

.NET og C#

Session 1, Intro



- Intro
 - Velkomst
 - Indhold
 - C# og .NET: Overblik
- Hello World
 - Visual Studio
 - Projects, solutions, languages
 - Console app
 - Kør console app
- Lidt mere på
 - Variabler
 - Conditionals og loops



Hvem er jeg

- Klaus Bøgestrand
- Cand.Scient datalogi/matematik 1987 (forrige århundrede)
- 2½ år på VIA, undervisning i Java
- 13 år hos IBM: Java, .NET mm.
- 4 år på teknologisk institut, bl.a. kurser i .NET
- Mere undervisning, for længe siden



Kursusplan overordnet

- Tre dele
 - Del 0: grundlæggende
 - Del 1: C# og Desktop Applikationer.
 - Del 2: C# og web applikationer

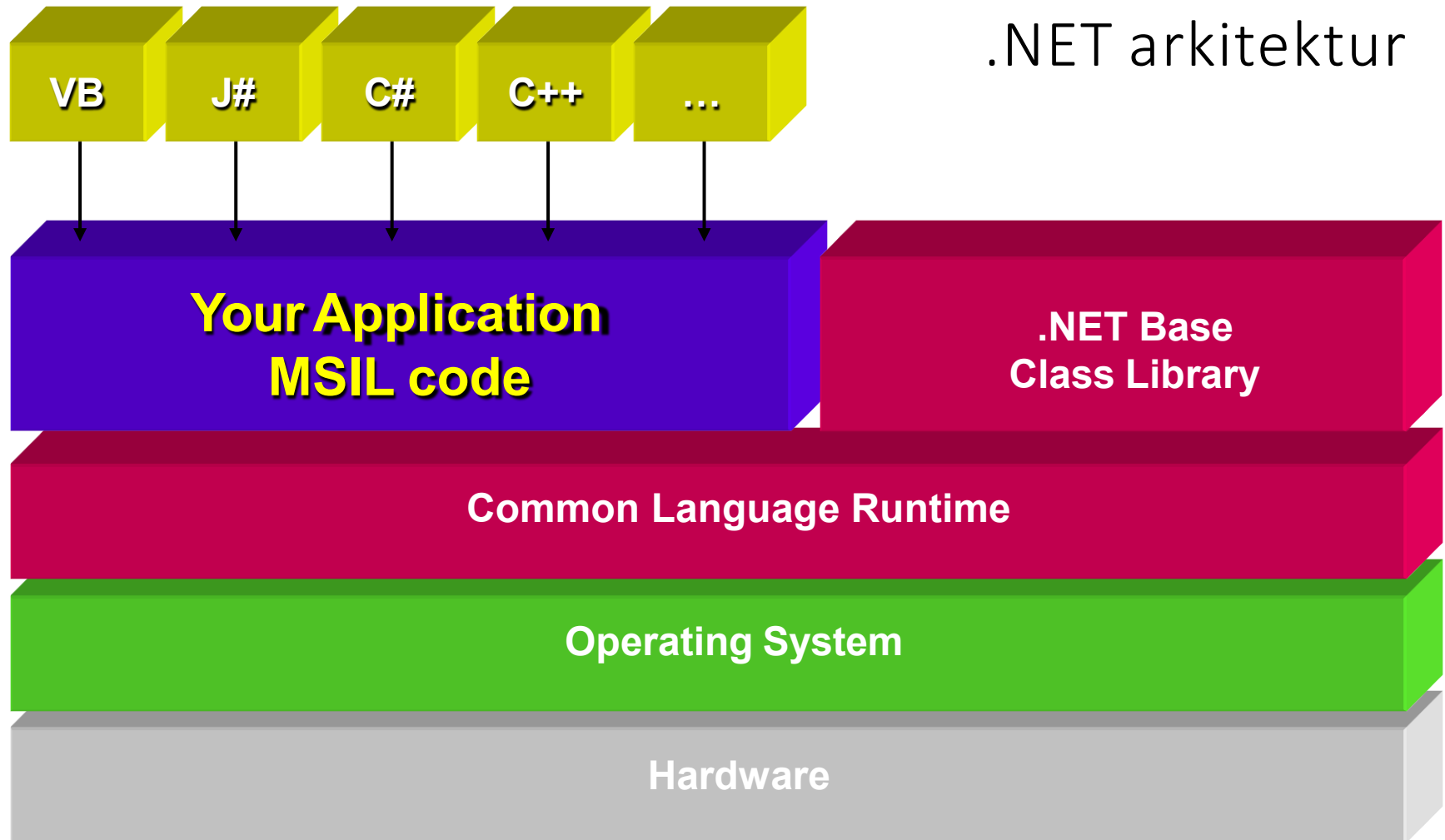


.NET og C# historisk

- .NET: Forsøg på flere ting:
 - Konkurrent til Java
 - Overbygning på/udskiftning af gamle teknologier som COM(+), Windows API, MFC. Lettere og mere objektorienteret måde at lave Windows-programmer på.
 - Køre på flere platforme
- C#: Sprog til .NET, inspireret fra Java, C++, C
- Visual Basic.net: inspireret fra Visual Basic (men meget anderledes), blot et andet sprog til .NET
- Andre sprog.



.NET arkitektur



.NET forkortelser

- CLR: Common Language Runtime
 - CLS: Common Language Specification
 - CTS: Common Type System
 - IL: Intermediate Language
 - CIL: Common Intermediate Language
 - MSIL: Microsoft Intermediate Language
 - JIT: Just In Time
 - FCL: Framework Class Library
 - COM: Component Object Model
-
- Det er ikke vigtigt at huske - blot for reference til de senere slides



Common Language Runtime (CLR) / Intermediate Language (IL)

- Source-code oversættes til IL-code af C#-compiler (eller VB, C++ compilers) **IL-code svarer til Java's Byte-code**
- IL-code bliver udført på en "controlled way" af CLR ("managed code").
- CLR svarer til JVM i Java (Java Virtual Machine)
- IL-code er JIT-compiled til native code
- IL er platform independent



Assemblies

- C# (og de andre sprog) oversættes til IL og pakkes i assemblies(dll's eller exe's)
