



2. Platforma Microsoft .NET

Maciej Piechówka
macpi@eti.pg.gda.pl
2010

Spis treści:

2.1 Microsoft .NET Framework: CLR, podzespoły, biblioteka klas

2.2 Elementy języka C#, BCL, typy uogólnione, obsługa XML

2.3 Dostęp do danych za pomocą ADO.NET, LINQ, Entity Framework

2.4 Wytwarzanie aplikacji .NET: podstawy ASP.NET

2.5 Usługi Web

Literatura:

- Materiały firmy Microsoft, MSDN msdn.microsoft.com/pl-pl/default.aspx, www.codeguru.pl, <http://grupa-net.eti.pg.gda.pl/grupanet/>, zasoby Internetu ...
- Liberty J. *C# programowanie*, Helion, 2005, Jones A. *C# Księga przykładów*, Promise, 2005
- Peryy C. *Core C# i .NET*, helion 2006
- Connolly R. *ASP.NET 2.0. Projektowanie aplikacji internetowych*, Helion 2008
- Evjen B., S. Hanselman, D. Rader *ASP.NET 3.5 z wykorzystaniem C# i VB. Zaawansowane programowanie*, Helion 2010
- Matulewski J., *C# 3.0 i .NET 3.5. Technologia LINQ*, Helion, 2008
- Troelsen A. *Pro C# 2008 and the .NET 3.5 Platform*, Apress 2009, WNT 2009
- Wei-Meng Lee, *C# 2008. Warsztat programisty*, Helion 2010
- M. Włodarczyk *ITA-105 Programowanie obiektowe*, Microsoft, 2008
- CSharp Language Specification, Version 3.0, 2007, msdn.microsoft.com/en-us/vcsharp/aa336809.aspx
- CSharp tutorial www.csharp-station.com/Tutorial.aspx

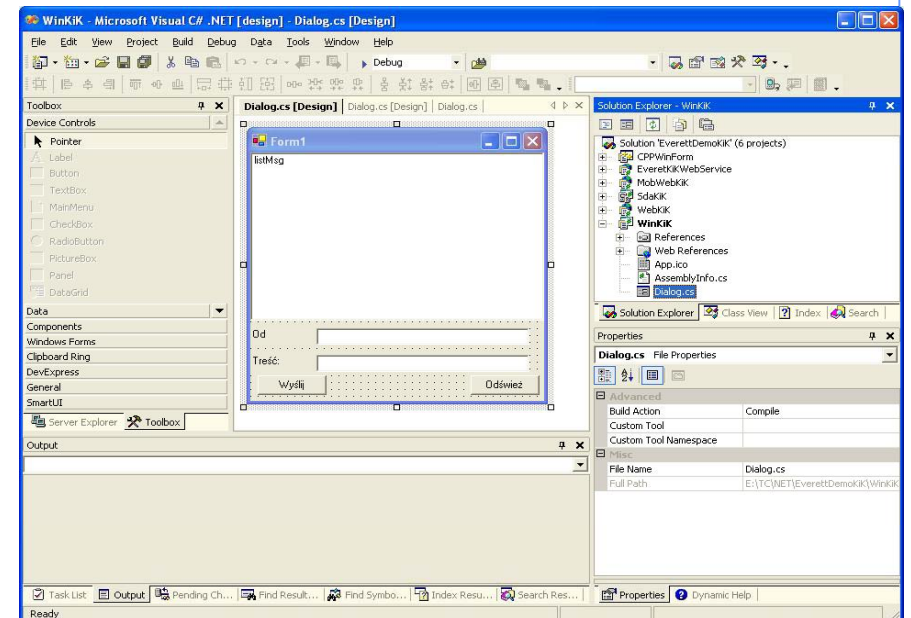


2.1 Platforma Microsoft .NET

.NET Framework - oferuje sposób wytwarzania oprogramowania w oparciu o niezwiązaną z konkretnym językiem programowania **platformę** składającą się z:

- **wspólnego systemu wykonawczego CLR** (*Common Language Runtime*)
- **biblioteki klas** (*Class Library*)
- **oraz dodatkowych serwerów i narzędzi.**

Tools for .NET Development: e.g. Visual Studio .NET 2008



.NET Enterprise Servers

Windows Server 2003/2008, BizTalk Server, Commerce Server, Content Management Server, Exchange Server, Host Integration Server, Internet Security and Acceleration Server, Operations Manager, Project Server, SharePoint Portal Server, Systems Management Server, SQL Server, ...

Windows 7, Vista, Windows XP, Windows CE,

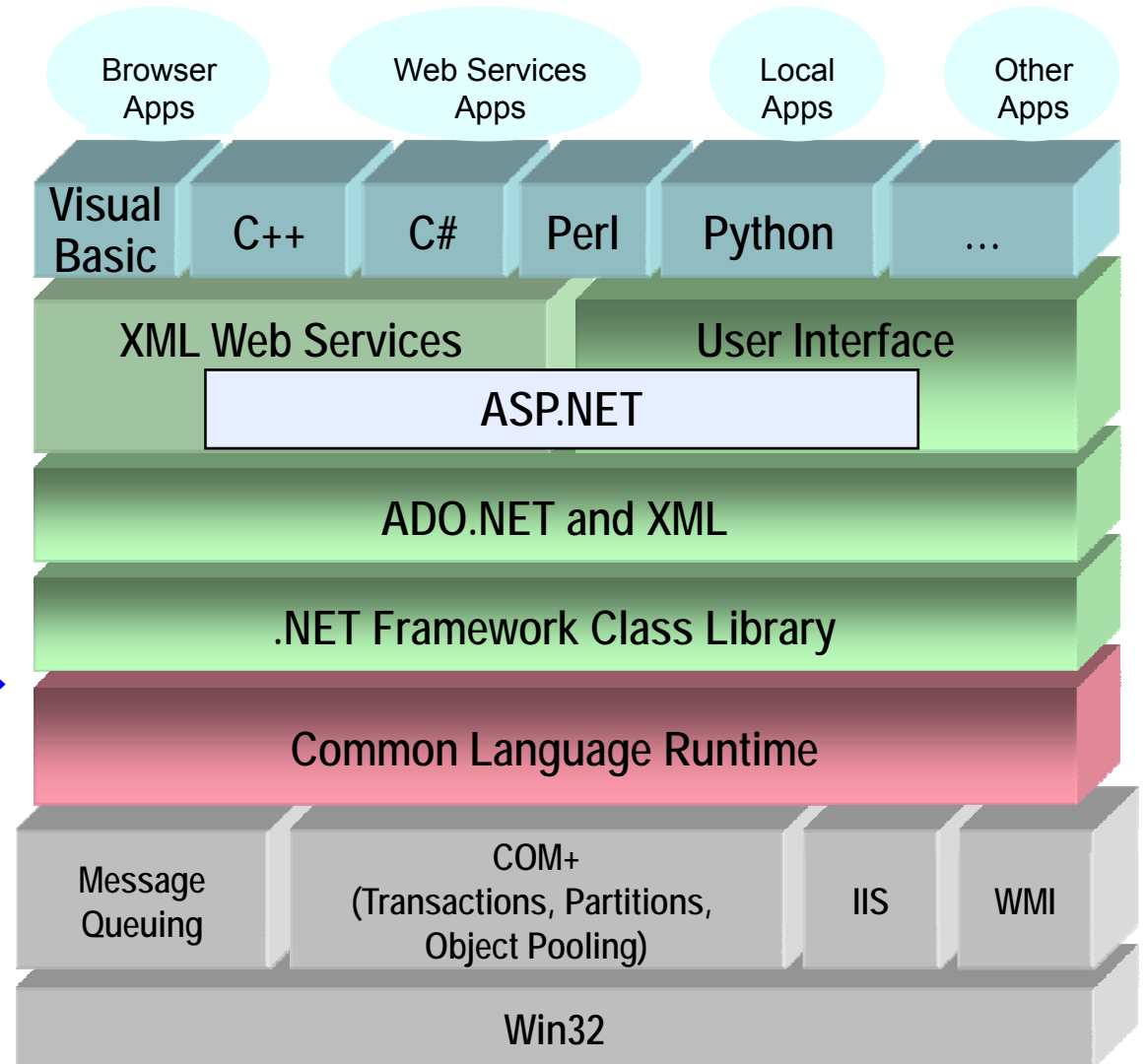




Elementy .NET 2.0 Framework

- **XML** (Extensible Markup Language)
 - Uniwersalna metoda opisywania danych
- **HTTP** (Hypertext Transport Protocol)
 - Standardowy protokół WWW, który można używać do współpracy między serwisami
- **SOAP** (Simple Object Access Protocol)
 - Bogaty mechanizm do interakcji (wywołania metod, argumenty itd.) między serwisami lub komponentami oparty o XML

