Programming .NET and C#

Paolo Burgio
paolo.burgio@unimore.it

Before .NET

- ✓ Windows GUI development
  - Win32 API, MFC, Visual Basic, COM
- ✓ Web
  - ASP
- ✓ Java
  - ….

C Win32 API

"Traditional SW development", with all its limitations we know

- ✓ Explicit memory management, pointers, malloc/free
- ✓ Functional/procedural approach, no object-oriented
- ✓ Win32 API has 1000s of global functions and datatypes

C++/MFC

C++ is an object-oriented layer built on top of C

- ✓ Can mix the two!!

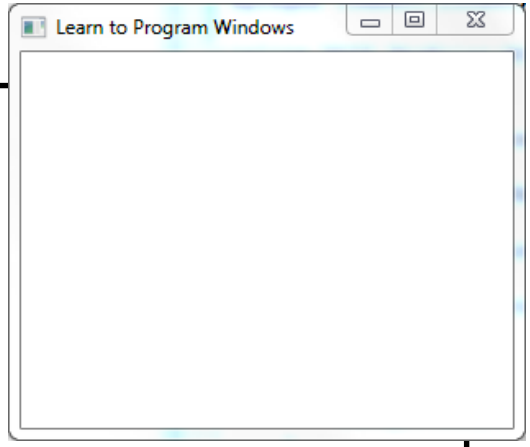
Microsoft Foundation Classes (MFC) is a set of C++ classes to build Win32 applications

- ✓ With a lot of tools, e.g., for code-generation
- ✓ GUI?

Greatly helps, but…

- ✓ Still, enormous
- ✓ Error-prone

Win32 API example




```
#include <windows.h>

int WINAPI wWinMain(HINSTANCE hInstance, HINSTANCE, PWSTR pCmdLine,
                    int nCmdShow) {
  // Register the window class.
  const wchar_t CLASS_NAME[] = L"Sample Window Class";
  WNDCLASS wc = { };
  wc.lpfnWndProc = WindowProc;
  wc.hInstance = hInstance;
  wc.lpszClassName = CLASS_NAME;
  RegisterClass(&wc);
  // Create the window.
  HWND hwnd = CreateWindowEx( 0, // Optional window styles.
                              CLASS_NAME, // Window class
                              L"Learn to Program Windows", // Window text
                              WS_OVERLAPPEDWINDOW, // Window style
                              // Size and position
                          CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,
                              NULL, // Parent window
                              NULL, // Menu
                              hInstance, // Instance handle
                              NULL // Additional application data );
  if (hwnd == NULL) { return 0; }
  ShowWindow(hwnd, nCmdShow);
  // Run the message loop.
  MSG msg = { };
  while (GetMessage(&msg, NULL, 0, 0))
   { TranslateMessage(&msg); DispatchMessage(&msg); }

  return 0;
}
```

Java

Pros

- ✓ Solves many issues of C++
- ✓ Provides "packages" with helpful classes/functionalities
- ✓ Can embed hightly-optimized C code with JNI

Cons

- ✓ Limited capability of accessing non-Java APIs
- ✓ Little support for real cross-language

Microsoft COM

Application development framework

- ✓ Reusable binary code, e.g., between different languages
- ✓ C++ can be used in VB6
- ✓ C can be used in Delphi

Cons

- ✓ Limited language independence
- ✓ High degree of complexity
- ✓ Active Template Library (ATL) to provide C++ support classes, templates, macros..

