

Programmazione Orientata agli Oggetti in Linguaggio Java

Introduzione Piattaforme di Riferimento

versione 1.4

Questo lavoro è concesso in uso secondo i termini di una licenza Creative Commons
(vedi ultima pagina)

G. Mecca – Università della Basilicata – mecca@unibas.it



Piattaforme >> Sommario



Sommario

- Linguaggi di Riferimento
- La Piattaforma Java
- La Piattaforma .NET
- Caratteristiche del Linguaggio Java
- Caratteristiche del Linguaggio C#
- Utilizzo degli Strumenti



Linguaggi di Riferimento

- In questo corso
 - ⇒ tecniche e metodi della programmazione a oggetti
- Linguaggio principale
 - ⇒ Java
- Linguaggio secondario
 - ⇒ C# della piattaforma Microsoft.NET



Linguaggi di Riferimento



- Domanda: Perché questi due linguaggi ?
 - ⇒ in effetti ne esistono molti altri
 - ⇒ Smalltalk (tradizionale)
 - ⇒ Delphi
 - ⇒ Python (recente)
- Risposta
 - ⇒ rappresentano le piattaforme di riferimento attuali per lo sviluppo software
 - ⇒ una serie di caratteristiche distintive comuni



La Piattaforma Java

- Complessa piattaforma software
 - ⇒ centrata attorno al linguaggio Java
 - ⇒ ideata dalla Sun (<http://www.java.sun.com>) e tuttora di “proprietà” della Sun
 - ⇒ sviluppata attraverso lo Java Community Process (JCP, <http://www.jcp.org>)
- Essenzialmente
 - ⇒ un complesso di specifiche e di componenti software



La Piattaforma Java

- Un po' di storia
 - ⇒ nasce da un progetto della Sun (progetto Green) con scopi totalmente diversi
 - ⇒ inizialmente si chiamava Oak
 - ⇒ Java 1.0 – rilasciato nel 1995 dalla Sun
 - ⇒ Java 1.1 – rilasciato nel 1997
- Java 2
 - ⇒ dalla v. 1.2 del 1998, riscrittura significativa



La Piattaforma Java

○ Le versioni di Java2

- ⇒ v. 1.2, la prima
- ⇒ v. 1.3, sostanzialmente equivalente alla 1.2
- ⇒ v. 1.4, introduce alcune limitate funzionalità nuove incompatibili con il passato
- ⇒ v. 1.5, ulteriore riscrittura significativa, con molte funzionalità nuove ed incompatibili



La Piattaforma Java

○ Java 2 v. 1.5 = Java 2 5.0

- ⇒ è stato alla fine chiamato Java 2 v. 5.0 per sottolineare le novità rispetto alla versione precedente

○ Il problema di Java 2 5.0

- ⇒ il codice compilato è incompatibile con le vecchie macchine virtuali
- ⇒ le funzionalità vanno utilizzate in modo controllato



La Piattaforma Java

- Attualmente
 - ⇒ la piattaforma è organizzata in tre livelli
- J2SE – Java 2 Standard Edition
 - ⇒ per lo sviluppo di applicazioni desktop
- J2EE – Java 2 Enterprise Edition
 - ⇒ per lo sviluppo di applicazioni lato server
- J2ME – Java 2 Micro Edition
 - ⇒ per lo sviluppo di applicazioni mobili

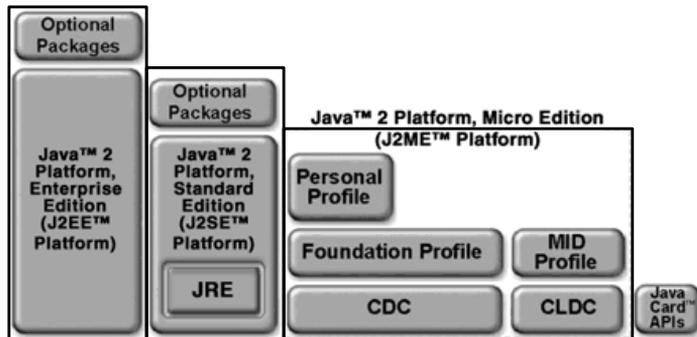


La Piattaforma Java

- Ognuno dei livelli si compone di
 - ⇒ una serie di specifiche (documenti)
 - ⇒ una serie di strumenti (es: compilatore, debugger ecc.)
 - ⇒ una serie di librerie o API (Application Program Interfaces)
 - ⇒ schematizzando (eccessivamente) è possibile dire che i livelli più complessi includono i più semplici



