Libraries (ASP.NET, Entity Framework, …)

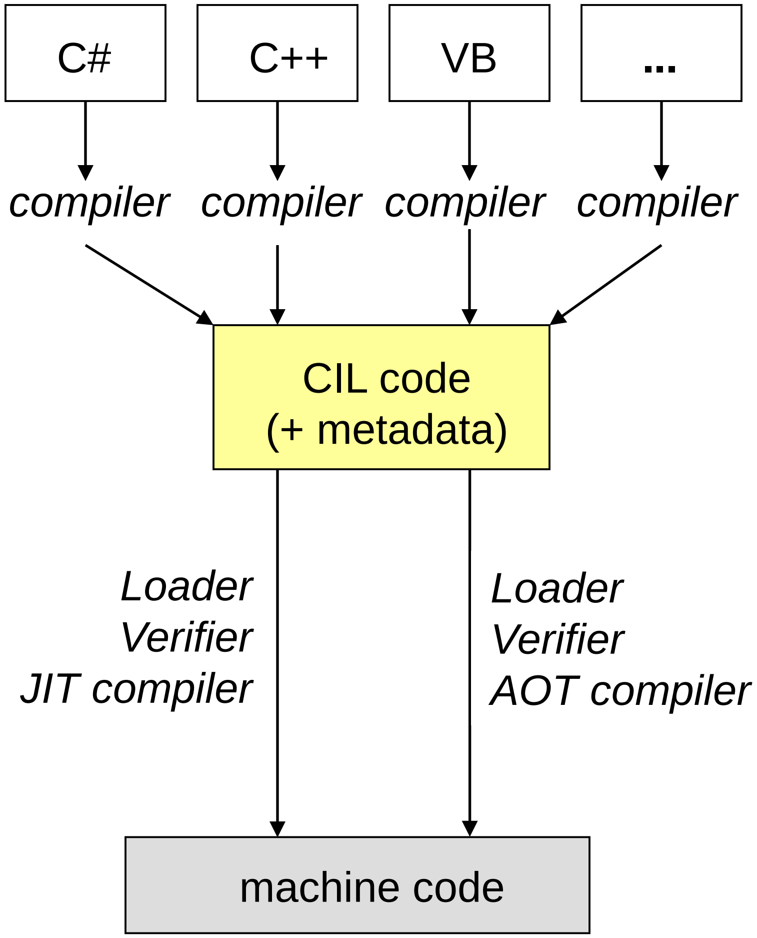
State in 2019

- .NET is cross platform
  - .NET Core runs on Windows, OS X, Linux
  - Mono/Xamarin runs on Windows, OS X, Linux, iOS, Android
- Embeddable
  - Unity
  - Arduino.NET
  - TouchDevelop
  - ...

Open standard

- Common Language Infrastructure (CLI)
  - ECMA-335 / ISO/IEC 23271
- C#
  - ECMA-334 / ISO/IEC 23270
- CLI implementations
  - .NET is Microsoft's implementation
  - Mono is Novell's/Ximian's implementation
  - ...

Intermediate Language



C#

```
if (a > b) max = a; else max = b;
```

CIL

```
IL_0004:  ldloc.0
IL_0005:  ldloc.1
IL_0006:  ble.s       IL_000c
IL_0008:  ldloc.0
IL_0009:  stloc.2
IL_000a:  br.s        IL_000e
IL_000c:  ldloc.1
IL_000d:  stloc.2
```