Wspólne środowisko uruchomieniowe
+
biblioteka klas

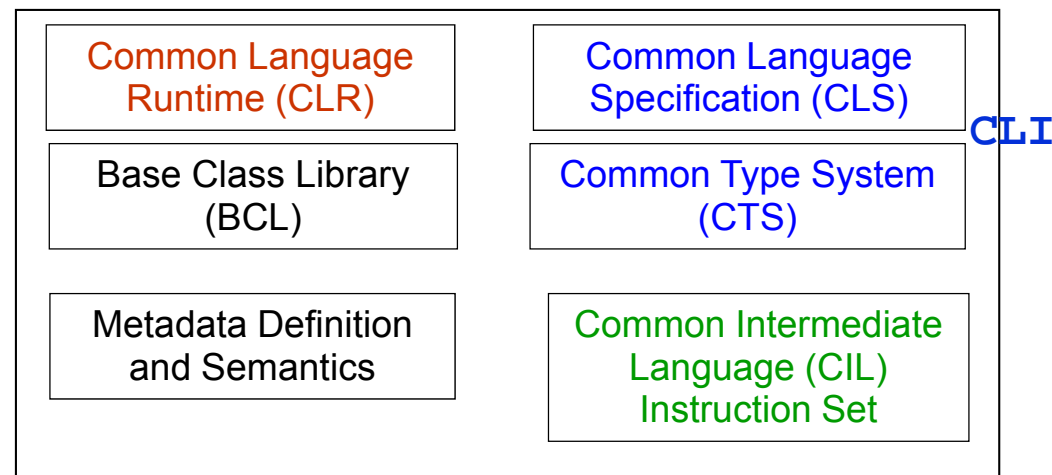


- **Web Services**
 - Standardowy sposób publikowania serwisów WWW i z nimi związanych mechanizmów, za pomocą których owe kontrakty można odnaleźć i zrozumieć



Platforma .NET

- **Common Language Runtime (CLR)** – środowisko wykonawcze odpowiedzialne za lokalizowanie, wczytywanie oraz zarządzanie typami .NET.
- **Common Language Infrastructure - ECMA-335**
 - **Common Type System (CTS)** - standard obejmujący sposób reprezentacji typów danych w pamięci komputera
 - **Common Language Specification (CLS)** - zbiór zasad i reguł, które musi spełnić język programowania aby współpracować z CLI
- **Common Intermediate Language (CIL)** - język pośredni (MSIL) składający się z niezależnego od procesora zestawu rozkazów, do którego kompilowane są wszystkie programy korzystające ze środowiska .NET
- **Metadane**

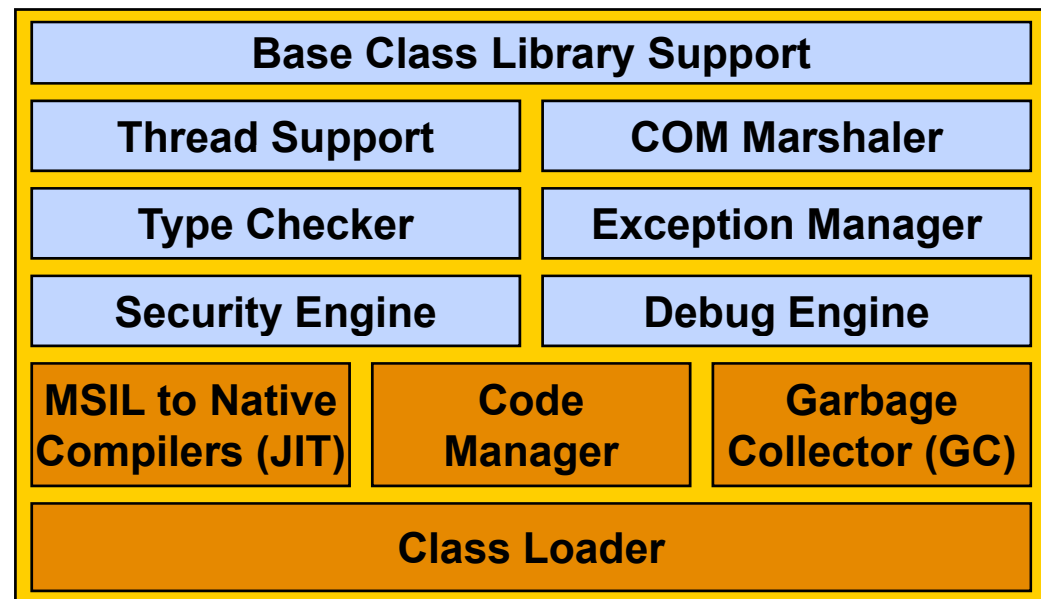




The Common Language Runtime (CLR)

Usługi CLR:

- Zarządzanie kodem (uruchamianie i nadzór nad jego wykonywaniem),
- Izolowanie obszarów pamięci przydzielonych poszczególnym aplikacjom,
- Konwersja języka pośredniego IL do kodu maszynowego,
- Zarządzanie pamięcią (w przypadku obiektów zarządzanych),
- Dostęp do metadanych (rozszerzonej informacji o typach),
- Stosowanie zabezpieczeń dostępu kodu do zasobów,
- Weryfikacja zgodności typów,
- Obsługa wyjątków i przekazywanie ich pomiędzy różnymi językami programowania,
- Obsługa współpracy pomiędzy kodem zarządzanym, obiektami COM i starszymi bibliotekami DLL,
- Automatyzacja tworzenia obiektów,
- Usługi związane z tworzeniem oprogramowania (debugowanie, profilowanie itp.)



CLI - Common Language Infrastructure - ECMA Standard 335
<http://www.ecma-international.org/publications/standards/ecma-335.htm>

CLR – implementacja MS standardu CLI,

Mono Project – http://www.mono-project.com/Main_Page, www.gnu.org/projects/dotgnu/



CTS: Common Type System

- Zdefiniowano wspólny zestaw typów danych:
CTS - Common Type System
- Typy są dzielone na kategorie:

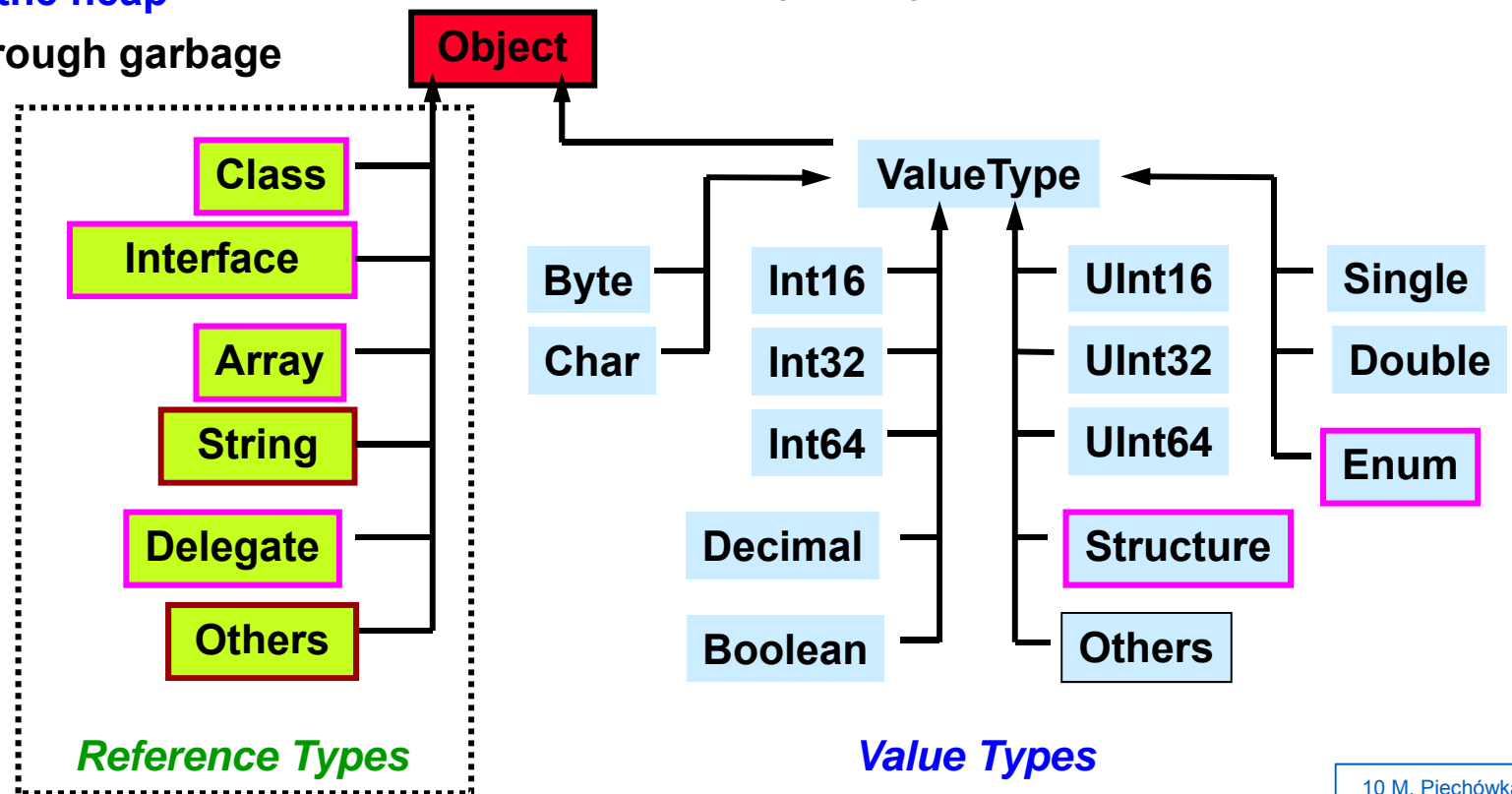
Dzięki temu, że wszystkie języki platformy .NET dzielą jeden wspólny system typów, mogą one korzystać wzajemnie ze swoich danych i w pełni się integrować

Reference types:

- More complex types
- Allocated on **the heap**
- Destroyed through garbage collection

Value types:

- Relatively simple types
- Typically allocated on **the stack**





C#'s type system

Category		Description
Value types	Simple types (numeric types, ...)	Signed integral: <code>sbyte</code> , <code>short</code> , <code>int</code> , <code>long</code>
		Unsigned integral: <code>byte</code> , <code>ushort</code> , <code>uint</code> , <code>ulong</code>
		Unicode characters: <code>char</code>
		IEEE floating point: <code>float</code> , <code>double</code>
		High-precision decimal: <code>decimal</code>
		Boolean: <code>bool</code>
	Enum types	User-defined types of the form <code>enum E { ... }</code>
Struct types	User-defined types of the form <code>struct S { ... }</code>	
Nullable types	Extensions of all other value types with a <code>null</code> value	
Reference types	Class types	Ultimate base class of all other types: <code>object</code>
		Unicode strings: <code>string</code>
		User-defined types of the form <code>class C { ... }</code>
	Interface types	User-defined types of the form <code>interface I { ... }</code>
	Array types	Single- and multi-dimensional, for example, <code>int[]</code> and <code>int[,]</code>
	Delegate types	User-defined types of the form e.g. <code>delegate int D(...)</code>



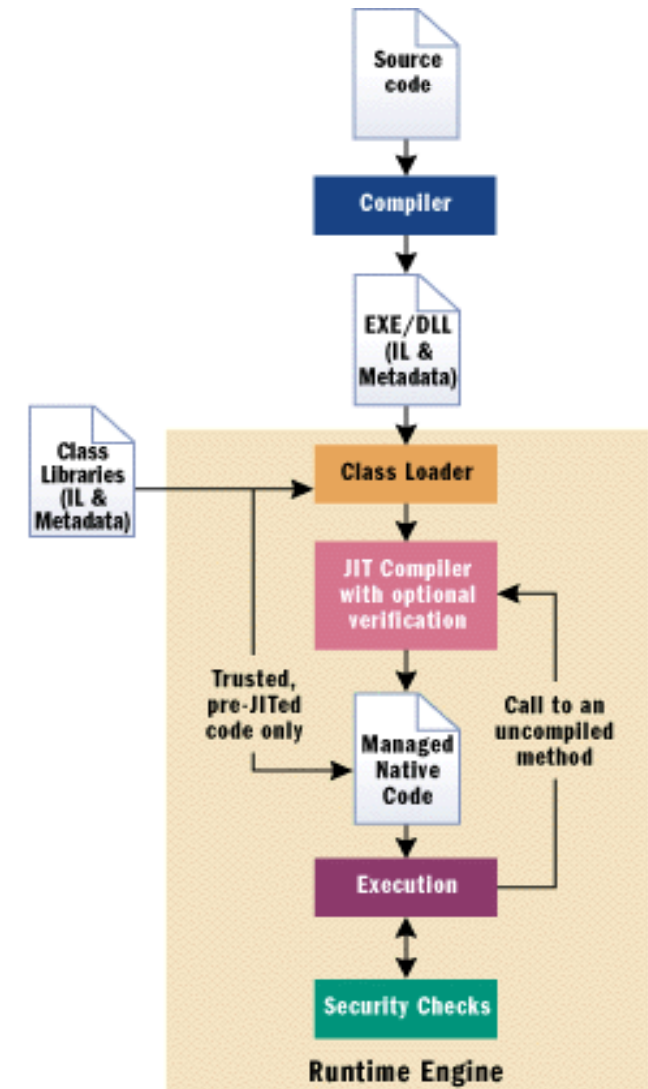
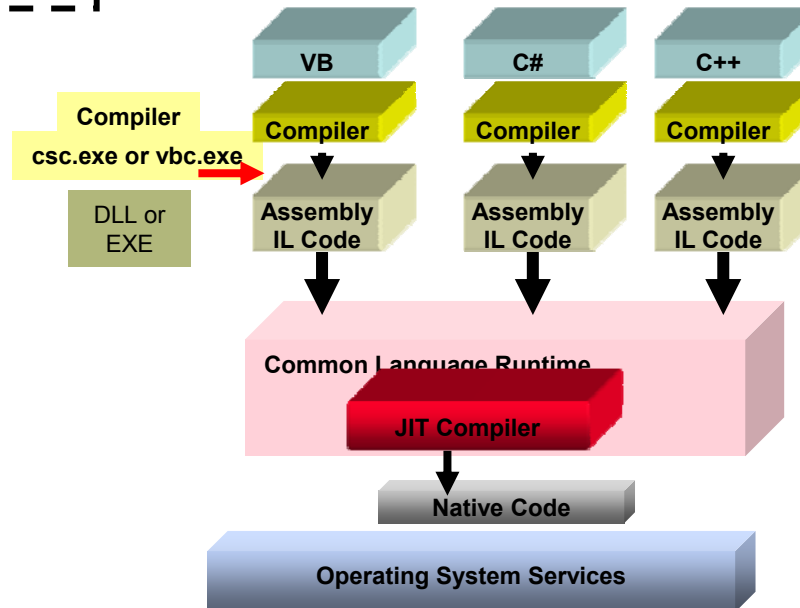
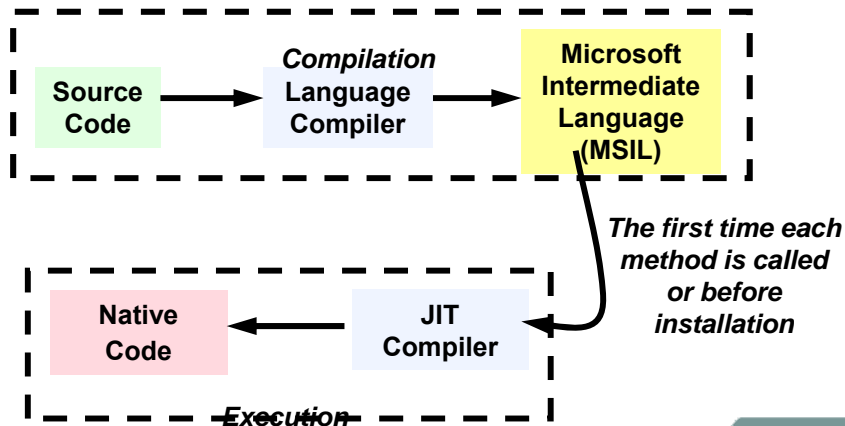
Język pośredni IL (Intermediate Language) Kod zarządzany (Managed Code)