.NET, finally

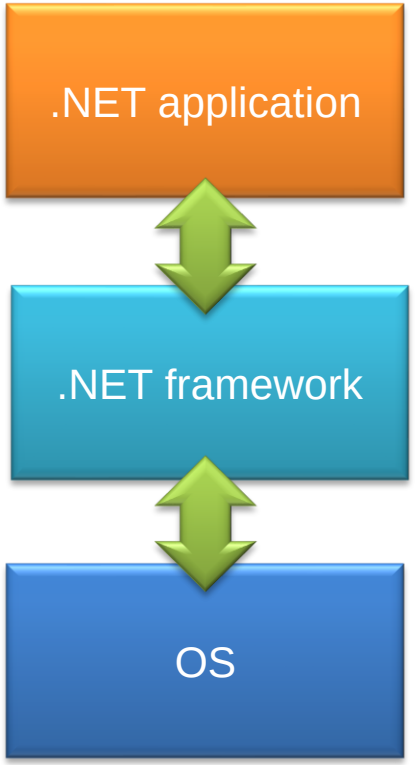
Built "after ~~failures~~ experience"

Key aspects

- ✓ OOP
- ✓ Runtime environment (MS "JVM")
- ✓ GUI
- ✓ Web
- ✓ Component-based design
- ✓ Multi-tier design
- ✓ Cross-language interoperability

What is .NET

- ✓ A framework
- ✓ A programming methodology
- ✓ Platform independent
- ✓ Language-insensitive