La Piattaforma Java



fonte: www.java.sun.com



La Piattaforma Java

- Principali caratteristiche della piattaforma
 - ⇒ I. scalabile e di ampia applicabilità
 - ⇒ II. fornisce un'ampia libreria di classi
 - ⇒ III. indipendente dalla piattaforma hardware e realmente "portabile"



La Piattaforma Java

○ I. Scalabilità e applicabilità

- ⇒ consente di sviluppare applicazioni di vario genere
- ⇒ sia di piccole e medie dimensioni
- ⇒ sia di grandi dimensioni e altissimo carico
- ⇒ sia in contesti applicativi tradizionali (sistemi informativi) sia in contesti meno tradizionali
- ⇒ con attenzione agli aspetti di sicurezza e prestazioni



La Piattaforma Java

○ Le varie classi di applicazioni

- ⇒ applicazioni “desktop” tradizionali
- ⇒ applicazioni client-server a due livelli
- ⇒ applicazioni Web a tre livelli, anche con componenti distribuiti e ad alto traffico
- ⇒ applicazioni per dispositivi mobili
- ⇒ applicazioni orientate ai servizi (Web Services)



La Piattaforma Java

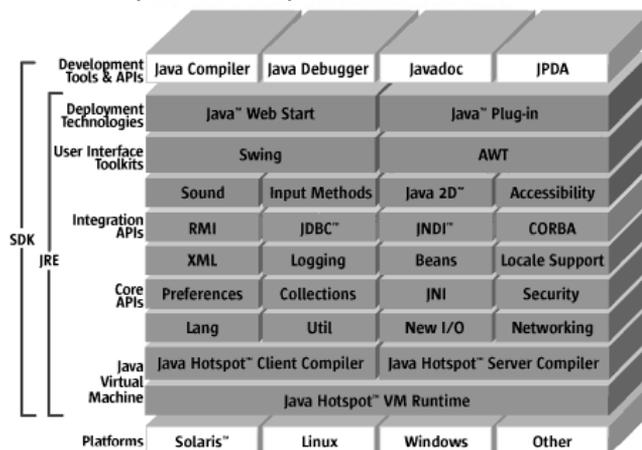
- II. Ampia libreria di classi
 - ⇒ librerie per qualsiasi esigenza di calcolo
 - ⇒ API (“Application Program Interface”): collezione di librerie di classi disponibili per l’uso
 - ⇒ attualmente più di 3000 classi, 32000 metodi e circa 130 package
 - ⇒ collezioni (liste, pile ...), prog. grafica (finestre, bottoni, menu...), prog. Web...



La Piattaforma Java

fonte: www.java.sun.com

Java™ 2 Platform, Standard Edition v 1.4

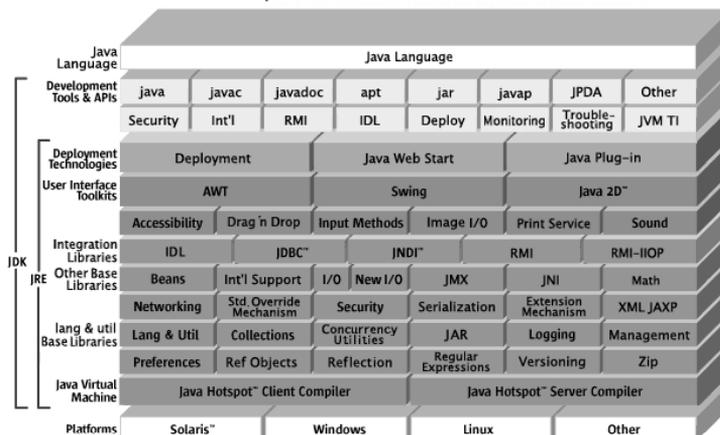




La Piattaforma Java

fonte: www.java.sun.com

Java™ 2 Platform Standard Edition 5.0



La Piattaforma Java

Versione di J2SE	Numero di Classi delle API Standard	Numero di Metodi delle API Standard
1.0	212	2125
1.1	504	5478
1.2	1781	20935
1.3	2130	23901
1.4	2991	32138
1.5 (5.0)	3279	??????