x86 code

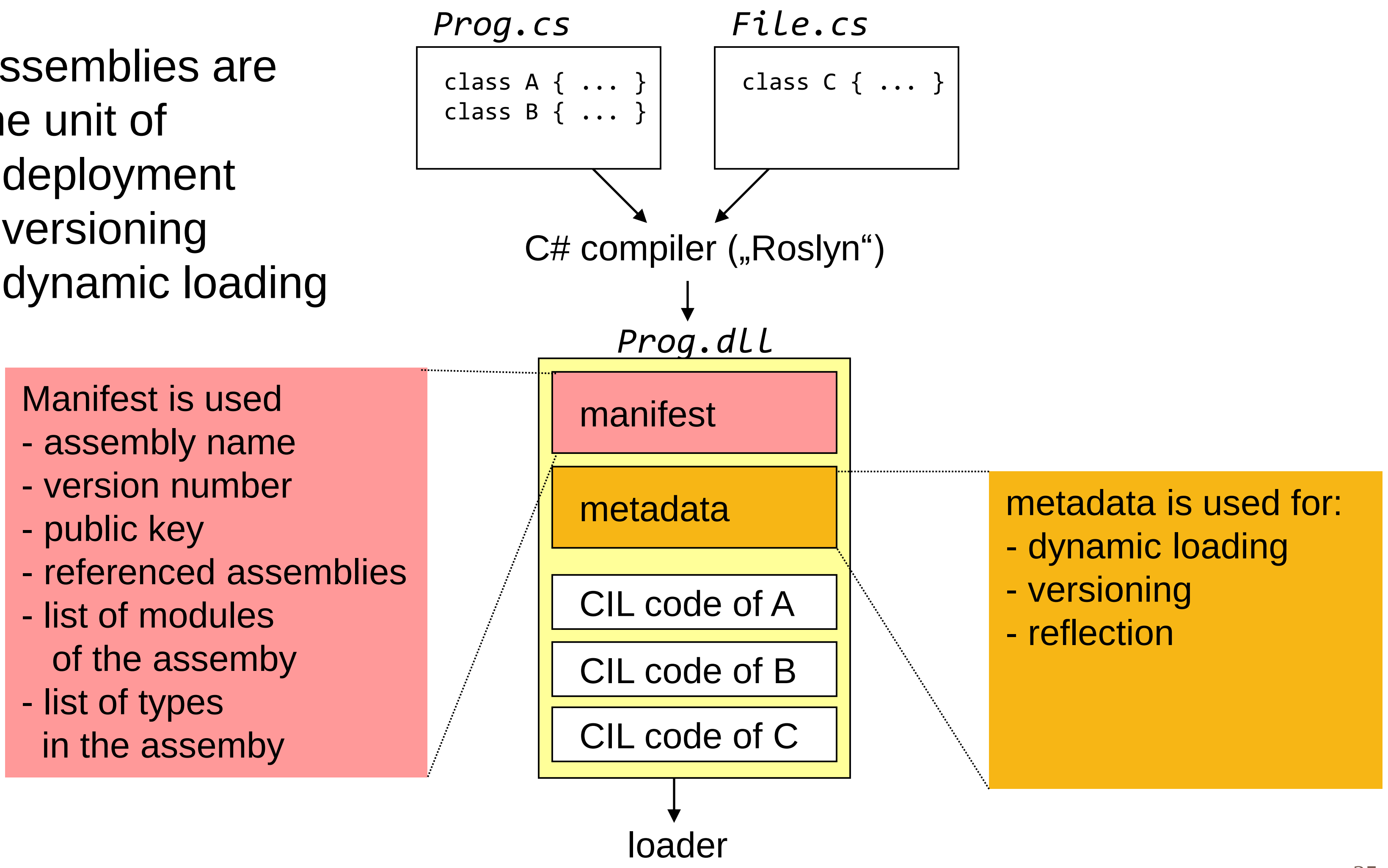
```
mov  ebx,[-4]
mov  edx,[-8]
cmp  ebx,edx
jle  17
mov  ebx,[-4]
mov  [-12],ebx
...
```