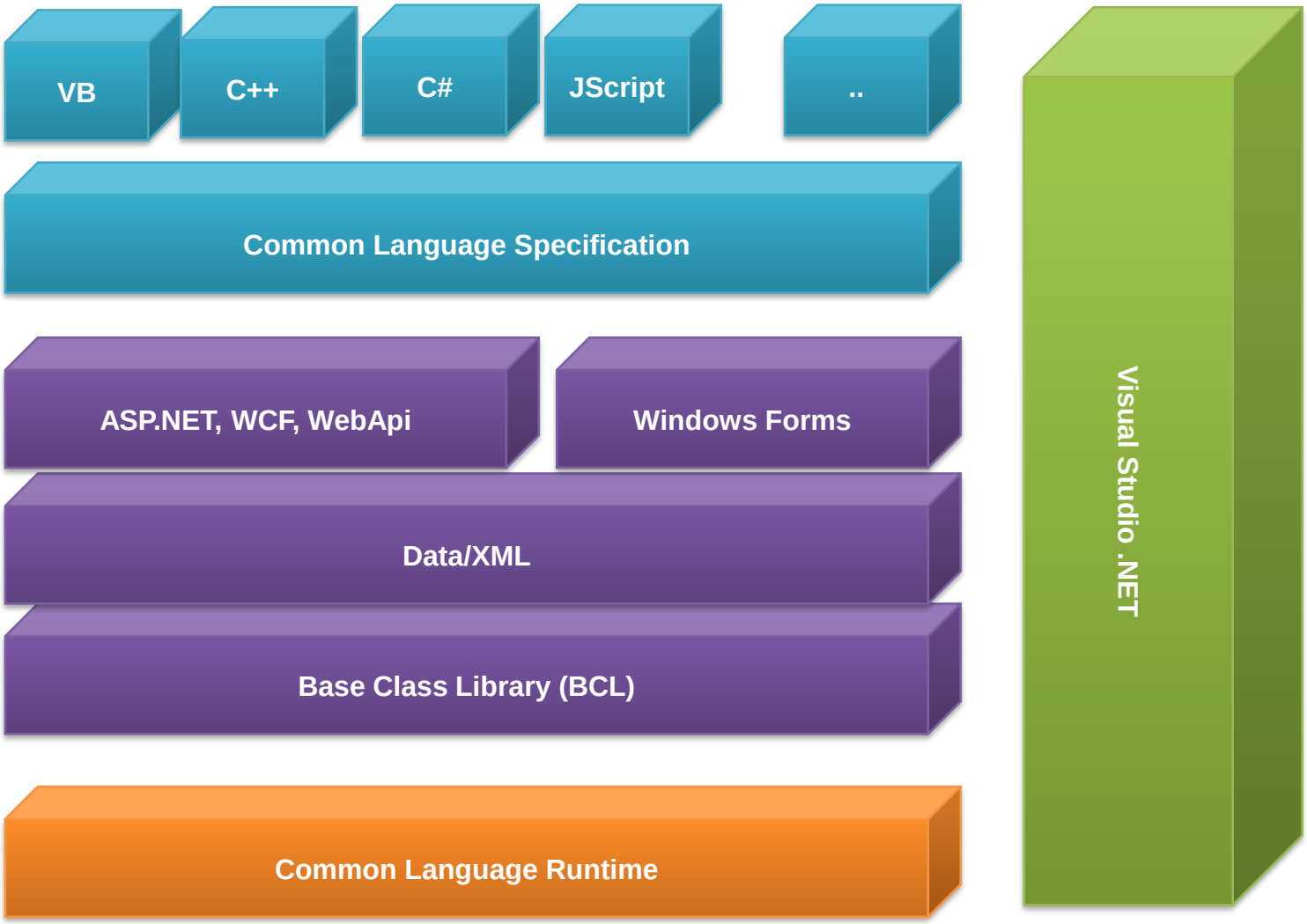
What .NET provides

- ✓ An integrated environment
- ✓ For internet | desktop | mobile app development
- ✓ Object-oriented
- ✓ Portable
- ✓ Managed (???)

.NET architecture

- ✓ Multi-language, cross-platform
- ✓ "JVM-like" environment called Common Language Runtime (CLR)
- ✓ Framework Class Library (FCL) providing a rich set of classes, templates…
- ✓ Just-in-Time conversion between CLR and "native" binary
- ✓ .NET components are packaged in assemblies

.NET technical architecture



.NET ecosystem

- ✓ Language interoperability
  - Common Language Specification (CLS)
  - Common Language Runtime (CLR)
  - Specific language compilers (C#, VB, Managed C++…)
- ✓ Frameworks & libraries
  - WebAPI, WCF, ASP.NET, Windows Forms, Xamarin, Framework Class Libs (FCL)…
- ✓ Design patterns
  - GUI, MV-VM, XAML