La Piattaforma Java

- III. Indipendenza dalla piattaforma
 - ⇒ caratteristica essenziale di Java
- Le origini
 - ⇒ le “applet” Java
 - ⇒ idea: scaricare codice dalla rete ed eseguirlo attraverso il browser
 - ⇒ requisito essenziale: il codice doveva essere eseguibile indipendentemente dall’hardware e dal sistema operativo



La Piattaforma Java

- Attenzione
 - ⇒ in C++ (e in tutti i linguaggi tradizionali):
 - ⇒ una volta compilato, il codice oggetto e poi l’eseguibile contengono codice macchina specifico del (a) processore (b) sistema operativo utilizzato
 - ⇒ per eseguire il codice su altre piattaforme hardware bisogna ricompilarlo (sperando che tutte le librerie usate e le funzionalità siano compatibili)



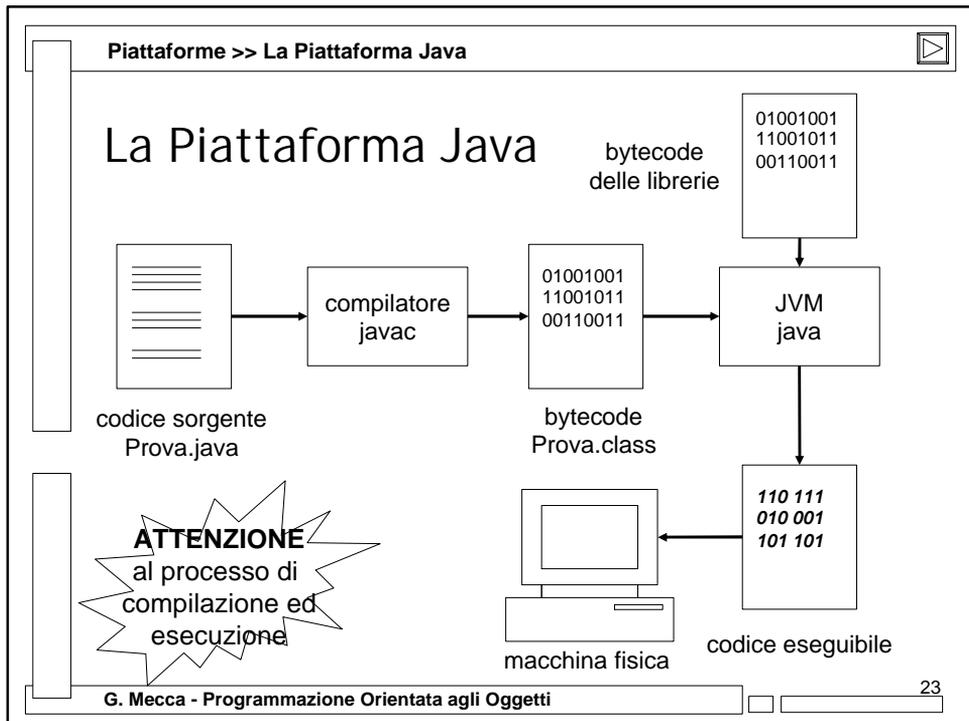
La Piattaforma Java

- Il compilatore di Java
 - ⇒ produce codice oggetto in un linguaggio intermedio per una “macchina virtuale”
- JVM – Java Virtual Machine
 - ⇒ macchina composta da componenti virtuali (processore virtuale, registri virtuali, ecc.)
 - ⇒ il tutto simulato attraverso un’applicazione software (java.exe)