- **Język pośredni (MSIL)** jest niezależnym od procesora zestawem rozkazów, do którego kompilowane są wszystkie programy korzystające ze środowiska .NET.
 - Zawiera on instrukcje niezbędne do ładowania, przechowywania i inicjalizowania obiektów oraz do wywoływania ich metod. Wraz z **metadanymi** i wspólnym systemem typów, MSIL pozwala na pełną integrację pomiędzy różnymi językami programowania.
- Kod skompilowany do języka IL jest wejściem do kompilatora JIT (Just-In-Time Compilation) przekształcającego go w **kod maszynowy**.
- **Kod zarządzany** jest to kod zgodny ze specyfikacją CLR (CLS)
 - dostępność metadanych umożliwia realizację usług automatycznego zarządzania zasobami, przezroczystej współpracy fragmentów oprogramowania napisanych w różnych językach programowania, zarządzania uprawnieniami, zarządzania cyklem życia obiektów.
- Kod pisany w języku C# lub Visual Basic.NET jest domyślnie kodem zarządzanym.
- Zarządzane dane - są alokowane i zwalniane przez **proces odśmiecający (garbage collector)** działający w ramach CLR.



Kompilacja i wykonywanie kodu zarządzanego – .Net Assemblies

- kompilacją na żądanie - jedynie dla potrzebnych w danej chwili fragmentów
- możliwe jest skorzystanie z kompilacji podczas instalacji
 - Native Image Generator (NGEN) – *ngen.exe*





Just-in-Time (JIT)

- IL is compiled into processor-specific code at runtime

C#

```
Point pt;  
pt = new Point();  
pt.x = 200;  
pt.y = 300;
```

dev-time compile

IL

```
// Point pt = new Point();  
.locals init ([0] class Point pt)  
newobj instance void Point::.ctor()  
stloc.0  
  
// pt.x = 200;  
ldloc.0  
ldc.i4 0xc8  
stfld int32 Point::x  
  
// pt.y = 300;  
ldloc.0  
ldc.i4 0x12c  
stfld int32 Point::y
```

run-time compile

Intel x86

```
call FD5B0AD8 ; allocate  
mov ecx, eax ; ecx == pt  
call dword ptr ds:[003E5144h] ; pt.ctor  
mov dword ptr [ecx+4], 0C8h ; pt.x = 200  
mov dword ptr [ecx+8], 12Ch ; pt.y = 300
```



Podzespoły (*assemblies*) (1)

Assemblies (podzespoły, zestawy) - logiczne bloki, skompilowane do kodu pośredniego IL, z jakich buduje się aplikacje .NET; zawierają kod w języku IL oraz metadane; opatrzone numerem wersji i stanowiące spójną całość pakiety udostępniające określoną funkcjonalność.

Dwa rodzaje metadanych: opisujące podzespół jako jednostkę – **manifest** oraz **metadane** opisujące indywidualne typy w podzespole.

- **Manifest** jest częścią podzespołu, opisujący jego kod i zasoby poprzez określenie:
 - nazwy podzespołu (tożsamości podzespołu)
 - numeru wersji i kultury (informacje o regionach i językach obsługiwanych przez podzespół)
 - podpisu cyfrowego podzespołu (jeżeli podzespół ma być używany przez różne aplikacje - *strong name*)
 - plików składających się na podzespół
 - typów i zasobów składających się na podzespół, wraz z informacją, które z nich są eksportowane,
 - powiązania z innymi podzespołami (np. *mscorlib.dll*),
 - zestawu uprawnień niezbędnych do poprawnego działania podzespołu

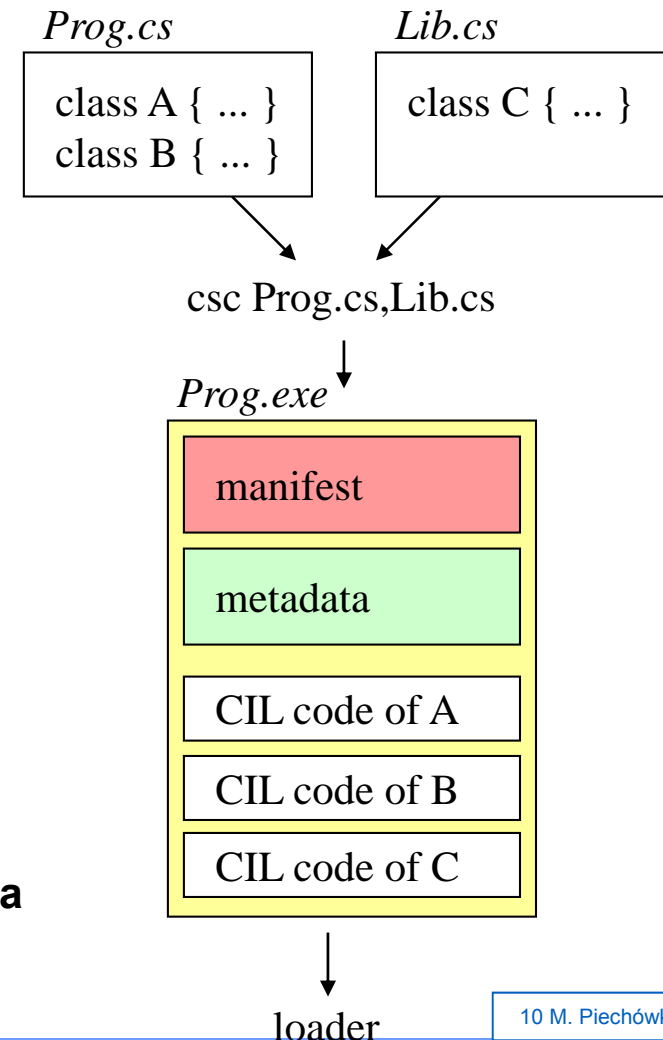
Definiowanie podzespołu: Assembly Generation Utility - Al.exe, IDE
Odczyt zawartości podzespołu: Disassembler – ildasm.exe



Podzespoły (*assemblies*): Metadane

Metadane typów - to szczegółowy opis typów zdefiniowanych w kodzie zarządzanym, z którym są związane. Opis ten przechowywany jest w tym samym pliku, w którym znajduje się kod MSIL. Informacje te zawierają:

- **Opis typów**
 - nazwę typu,
 - zasięg typu (publiczny lub w granicach podzespołu),
 - nazwę typu, po którym dziedziczy opisywany typ,
 - implementowane interfejsy,
 - implementowane metody,
 - udostępniane właściwości,
 - obsługiwane zdarzenia.
- **Atrybuty** – dodatkowe informacje o elementach kodu - adnotacje
 - atrybuty mają określone nazwy i funkcje, definiowane przez różne części środowiska .NET Framework
 - mogą one służyć do sterowania sposobem wykonywania kodu, opisać wymagania bezpieczeństwa