IDE

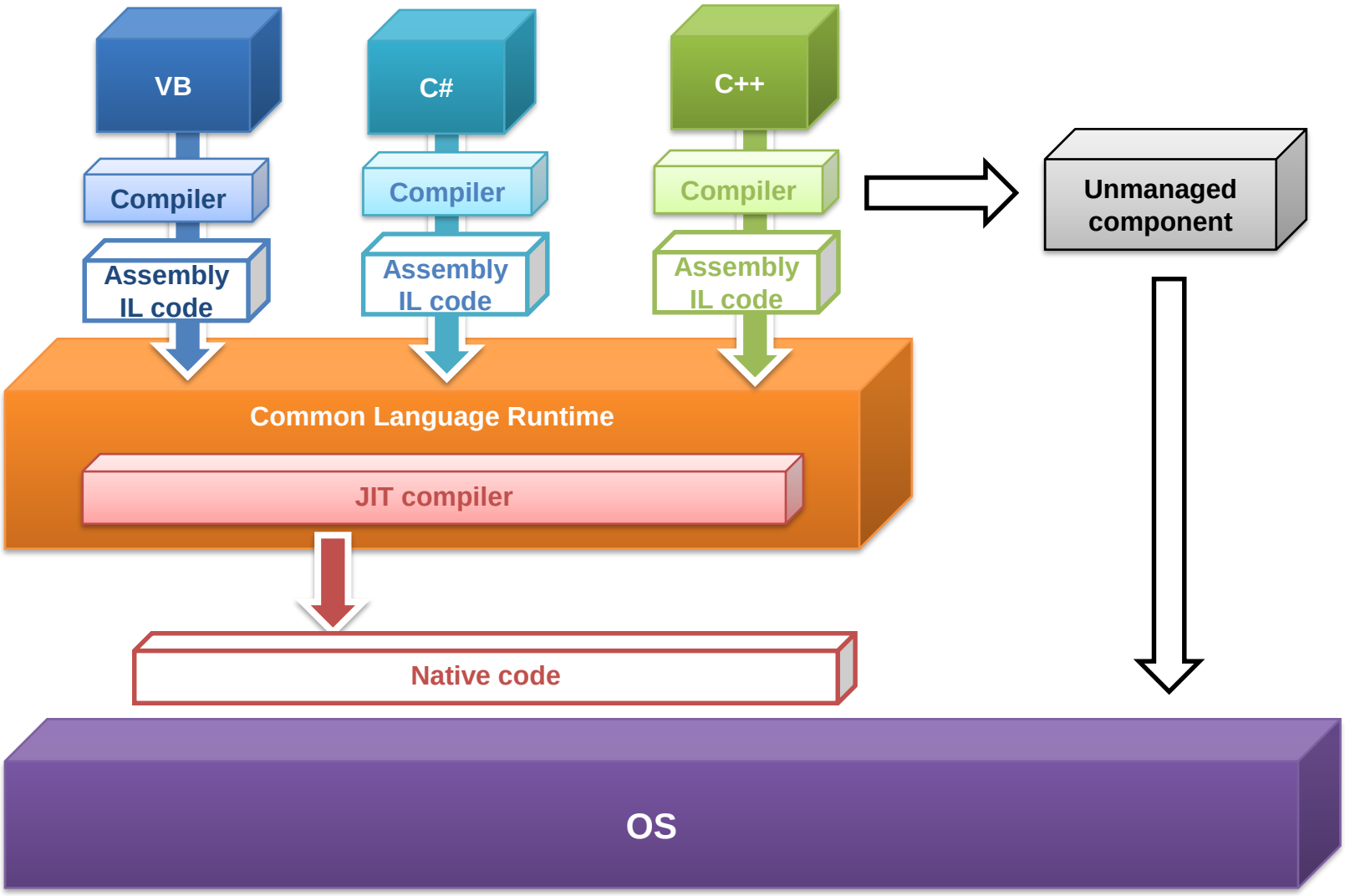
- ✓ Visual Studio

Common Language Runtime

- ✓ A common runtime for all .NET languages
  - Common type system
  - Common metadata
  - Intermediate Language (IL) to native code compilers
  - Memory allocation and garbage collection

- ✓ "Managed C++" vs "Unmanaged C++"