La Piattaforma Java

- Bytecode
 - ⇒ codice oggetto della macchina virtuale (istruzioni eseguibili dalla macchina virtuale)
 - ⇒ linguaggio intermedio
- Funzionamento
 - ⇒ il codice sorgente viene compilato in bytecode
 - ⇒ il bytecode viene eseguito dall’emulatore della m.v. (java.exe) simulando il funzionamento della macchina virtuale sulla macchina fisica



- Piattaforme >> La Piattaforma Java
- ## La Piattaforma Java
- Esecuzione da parte della JVM
 - ⇒ la macchina virtuale potrebbe eseguire lo bytecode interpretandolo
 - Interpretazione del codice
 - ⇒ modo alternativo alla compilazione
 - ⇒ processo per cui il codice di un'applicazione viene tradotto da un interprete in codice eseguibile dal processore
 - ⇒ durante l'esecuzione
- G. Mecca - Programmazione Orientata agli Oggetti 24



La Piattaforma Java

- In effetti, però
 - ⇒ l'interpretazione è a volte inefficiente (la stessa istruzione viene tradotta tutte le volte che viene eseguita)
- Tipicamente
 - ⇒ per ragioni di prestazioni il bytecode viene precedentemente compilato "al volo" dalla macchina virtuale in codice macchina per la macchina fisica



La Piattaforma Java

- JIT Compiler (Just in Time Compiler)
 - ⇒ componente della macchina virtuale che analizza pezzi di bytecode alla prima esecuzione
 - ⇒ e li compila "al volo" in codice oggetto per la macchina fisica
 - ⇒ successivamente viene eseguito direttamente il codice compilato



La Piattaforma Java

- In questo modo
 - ⇒ il codice può essere compilato una volta ed eseguito da molte macchine diverse
 - ⇒ basta che ciascuna delle macchine disponga di una implementazione della JVM e delle librerie necessarie
- JRE – Java Runtime Environment
 - ⇒ JVM e API di J2SE
 - ⇒ “compile once, run anywhere”



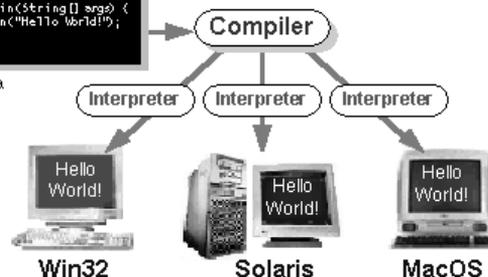
Introduzione

fonte: www.java.sun.com

Java Program

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

HelloWorldApp.java





ATTENZIONE

all'importanza di Java
nell'evoluzione dell'
informatica

La Piattaforma Java

- Prima di Java: le piattaforme principali
 - ⇒ C/C++, Visual Basic, PHP, Delphi
- Differenze sostanziali rispetto a Java
 - ⇒ sono relativamente specializzate (nessuna ha lo stesso ambito di applicabilità di Java)
 - ⇒ le librerie predefinite sono relativamente povere
 - ⇒ il codice compilato non è portabile da una piattaforma all'altra



La Piattaforma .NET

- Piattaforma di riferimento della Microsoft
 - ⇒ per molti versi, una risposta alla larghissima diffusione di Java
- Storia della piattaforma .NET
 - ⇒ nel 2000 la versione 1.0
 - ⇒ nel 2004 la versione 1.1
 - ⇒ attualmente: versione 2.0 beta



La Piattaforma .NET

○ Principali caratteristiche

- ⇒ I. scalabile e di ampia applicabilità
- ⇒ II. fornisce un'ampia libreria di classi
- ⇒ III. multilinguaggio



La Piattaforma .NET

○ I. Scalabilità e applicabilità

- ⇒ esattamente come la piattaforma Java
- ⇒ dalle applicazioni di fascia alta ai palmari
- ⇒ maggiore enfasi su standard recenti
- ⇒ XML e Web services

