VSR://EDU/SSE

Software Service Engineering

WS 2025/2026 – 0. Tutorial

Valentin Siegert M.Sc.

Maheshika Walpola M.Sc.

*VSR.Informatik.*TU-Chemnitz.*de*

Form:

- Detailing course knowledge
- Discussion
- Task Solving (by students, NOT by the tutor)
- Homework (voluntary)

Computer access is a must-have for this course!

News, Materials:

- https://vsr.informatik.tu-chemnitz.de/news/edu/
- https://vsr.informatik.tu-chemnitz.de/edu/current/sse/
- OPAL: Software Service Engineering WS25/26
  https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/51007553539

Contact:

- vsr-sse@informatik.tu-chemnitz.de
- A11.204 (find us: https://mytuc.org/r/A11.204)

Task 1

The programming language of the tutorial is C#.

Read any C#-Tutorial of your choice and create a simple Hello-World console application using Visual Studio.

Visual Studio Community can be downloaded from Microsoft.

https://visualstudio.microsoft.com/

- C# Characteristics
  - Very similar to Java
  - Strict type system
  - Object-oriented
  - Automatic Garbage Collection

➔Focus on robustness, durability, productivity

- Data types
  - Value types: int, double, bool, …
  - Reference types: string, object, Exception,…
  - Generic types: List<string>, Stack<int>
  - Boxing and Unboxing

```
// Value types
int i = 1;
double d = 0.25;
bool b = true;

// Reference types
string s = @"This is escaped \string";
object obj = new StringBuilder();
```

```
// Boxing and unboxing
Int32 boxedInt = i;
object boxedInt2 = i;
int unboxedInt = (int) boxedInt;
int unboxedInt2 = (int) boxedInt2;
```

Control structures

```
if(a == b)
{
    foreach (var item in items)
    {
        switch (item)
        {
            case "a":
              a++;
              break;
            default:
              b++;
              break;
        }
    }
}
```

```
for(int i=0; i<5; i++)
{
    while(i > 2)
    {
        do
        {
          Console.WriteLine("hello");
        } while (i < 3);
    }
}
```

Inheritance and interfaces

```
public interface INetwork
{
    NetworkAddress ResolveHostName(string serverName);
}

public abstract class AbstractNetwork : INetwork
{…}

public class EthernetNetwork : AbstractNetwork
{…}

public class WirelessNetwork : AbstractNetwork
{…}
```

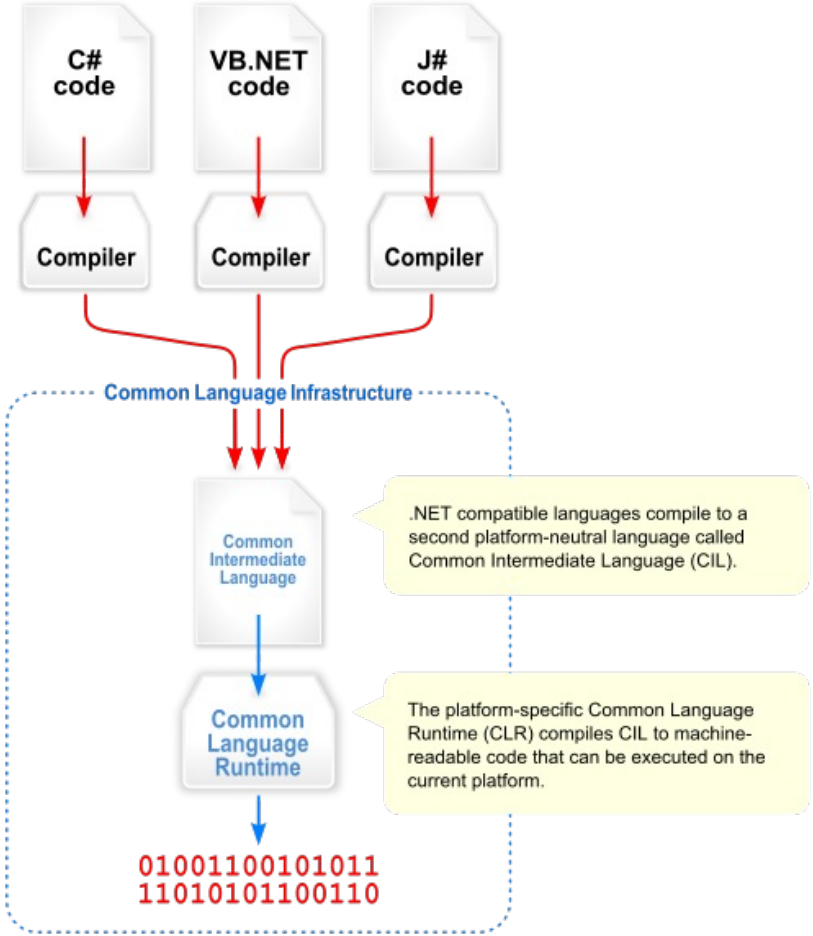
- Class library
  - IEnumerable
  - object[]
  - List<…>
  - Dictionary<… , …>
  - Exception
  - Regex
  - Random
  - Console

Task 2

Answer the following questions:

- What are the responsibilities of the .NET Framework?
- What is common to C#, VB.NET, F#, J#?
- What are the Lambda-Expressions and LINQ?

- Development using .NET-Framework
  - Runtime Environment
    - Memory and resource management
  - Class Library
    - More than 12000 classes and datatypes
    - Grouping into namespaces
  - Tools and services
- Outcomes:
  - Console, Desktop, Web Applications
  - (Web)Services
  - Class Libraries
  - Components


C# code
VB.NET code
J# code
Compiler
Compiler
Compiler
Common Language Infrastructure
Common Intermediate Language
.NET compatible languages compile to a second platform-neutral language called Common Intermediate Language (CIL).
Common Language Runtime
The platform-specific Common Language Runtime (CLR) compiles CIL to machine-readable code that can be executed on the current platform.
01001100101011
11010101100110

Lambda-Expressions and LINQ

```
Func<int, int> transformer = x => x*x;
Func<int, int, int> transformer2 = (x,y) => x + y;
Console.WriteLine(transformer(3));
Console.WriteLine(transformer2(4,5));

IEnumerable<Point> points = GetAllPoints()
IEnumerable<string> labels = points.Where(point => point.X > 10).
                             Select(point => point.Label);


labels = from point in points
         where point.X > 10
         select point.Label;
```

VSR

Your feedback on today's session:

mytuc.org/ttbw

Questions?

vsr-sse@informatik.tu-chemnitz.de

VSR.*Informatik*.TU-Chemnitz.*de*