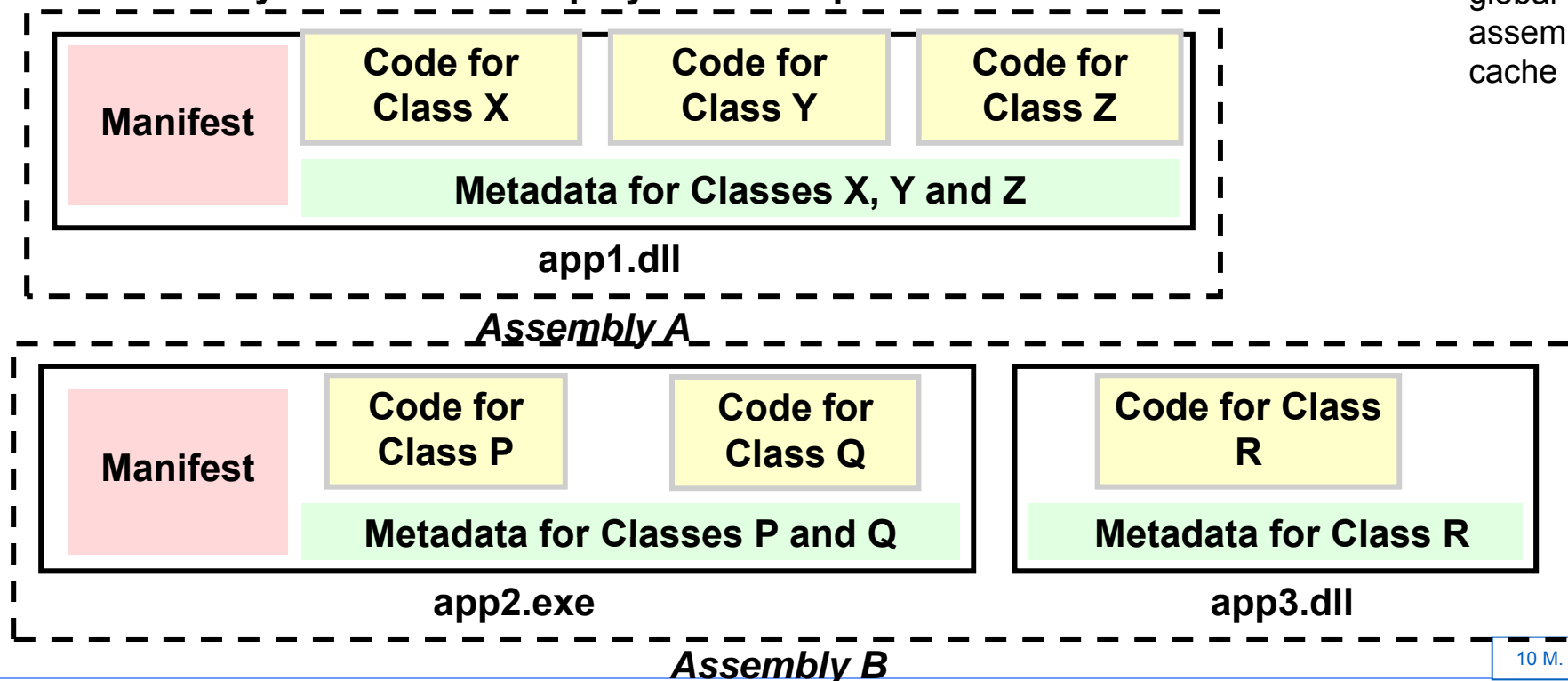


Podzespoły (assemblies) (2)

- Po skompilowaniu do kodu IL, **zarządzany kod** jest częścią podzespołu
 - podzespół zawiera jedną lub wiele DLLs or EXEs
- Podzespoły mogą być używane jako **prywatne** albo **współdzielone**
- Podzespół prywatny jest instalowany poprzez kopiowanie do folderu
 - nie wymagane jest umieszczenie w rejestrze
- Są podstawą bezpieczeństwa kodu
 - Pobierany kod może mieć przydzielone uprawnienia

+ strong name

GAC –
global
assembly
cache





Podzespoły - wdrażanie, obsługa wersji (3)

Wdrażanie

- Umieszczanie podzespołu w katalogu aplikacji (...\.bin) - nie wymagane żadne wpisy do rejestru
- Współużytkowanie podzespołów – umieszczenie ich w globalnej pamięci podręcznej podzespołów **GAC** (*global assembly cache*)
 - \WINNT\assembly, \WINDOWS\assembly, GACUTIL.exe – SDK
 - Rozróżnianie podzespołów – poprzez silną nazwę (*strong name*)
 - generowana z wykorzystaniem kluczy kryptograficznych (*sn.exe*) i nazwy podzespołu
 - Podpis cyfrowy podzespołu – możliwość sprawdzenia integralności przy ładowaniu podzespołu

Usunięcie problemu „piekła dll-i”

Wersja podzespołu – charakteryzowana przez:

- numer kompatybilnościowy wersji (*compability version*) składający się z numerów:
 - główny (*major*), drugorzędny (*minor*), kompilacji (*build*), korekty (*revision*) np. 2.1.1243.0
- wersję informacyjną (*information version*) – ciąg znaków czytelnych dla ludzi
- numery wersji zewnętrznych podzespołów od których jest zależny



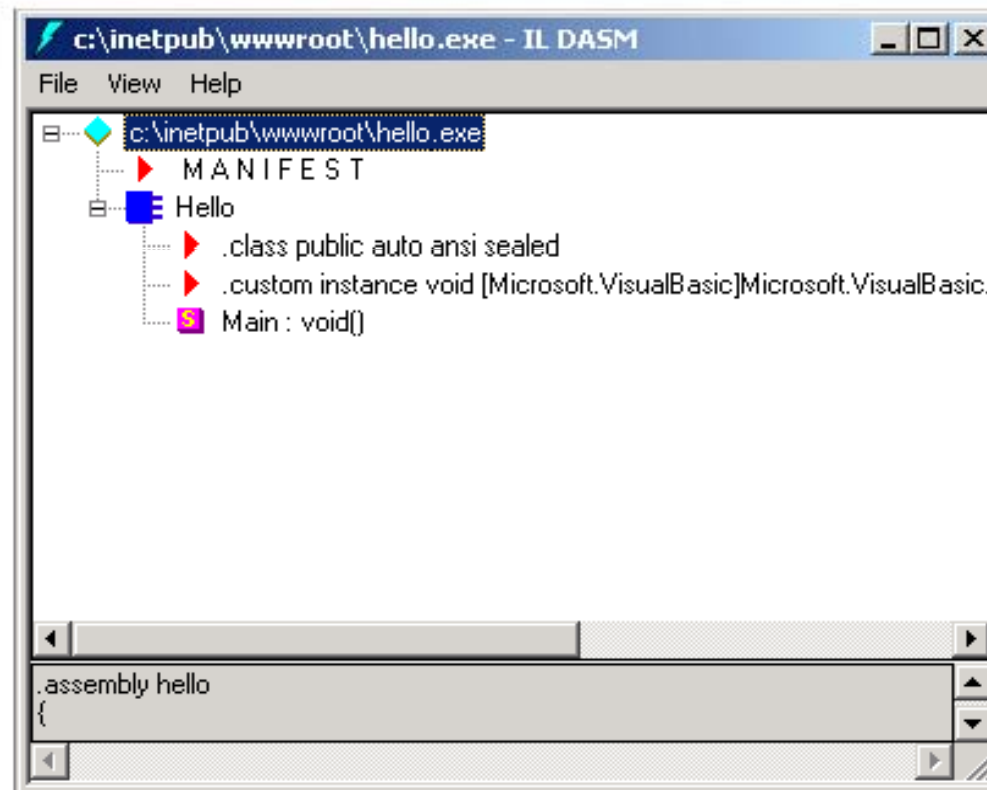
Zawartość folderu GAC

Global Assembly Name	Type	Version	Culture	Public Key Token
Accessibility		1.0.5000.0		b03f5f7f11d50a3a
ADODB		7.0.3300.0		b03f5f7f11d50a3a
ConMan		7.0.5000.0		b03f5f7f11d50a3a
ConManDataStore		7.0.5000.0		b03f5f7f11d50a3a
ConManServer		7.0.5000.0		b03f5f7f11d50a3a
CRVsPackageLib		9.1.5000.0		692fba5521e1304
CrystalDecisions.CrystalReports.Engine		9.1.5000.0		692fba5521e1304
CrystalDecisions.ReportSource		9.1.5000.0		692fba5521e1304
CrystalDecisions.Shared		9.1.5000.0		692fba5521e1304
CrystalDecisions.Web		9.1.5000.0		692fba5521e1304
CrystalDecisions.Windows.Forms		9.1.5000.0		692fba5521e1304
CrystalEnterpriseLib		9.1.5000.0		692fba5521e1304
CrystalInfoStoreLib		9.1.5000.0		692fba5521e1304
CrystalKeyCodeLib		9.1.5000.0		692fba5521e1304
CrystalPluginMgrLib		9.1.5000.0		692fba5521e1304
CrystalReportPluginLib		9.1.5000.0		692fba5521e1304
cscompmgd		7.0.5000.0		b03f5f7f11d50a3a
CustomMarshalers	Native Images	1.0.5000.0		b03f5f7f11d50a3a
CustomMarshalers	Native Images	1.0.5000.0		b03f5f7f11d50a3a
CustomMarshalers		1.0.5000.0		b03f5f7f11d50a3a
emucm		1.0.0.0		b03f5f7f11d50a3a
EnvDTE		7.0.3300.0		b03f5f7f11d50a3a
Extensibility		7.0.3300.0		b03f5f7f11d50a3a
extractsdk		7.0.5000.0		b03f5f7f11d50a3a
IEExecRemote		1.0.5000.0		b03f5f7f11d50a3a