○ II. Ampia libreria di classi

- ⇒ esattamente come la piattaforma Java

Piattaforme >> La Piattaforma .NET ▶

fonte: Microsoft .NET Academic Resource Kit

Introduzione

<p style="text-align: center;">System.Web</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Services</td></tr> <tr><td>Description</td></tr> <tr><td>Discovery</td></tr> <tr><td>Protocols</td></tr> </table> </td> <td style="width: 50%;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>UI</td></tr> <tr><td>HtmlControls</td></tr> <tr><td>WebControls</td></tr> </table> </td> </tr> <tr> <td> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Caching</td></tr> <tr><td>Configuration</td></tr> </table> </td> <td> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Security</td></tr> <tr><td>SessionState</td></tr> </table> </td> </tr> </table>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Services</td></tr> <tr><td>Description</td></tr> <tr><td>Discovery</td></tr> <tr><td>Protocols</td></tr> </table>	Services	Description	Discovery	Protocols	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>UI</td></tr> <tr><td>HtmlControls</td></tr> <tr><td>WebControls</td></tr> </table>	UI	HtmlControls	WebControls	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Caching</td></tr> <tr><td>Configuration</td></tr> </table>	Caching	Configuration	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Security</td></tr> <tr><td>SessionState</td></tr> </table>	Security	SessionState	<p style="text-align: center;">System.Windows.Forms</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Form</td> <td style="width: 50%;">Button</td> </tr> <tr> <td>MessageBox</td> <td>ListControl</td> </tr> </table> <p style="text-align: center;">System.Drawing</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Drawing2D</td> <td style="width: 50%;">Printing</td> </tr> <tr> <td>Imaging</td> <td>Text</td> </tr> </table> <p style="text-align: center;">System.Xml</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">XSLT</td> <td style="width: 50%;">Serialization</td> </tr> <tr> <td>XPath</td> <td></td> </tr> </table>	Form	Button	MessageBox	ListControl	Drawing2D	Printing	Imaging	Text	XSLT	Serialization	XPath	
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Services</td></tr> <tr><td>Description</td></tr> <tr><td>Discovery</td></tr> <tr><td>Protocols</td></tr> </table>	Services	Description	Discovery	Protocols	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>UI</td></tr> <tr><td>HtmlControls</td></tr> <tr><td>WebControls</td></tr> </table>	UI	HtmlControls	WebControls																				
Services																												
Description																												
Discovery																												
Protocols																												
UI																												
HtmlControls																												
WebControls																												
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Caching</td></tr> <tr><td>Configuration</td></tr> </table>	Caching	Configuration	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Security</td></tr> <tr><td>SessionState</td></tr> </table>	Security	SessionState																							
Caching																												
Configuration																												
Security																												
SessionState																												
Form	Button																											
MessageBox	ListControl																											
Drawing2D	Printing																											
Imaging	Text																											
XSLT	Serialization																											
XPath																												
<p style="text-align: center;">System.Data</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">OLEDB</td> <td style="width: 50%;">SQL</td> </tr> <tr> <td>Design</td> <td>SQLTypes</td> </tr> </table>		OLEDB	SQL	Design	SQLTypes																							
OLEDB	SQL																											
Design	SQLTypes																											
<p style="text-align: center;">System</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Collections</td> <td style="width: 25%;">IO</td> <td style="width: 25%;">Security</td> <td style="width: 25%;">Runtime</td> </tr> <tr> <td>Configuration</td> <td>Net</td> <td>ServiceProcess</td> <td>InteropServices</td> </tr> <tr> <td>Diagnostics</td> <td>Reflection</td> <td>Text</td> <td>Remoting</td> </tr> <tr> <td>Globalization</td> <td>Resources</td> <td>Threading</td> <td>Serialization</td> </tr> </table>		Collections	IO	Security	Runtime	Configuration	Net	ServiceProcess	InteropServices	Diagnostics	Reflection	Text	Remoting	Globalization	Resources	Threading	Serialization											
Collections	IO	Security	Runtime																									
Configuration	Net	ServiceProcess	InteropServices																									
Diagnostics	Reflection	Text	Remoting																									
Globalization	Resources	Threading	Serialization																									