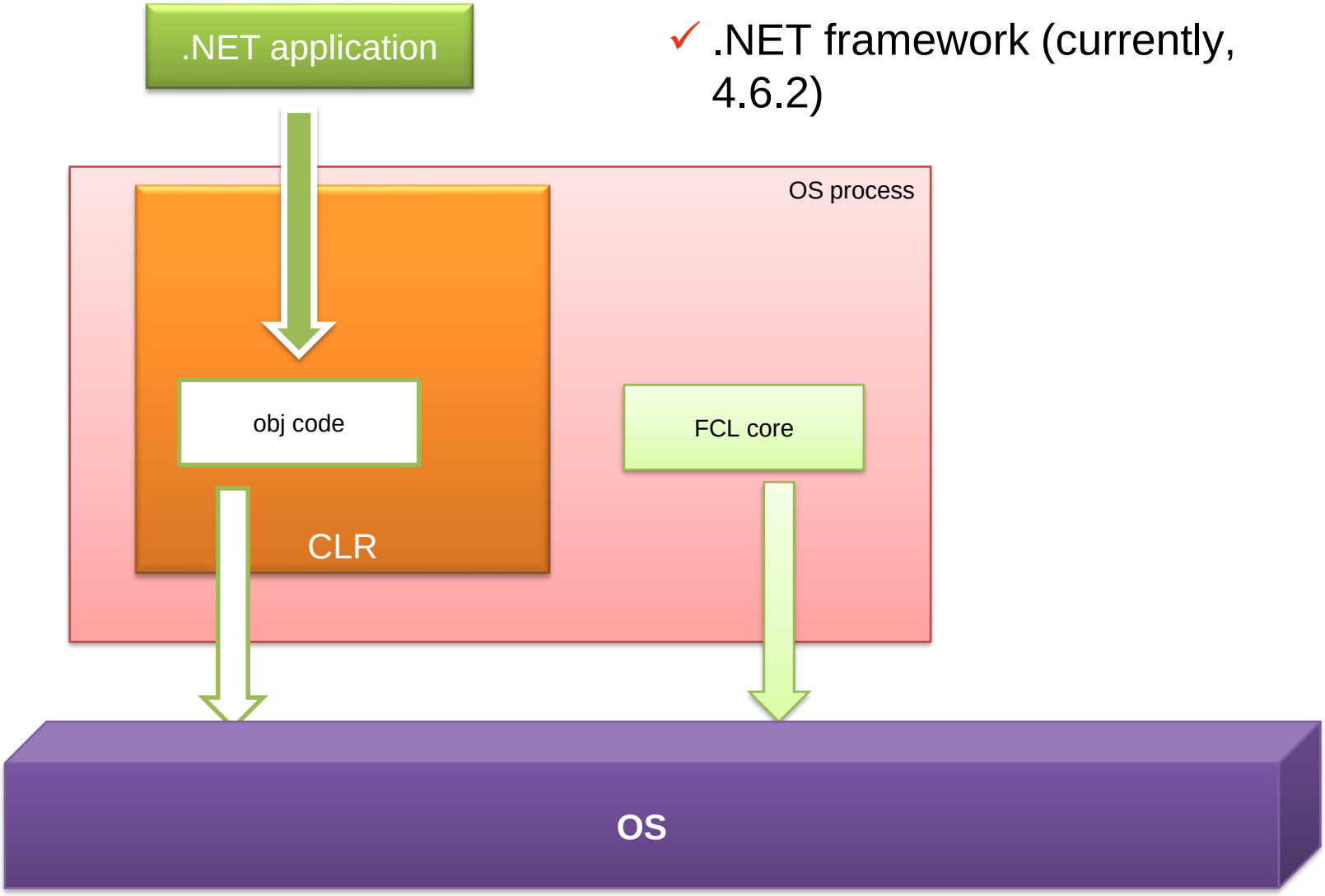
CLR Execution Model



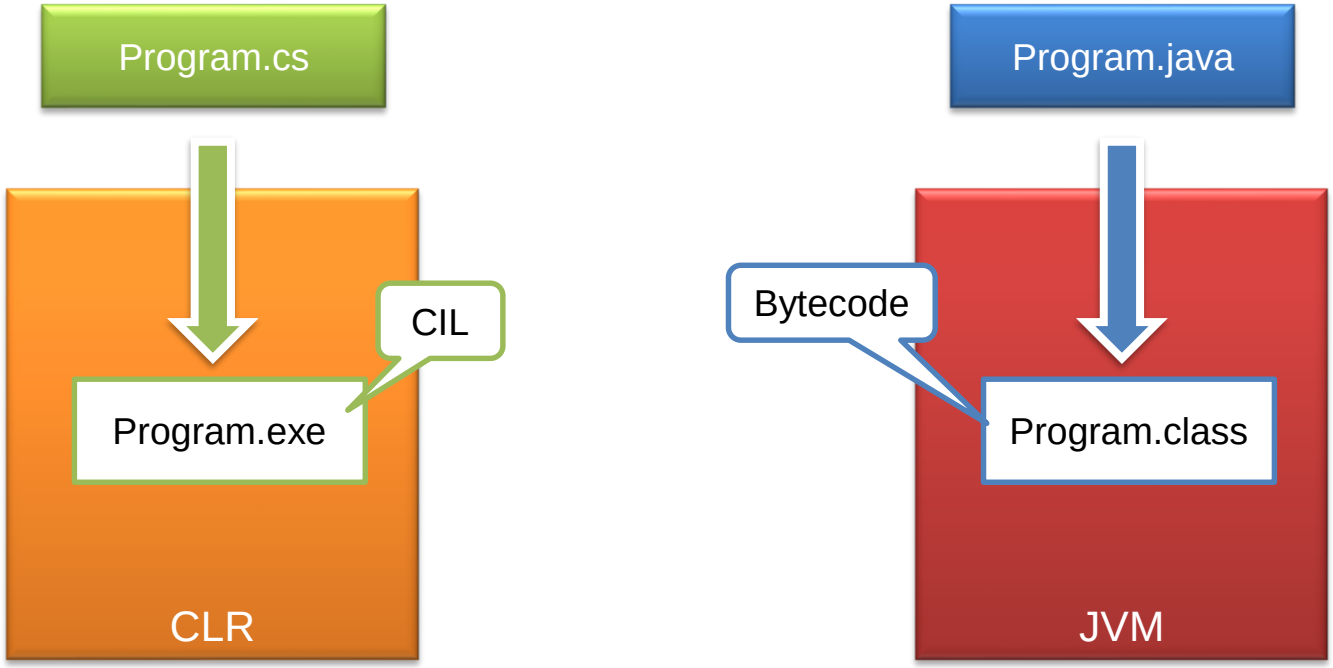
CLR Execution Model

CLR and FCL are distributed

- ✓ .NET framework (currently, 4.6.2)



Comparison to java



CLR and JIT compilation

"Just in time"

- ✓ All .NET languages compile to the same CIL
- ✓ The CLR transforms CIL to a given architecture ASM
- ✓ Highly-optimized process

Pros

- ✓ Support for developer services (debugging)
- ✓ Interoperation between un/managed code
- ✓ Memory management and garbage collection
- ✓ Protected "sandbox" execution
- ✓ Access to metadata (type information - reflection)

Base Class Library

- ✓ Similar to Java's System namespace
- ✓ "Core" subset of FCL used by all .NET applications
- ✓ Provides IO, Threading, DB, text, GUI, console, sockets, web…

Java vs .NET platforms

.NET is 100% proprietary (although some components are open-source), while Java is more "open"

- ✓ Mono project to provide open .NET platform
- ✓ Visual Studio Code

In desktop app, Java struggled to establish its presence