Fachhochschule Nordwestschweiz
Hochschule für Technik

Assemblies

Assemblies are the unit of

- deployment
- versioning
- dynamic loading



Manifest is used
- assembly name
- version number
- public key
- referenced assemblies
- list of modules of the assemby
- list of types in the assemby

metadata is used for:
- dynamic loading
- versioning
- reflection

An example

Namespaces

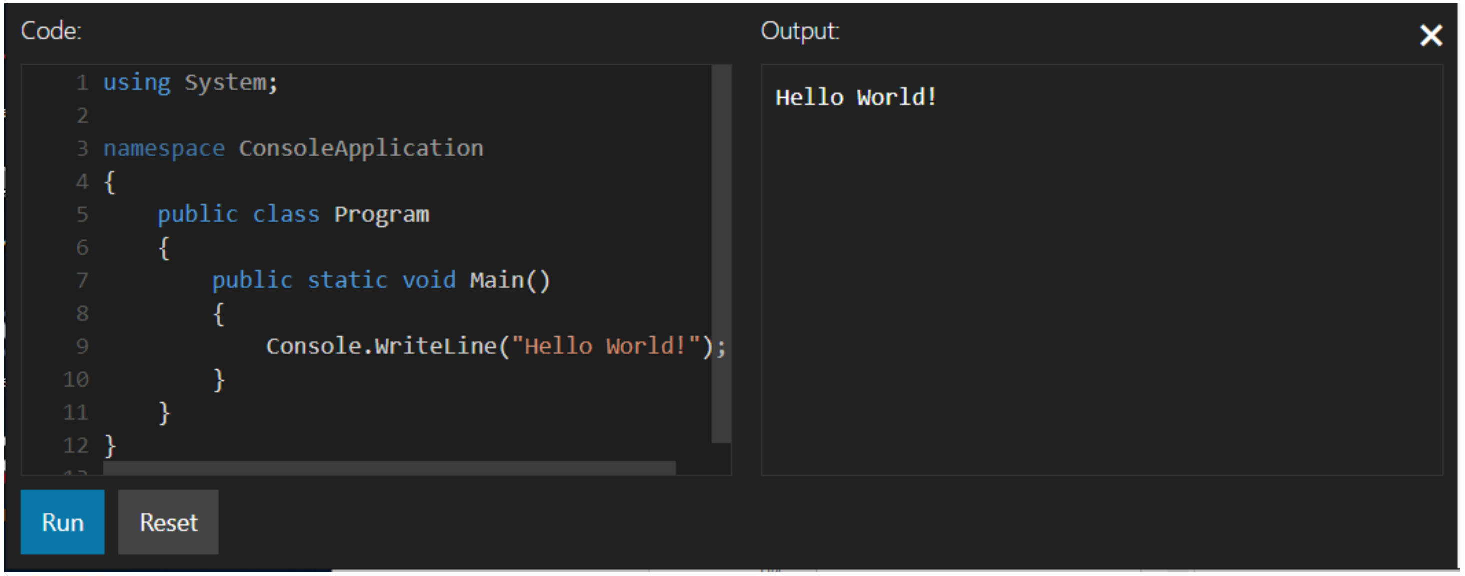
Methods with capital letters

String interpolation

```
using System;

namespace HelloTest
{
    class Hello
    {
        private string nam

        private void Greet()
        {
            Console.WriteLine($"Hello {name}");
        }

        public static void Main(string[] args)
        {
            var me = new Hello();
            me.name = args[0];
            me.Greet();
        }
    }
}
```

Try it yourself

- Web-Code Editor and Tutorial
- https://www.microsoft.com/net/tutorials/csharp/getting-started