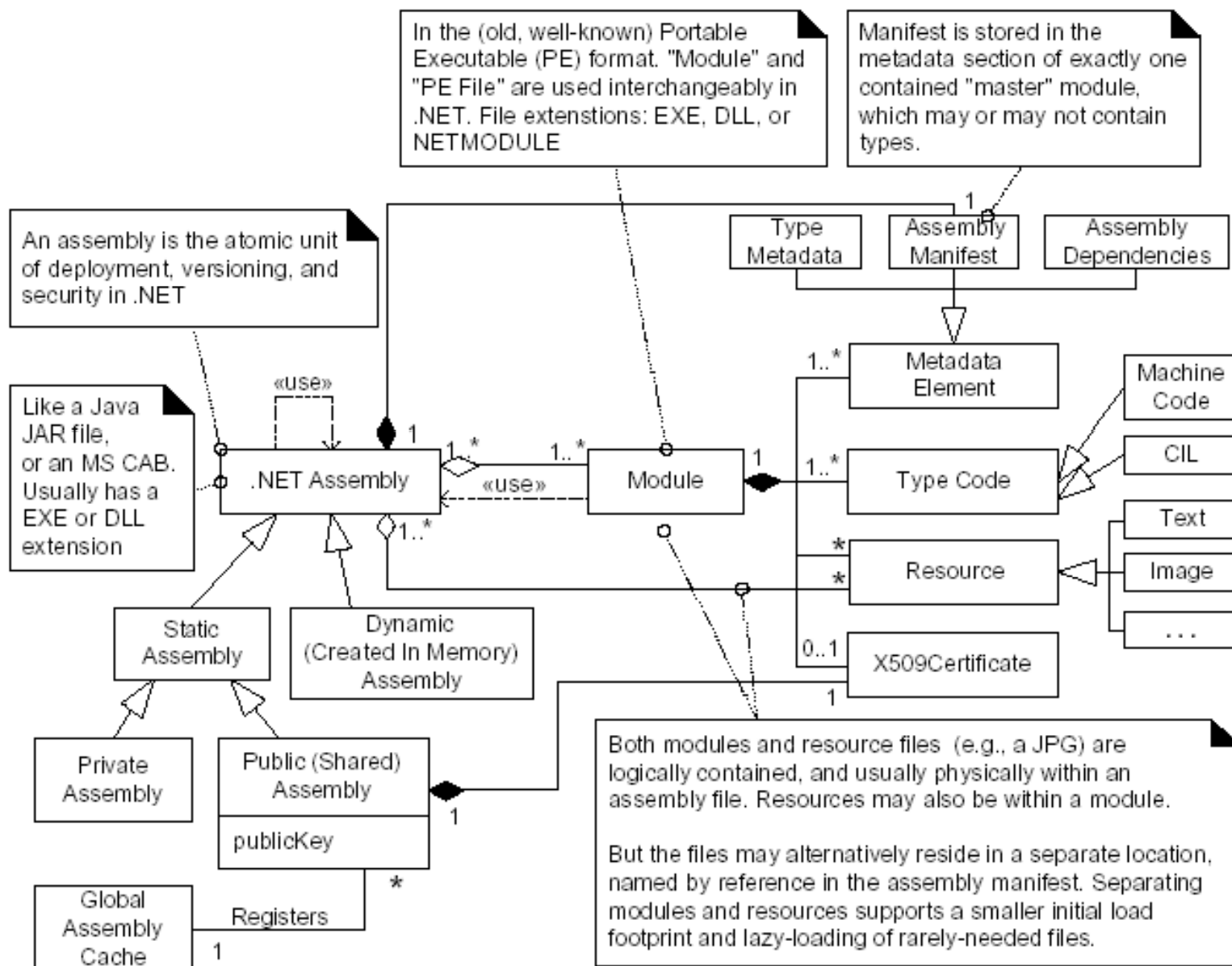
Using the ILDASM to View the Assembly and Classes

	Meaning
▶	More info
■	Namespace
■	Class
■	Value type
I	Interface
■	Method
S	Static method
◆	Field
S	Static field
▼	Event
▲	Property





Kod zarządzany: Model podzespołu





Elementy .NET Framework 3.5

.Net Framework 4.0

.NET Framework 3.5

LINQ

ASP.NET 3.5

CLR Add-in
Framework

Dodatkowe
rozszerzenia

.NET Framework 3.0 + SP1

Windows
Presentation
Foundation

Windows
Communication
Foundation

Windows
Workflow
Foundation

Windows
CardSpace

.NET Framework 2.0 + SP1

VS2008



.NET Framework 3.5

Windows Presentation Foundation

- Umożliwia łatwe łączenie tekstu, grafiki i multimediiów (animacja, dźwięk, wideo)
- Budowanie interfejsów przy pomocy deklaratywnego języka opisu wyglądu aplikacji XAML (Extensible Application Markup Language)
- Wsparcie grafiki wektorowej
- Wykorzystanie akceleracji sprzętowej
- Dwa modele aplikacji:
 - zwykła (desktop), uruchamiana w przeglądarce – *XBAP (XAML Browser Applications)*

LINQ - Language Integrated Query

- integruje język zapytań bazodanowych bezpośrednio w językach programowania .NET do
 - LINQ to Objects, LINQ to DataSets, LINQ to SQL, LINQ to Entities, LINQ to XML

Windows Communication Foundation

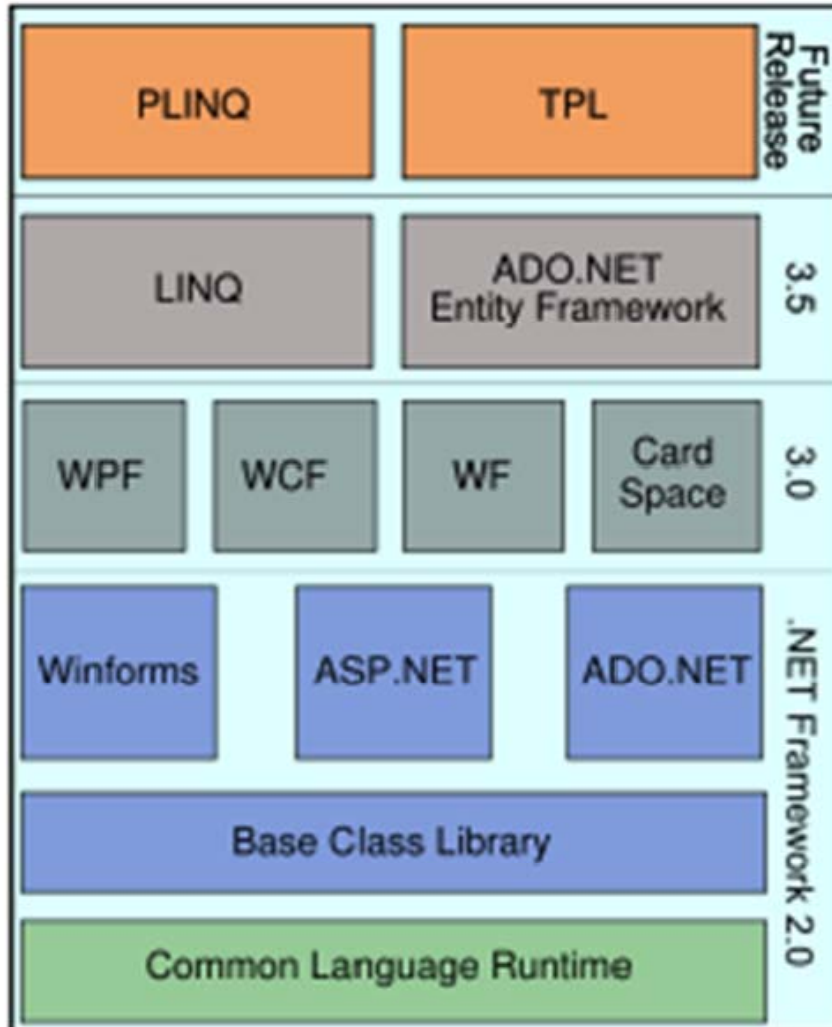
- WCF – uniwersalny model programowania aplikacji oparty o usługi
- Wspólny komponent dla różnych technologii związanych z komunikacją:
 - COM/COM+, .NET Remoting, Usługi, ASP.Net Webservices, Kolejowanie wiadomości
- Wsparcie dla ASP.Net/Ajax
- Wsparcie dla syndykacji (RSS/Atom)
- Integracja między WF i WCF