- ✓ Swing, AWT (Abstract Window Toolkit), SWT (Standard Widget Toolkit)
- ✓ Also, Sun failed in promoting it properly as appealing choice for desktop products

.NET is the most popular choice for developing Windows apps

- ✓ Visual Studio Express and Community are free-of cost
- ✓ Windows doesn't ship with Java JRE; it ships with .NET

Server applications

.NET is highly integrated with Windows Server infrastructure

- ✓ E.g., integrated **Security**, Debugging, Deployment…

Server applications

- ✓ Java EE vs. ASP.NET (which is however not part of standard CLI)
- ✓ Of the top 1,000 websites, approximately 24% use ASP.NET and also 24% use Java, whereas of all the websites approximately 17% use ASP.NET and 3% use Java
  (http://w3techs.com/technologies/cross/programming_language/ranking)

..and mobile?

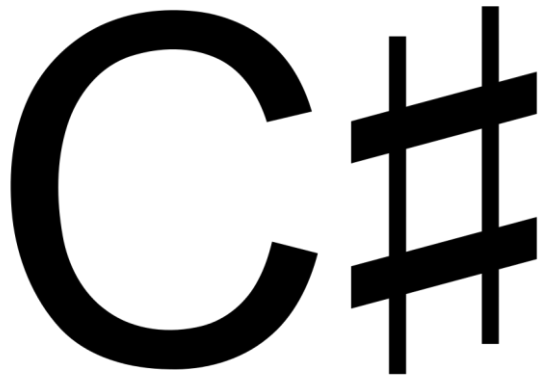
Android is based on Java

- ✓ It is also possible to develop in C (NDK)

Microsoft recently acquired Xamarin

- ✓ Cross-(mobile)platform IDE + libraries for c#
- ✓ Integrated in VS 2017

The C# language



See sharp?