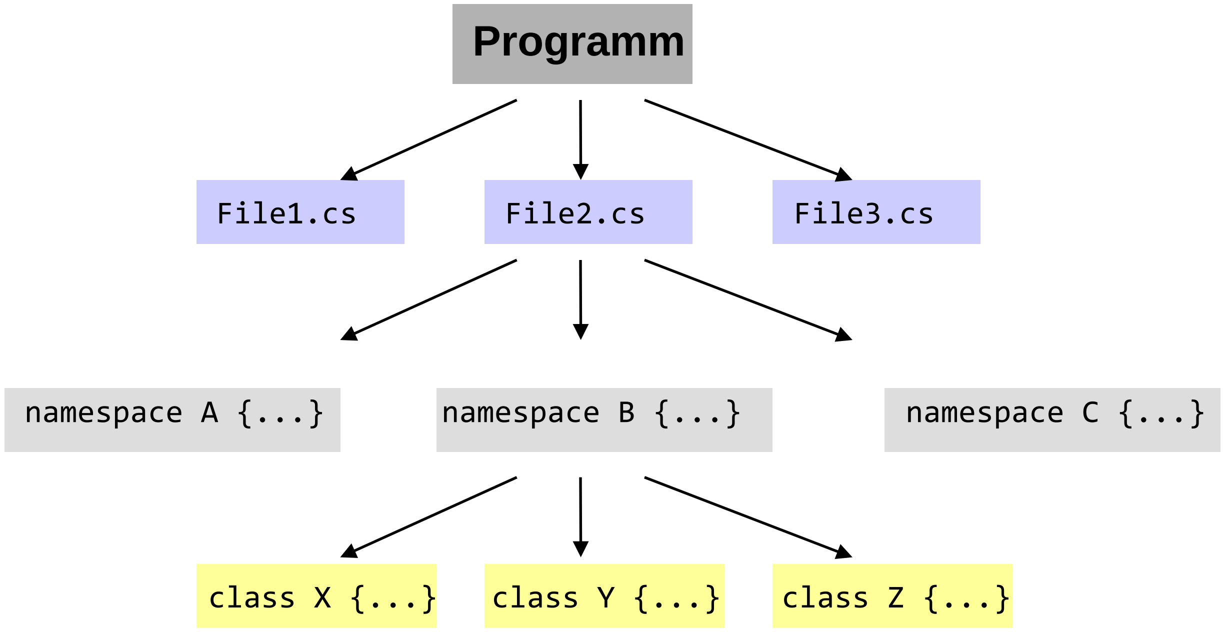
```
using System;

namespace ConsoleApplication
{
    public class Program
    {
        public static void Main()
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

```
Hello World!
```

Visual Studio Intro

Fachhochschule Nordwestschweiz
Hochschule für Technik

Structure of C# programs



- If no namespace is specified → anonymous default namespace
- Namespaces may contain classes, structs, interfaces, delegates and enums
- Namespace can be used by multiple files

Namespaces

Allows hierarchical organization of classes:

Declaration

```
namespace A
{
        namespace B      //name A.B
        {
                ...
        }
}
```

Usage

```
using A;
    class C
    {
        ...
    }
```

- A file can declare multiple namespaces
- Namespaces and classes are not mapped to directories and files (**but recommended**)
- Similar to Java packages (not identical)

More on namespaces

- Scope
  - Outer namespace names are imported into inner
  - Inner namespace names are hidden in outer

```
namespace A
{
    class ClassA {}
}
```

```
namespace B
{
  using A;
  class ClassB : ClassA {}
}
class ClassC : ~~class ClassA~~
```

Not known outside namespace B

- Aliasing

```
using ClassTypeAlias = Deep.Neested.Namespace.ActualType;
```

About symbols and naming

- Identifiers in Unicode and case-sensitive
- The @ string literal prefix
  - Backslashes won't be interpreted as escape characters e.g. `@"d:\temp"`, instead of `"d:\\temp“`
  - Use of new line is allowed after @ prefix
- Comments
  - As in Java
  - `///` is a documentation comment
- Naming Conventions
  - CamelCasing (e.g. ShowDialog)
  - First letter in upper case, **except for private or locals** (variables and fields)
  - Details: http://msdn.microsoft.com/en-us/library/ms229002(v=vs.110).aspx

```
string x =
@"Mehrzeiliger
Text";
```

Summary

- .NET is a virtual machine based system. It’s VM is called *CoreCLR (Common Language Runtime)*

- .NET emphasizes language interoperability and platform independence

- Managed programs are compiled into the Common Intermediate Language (CIL)

- CIL-code has to be (JIT) compiled and executed as machine code