Windows Workflow Foundation

- Workflow – grupa czynności
- Framework do budowy przepływów prac
 - Podstawowe Activity, Persistence Service, Tracking, Synchronizacja



.Net Framework 4.0



- **Parallel LINQ (*PLINQ*)**
- **Task Parallel Library (*TPL*)**
- **Support for IronPython, IronRuby, and F#**
- **Support for Code Contracts**



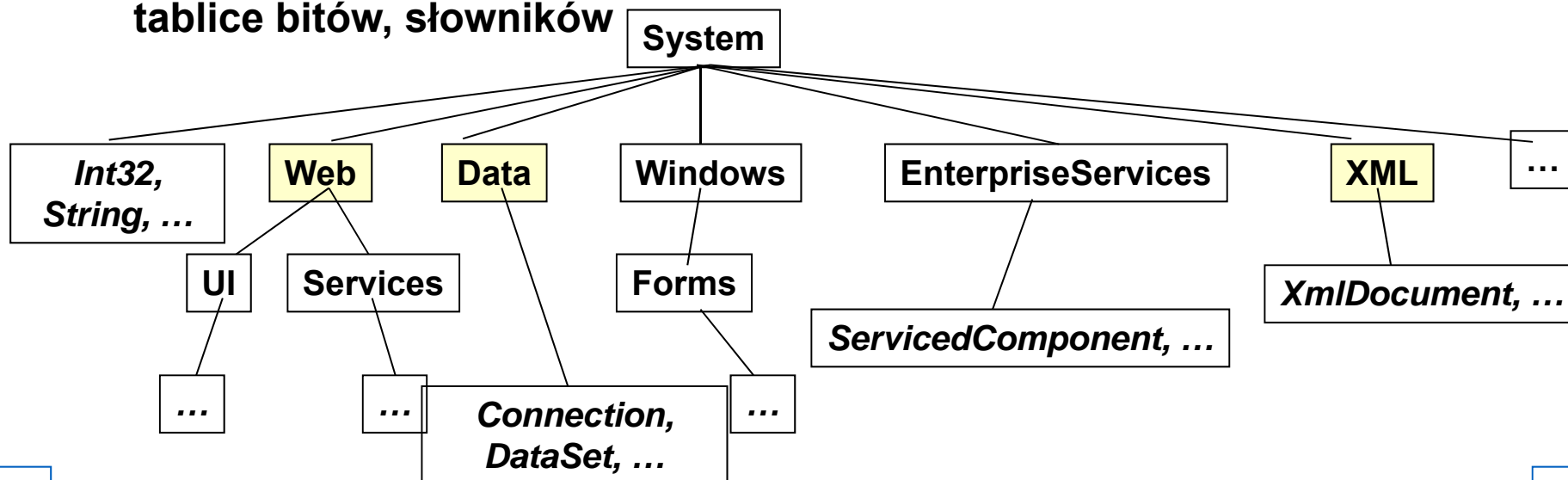
Biblioteki klas

Framework Class Libraries (1)

- Elementy biblioteki klas .NET Framework, pogrupowane są za pomocą hierarchicznej struktury **przestrzeni nazw (namespace)**
 - ok. 100 namespaces with about 2000 types
- Umożliwiają dostęp do usług systemowych i zostały zaprojektowane tak, aby stanowiły podstawę do tworzenia aplikacji, komponentów i kontrolki .NET
- Przestrzenie nazw :

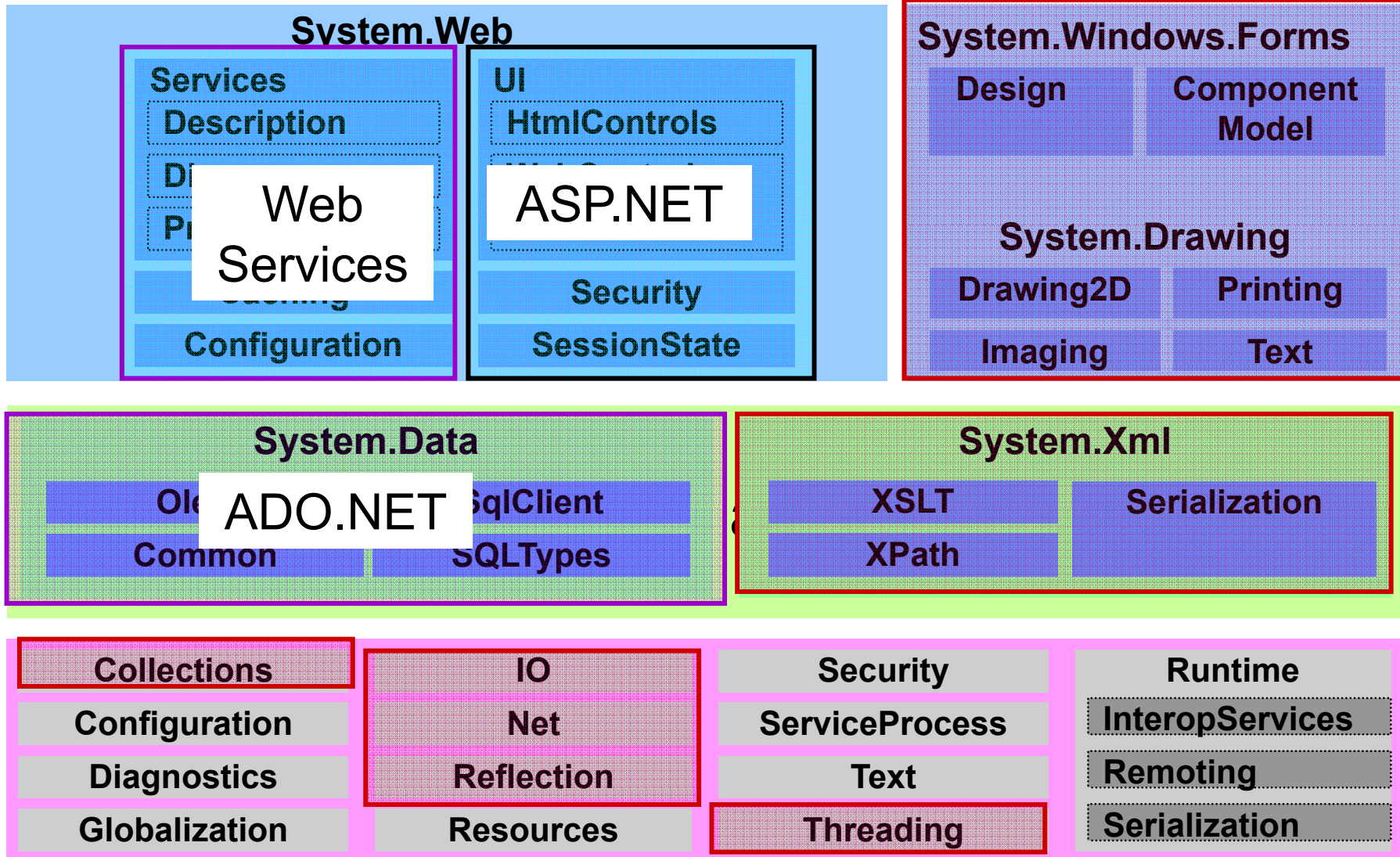
System - zawiera klasy, które definiują najczęściej używane typy danych, zdarzenia, procedury obsługi zdarzeń, interfejsy, atrybuty i wyjątki.