A bit of history…

- ✓ Born during the development of the .NET Framework, in 1999
- ✓ Team lead by Anders Hejlsberg
- ✓ Initially called "C-like Object Oriented Language" (Cool)
- ✓ When MS announced .NET project, the language had been renamed C#, and the class libraries and ASP.NET runtime had been ported to it

- ✓ Hejlsberg is C#'s principal designer and lead architect at BigM
- ✓ "Flaws in most major programming languages (e.g. C++, Java, Delphi, and Smalltalk) drove the fundamentals of the Common Language Runtime (CLR), which, in turn, drove the design of the C# language itself"

Design goals

- ✓ "Simple, modern, general-purpose, object-oriented"
- ✓ "Provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. Software robustness, durability, and programmer productivity are important."
- ✓ Should support software for distributed environments.
- ✓ Portability (C/C++), internationalization
- ✓ No need to compete directly on performance and size with C or assembly language.

I've been talking too much…

- ✓ …let's see it in action!

Compilation, preprocessing & packaging

Features

- ✓ Conditional compilation via preprocessor directives
- ✓ Roslyn: compiler as-a-service
- ✓ Namespaces and assemblies (vs. Java Packages)
- ✓ Free file contents / folder structure compared to Java

Visual Studio

Overview

- ✓ Solutions & projects
  - References, shared projects
  - Publishing
- ✓ Remote debugging
  - Azure
- ✓ Config files
  - Publishing, transformations
- ✓ Versioning
  - Git, TFS

Data types

Overview & differences

Strongly typed - more type safe than C++

- ✓ No non-safe implicit conversions
- ✓ Object-oriented (Like Java)

Local variables cannot shadow variables in enclosing non-classes blocks

- ✓ Unlike any other lang.

Arithmetic overflow automatically handled

- ✓ `checked`/`unchecked` (**statically** check blocks for arithmetic overflow)

Why are arithmetic overflow not nice?

ARIANE 5 FAILURE

- BACKGROUND:-
- European space agency's re-useable launch vehicle.
- Ariane-4 was a major success
- Ariane -5 was developed for the larger payloads

- LAUNCHED:-on June 4 1996
- MISSION was to delivered $500 million payloads to the orbit
- THE MAIDEN FLIGHT OF THE ARIANE 5 ENDED IN A FAILURE.
- ONLY AFTER 40 SECONDS THE FLIGHT VEERED OFF ITS PATH AND BROKE UP AND EXPLODED
- CAUSE: Unhandled floating point exception in code
- ENGINEERS FROM THE ARIANE PROJECT STARTED TO INVESTIGATE THE CAUSES OF LAUNCH FAILURE.

Features

- ✓ Base types and unified type system (unlike Java)
- ✓ Structs
- ✓ Arrays and collections

- ✓ Pointers and references
- ✓ Boxing/unboxing
- ✓ Implicit types
- ✓ Nullables
- ✓ Dynamic types

Syntax & keywords

Features

- ✓ `get` & `set` Properties
- ✓ `lock` (vs. `synchronized`)
- ✓ ~~`throws`~~ `try`/`catch`/`finally` (no unchecked exceptions in Java)
- ✓ `using` (vs try-with-resources)
- ✓ Expression bodies
- ✓ Generics
- ✓ Delegates (later…)

OOP

Features

- ✓ Classes, interfaces, abstract classes
- ✓ Members visibility
- ✓ Static
- ✓ Dynamic types
- ✓ Virtual/override/new/final methods
- ✓ Partial classes
- ✓ Properties (…)
- ✓ Events
- ✓ Operator overloading (!!)
- ✓ No `final` parameters
- ✓ Generic are not a type, but they are supported inside the runtime (reification)

Advanced features

(Advanced features)

- ✓ Attributes
- ✓ Functional programming
  - Closures
  - Delegates => predicates, anonymous, lambda functions
- ✓ Threading & async/await (from C# 5)
- ✓ Language-Integrated Query (LINQ)

References

- ✓ Slides will be provided to prof. Cabri
  - If you haven'd had enough..
- ✓ My contacts
  - paolo.burgio@unimore.it
  - http://hipert.mat.unimore.it/people/paolob/
- ✓ Useful links
  - http://www.google.com