- Så en applikation kan bestå af et assembly, der anvender andre assemblies
- Disse assemblies kan være lavet i forskellige sprog. Der ligger ikke noget om sproget i assemblies/IL
- Man kan f.eks. subclasse en klasse skrevet i et andet sprog
- Det "skal" dog være CLS compliant
- Assemblies indeholder metadata, f.eks. version



Common Type System (CTS)

- CTS definerer data typerne i Intermediate Language - f.eks. Klasser, interfaces etc.
- CTS: garanterer at en sub-klasse kan programmeres i et andet sprog end den oprindelige superklasse.
- CTS: exceptions kastet i et sprog kan fanges i et andet sprog



Common Language Specification (CLS)

- CLS er et delmængde af CTS (Common Type System) og nogle language features et.NET-language skal implementere for at sprogene kan bruges sammen.
- Det er f.eks., at (public) identifiers ikke må afvige kun i case
- Et assembly kan markeres som CLS-compliant. Så checker compileren.
- CTS og CLS er primært af interesse, hvis man skal skrive en compiler. Så vi vil ikke gå i flere detaljer med det!

.NET Framework Class Library

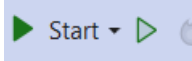
- En stor samling af klasser til Windows GUI, Web Forms, Data Access, Network Communication, XML-handling og meget mere
- Svarer til Java Foundation Classes
- System.Data
- System.Deployment
- System.Runtime
- System.Globalization
- System.IO
- System.Net
- System.Security
- System.Web
- System.Threading
- System.Xml
- Og mange flere

Application Domains – hvordan apps køres.

- En application kører i sit eget "application domain", en slags "sandbox"
- CLR tjekker at en application ikke kan referere til hukommelse i andres application domæner.
- Svarer en del til en proces i Windows



Visual Studio

- Forskellige project typer: Console app, Class Library, WPF app
- Kør applikation ved Start (F5) med debug eller 'Start without Debugging' (Ctrl+F5) 
- Indsæt breakpoints med F9
- Det tager tid at lære et nyt IDE. I starten vil I hade det, når I er vant til Eclipse