System.Collections - zawiera klasy obsługi kolekcji obiektów - list, kolejek, tablice bitów, słowników





.NET Class Library





Biblioteka klas .NET

Przestrzeń nazw	Opis
System	Podstawowe klasy, które definiują najczęściej używane typy danych, zdarzenia, procedury obsługi zdarzeń, interfejsy, atrybuty i wyjątki. Inne klasy w tej przestrzeni nazw udostępniają usługi związane z konwersją typów danych, manipulowaniem parametrami metod, obliczeniami matematycznymi, zdalnym i lokalnym wywoływaniem programów, zarządzaniem środowiskiem aplikacji, oraz nadzorowaniem zarządzanego i niezarządzanego kodu
System.CodeDom	Reprezentacja elementów i struktury kodu źródłowego dokumentu
System.ComponentModel	Tworzenie komponentów i kontrolek, rejestracja i adaptacja w czasie projektowania
System.Collections, System.Collection.Generic	Definiują różne kolekcje obiektów - listy, kolejki, tablice bitów, tablice skrótów (ang. hashtables) i słowniki (ang. dictionaries).
System.Configuration	Programowy dostęp do ustawień konfiguracyjnych .NET Framework i podzespołów oraz obsługa błędów w plikach konfiguracyjnych (pliki *.config) i tworzenie własnych instalatorów.
System.IO	Podstawowy dostęp i zarządzanie strumieniami danych. Zawiera typy umożliwiające synchroniczny i asynchroniczny odczyt i zapis do strumieni danych i plików.
System.Text	Kodowanie znaków, konwersje znaków i manipulacje łańcuchami znaków
System.TextRegular Expression	Obsługa wyrażeń regularnych



Biblioteka klas .NET - Dane

Dane	
System.Data	Dostęp i zarządzanie danymi i źródłami danych. Architektura dostępu do danych ADO.NET, umożliwia tworzenie komponentów, które pozwalają na wydajne zarządzanie danymi pochodzącymi z różnych źródeł.
System.Data.Common	Klasy współdzielone przez dostawców danych .NET Framework. Dostawca danych - to zbiór klas pozwalających na korzystanie z danych pochodzących z określonego źródła, np. bazy danych.
System.Data.ODBC	Klasy składające się na dostawcę danych dla interfejsu ODBC (.NET Framework Data Provider for ODBC)
System.Data.OleDb	Klasy składające się na dostawcę danych dla interfejsu OleDb (.NET Framework Data Provider for OleDb)
System.Data.SqlClient	Klasy składające się na dostawcę danych dla serwera MS SQL Server (.NET Framework Data Provider for SQL Server).
System.Data.SqlTypes	Klasy reprezentujące rodzime typy danych serwera Microsoft SQL Server. Pozwalają one na bezpieczniejsze i wydajniejsze korzystanie z danych pochodzących z tego serwera.
System.Linq	Typy używane w programowaniu dostępu do danych za pomocą języka wyrażeń zapytań
System.XML	Zgodna ze standardami obsługa przetwarzania XML'a
System.XML.Serialization	Odwzorowanie obiektów XML i odwrotnie



Biblioteka klas .NET

Sieć

System.Net	Udostępnia proste interfejsy programistyczne dla wielu protokołów używanych we współczesnych sieciach.
------------	--

System.Net.Sockets	Udostępnia zarządzaną implementację interfejsu Windows Sockets.
--------------------	---

Usługi infrastruktury działania

System.Runtime	Przestrzeń nazw obsługuje funkcje takie jak serializacja, remoting, interoperacyjność.
----------------	--

System.Runtime.InteropServices	Udostępnia różnorodne klasy obsługujące współpracę z modelem COM i wywoływanie usług systemowych.
--------------------------------	---

System.Runtime.Remoting	Wsparcie dla tworzenia ściśle lub luźno powiązanych aplikacji rozproszonych
-------------------------	---

System.Runtime.Serialization	Klasy używane do serializowania i deserializowania obiektów. Proces serializacji polega na konwersji obiektu lub całego grafu obiektów do postaci liniowej sekwencji bajtów, którą można następnie zachować lub przesłać do innej lokalizacji. Deserializacja jest procesem odwrotnym.
------------------------------	--

System.Threading	Klasy i interfejsy umożliwiające programowanie wielowątkowe.
------------------	--

Bezpieczeństwo

System.Security	Dostęp do podsystemu zabezpieczeń .NET Framework, zarządzanie zasadami, zezwolenia, sprawdzanie stosu
-----------------	---

System.Security.Cryptography	Usługi kryptograficzne, w tym bezpieczne kodowanie i dekodowanie danych, tworzenie skrótów jednokierunkowych, generowanie liczb losowych i uwierzytelnianie komunikatów.
------------------------------	--



Biblioteka klas .NET

Usługi platformy

System.Diagnostics	Obsługa i diagnostyka aplikacji; interakcje z procesami systemowymi, dziennikami zdarzeń i licznikami wydajności. W tej przestrzeni nazw znajdują się klasy umożliwiające debugowanie aplikacji i śledzenie wykonywania kodu.
System.DirectoryServices	Umożliwia wygodny dostęp do Active Directory z poziomu kodu zarządzanego.
System.Management	Narzędzia zarządzania usługami i aplikacjami zgodnie z WBEM (Web Based Enterprise Management)
System.Messaging	Dostęp i zarządzanie MSMQ, wysyłanie i odbieranie wiadomości
System.ServiceProcess	Instalacja i wykonywanie usługowych Windows
System.Timers	Zawiera komponent Timer umożliwiający wywoływanie zdarzenia w określonych odstępach czasu.

Odbicia

System.Reflection	Dostęp do metadanych typów, podzespółów oraz dynamicznego tworzenia typów i odwoływania się do nich
-------------------	---

Globalizacja i lokalizacja

System.Globalization	Wsparcie dla lokalizacji i globalizacji (wielojęzykowości aplikacji) kodu i zasobów
System.Resources	Zarządzanie i dostęp do zasobów



Biblioteka klas .NET - usługi WEB

Usługi Web	
System.Web	Wsparcie dla zarządzania, komunikacji i projektowania serwerów i klientów oraz usług internetowych XML Web Services; m.i.: manipulacje plikami cookie, transfer plików, obsługę wyjątków, zarządzanie buforem wyjściowym, stanem sesji.
System.Web.Caching	Kasy pozwalające na buforowanie często używanych danych po stronie serwera.
System.Web.Configuration	Klasy pozwalające na odczyt i modyfikację konfiguracji ASP .NET
System.Web.Mail	Klasy umożliwiające tworzenie i wysyłanie wiadomości email
System.Web.Security	Klasy używane do implementowania zabezpieczeń ASP .NET w aplikacjach sieci Web.
System.Web.Services	Klasy umożliwiające tworzenie usług webowych XML Web Services oraz klientów tych usług.
System.Web.SessionState	Klasy i interfejsy umożliwiające przechowywanie danych specyficznych dla poszczególnych klientów aplikacji ASP .NET. Informacje o stanie sesji dają użytkownikowi wrażenie stałego połączenia z aplikacją. Informacje o stanie sesji mogą być przechowywane w lokalnej pamięci procesu na serwerze Web lub np. w zewnętrznej bazie danych SQL Server.
Aplikacje rozproszone, przepływy prac	
System.ServiceModel	Budowanie aplikacji rozproszonych za pomocą API WCF
System.Workflow.Runtime System.Workflow.Activities	Budowanie aplikacji rozproszonych za pomocą API WF



Biblioteka klas .NET

GUI

System.Drawing	Umożliwia dostęp do podstawowej funkcjonalności graficznej biblioteki systemowej GDI+.
System.Drawing.Drawing2D	Udostępnia zaawansowaną funkcjonalność związaną z grafiką 2D i grafiką wektorową.
System.Drawing.Text	Udostępnia zaawansowaną funkcjonalność związaną z typografią i wykorzystaniem kolekcji czcionek.
System.Windows.Forms	Funkcje interfejsu użytkownika dla aplikacji działających pod kontrolą systemu Windows.
System.Web.UI	Klasy i interfejsy pozwalające na tworzenie kontroltek i stron składających się na interfejs użytkownika aplikacji ASP .NET.
System.Web.UI.HtmlControls	Klasy umożliwiające tworzenie kontroltek HTML działających po stronie serwera. Kontrolki HTML są mapowane bezpośrednio na tagi HTML interpretowane przez przeglądarki internetowe i pozwalają na programowe manipulowanie elementami HTML na stronie internetowej.
System.Web.UI.WebControls	Klasy umożliwiające umieszczanie na stronie internetowej kontroltek webowych działających po stronie serwera. Zaliczają się do nich elementy formularzy, takie jak przyciski i pola tekstowe, oraz kontrolki takie jak kalendarz czy DataGrid. Kontrolki webowe mogą być manipulowane za pośrednictwem kodu działającego po stronie serwera, a ich model obiektowy niekoniecznie musi przekładać się bezpośrednio na kod HTML.