33

G. Mecca - Programmazione Orientata agli Oggetti

Piattaforme >> La Piattaforma .NET ▶

La Piattaforma .NET

- III. Multilinguaggio
 - ⇒ pensata per consentire l'interoperabilità di codice scritto con linguaggi diversi
- I linguaggi di .NET
 - ⇒ ripensamento di linguaggi tradizionali
Microsoft: Visual Basic.NET, C++.NET, J#
 - ⇒ un linguaggio completamente nuovo: C#
 - ⇒ tutti i linguaggi condividono la stessa libreria

34

G. Mecca - Programmazione Orientata agli Oggetti



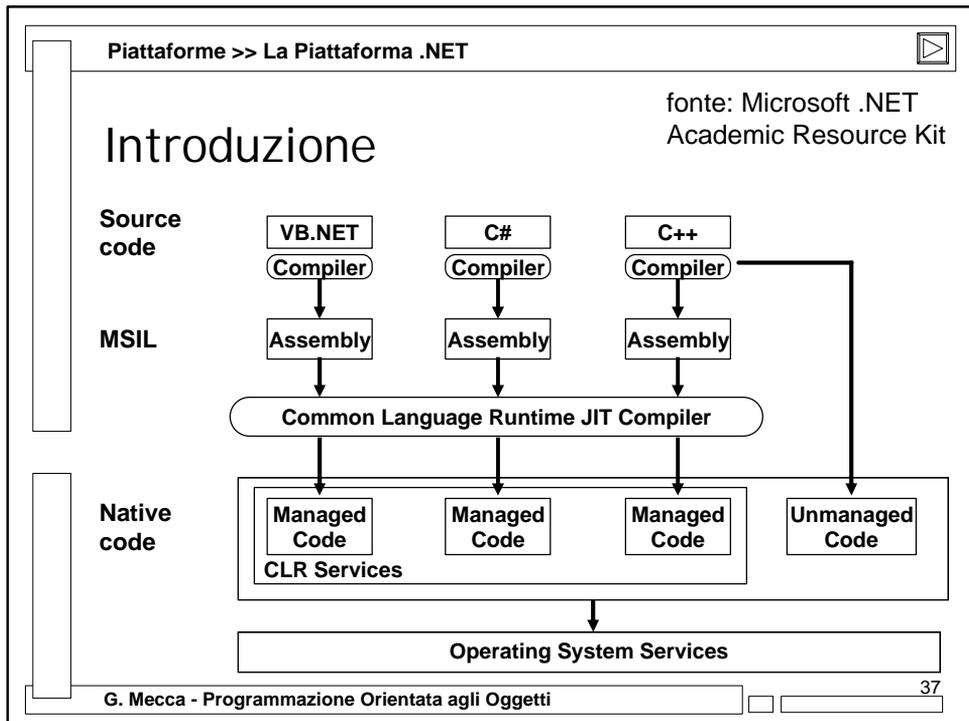
La Piattaforma .NET

- Come consentire l'interoperabilità ?
 - ⇒ idea: architettura software basata su una macchina virtuale
 - ⇒ ed un linguaggio intermedio in cui viene compilato il codice sorgente di ciascun linguaggio
- Nota
 - ⇒ è la stessa architettura di Java ma pensata per uno scopo diverso



La Piattaforma .NET

- CLR – Common Language Runtime
 - ⇒ macchina virtuale .NET
 - ⇒ dotata di un compilatore JIT per Windows
- MSIL – Microsoft Intermediate Language
 - ⇒ linguaggio intermedio di .NET
 - ⇒ linguaggio in cui viene compilato il codice di tutti i linguaggi della famiglia .NET
 - ⇒ sotto forma di assembly (>>)



Piattaforme >> La Piattaforma .NET

La Piattaforma .NET

- **Attenzione**
 - ⇒ la compilazione, come nel caso di Java, genera codice intermedio (IL)
 - ⇒ nel caso di .NET, questo codice viene definito assembly, e può avere varie estensioni, tra cui .exe
 - ⇒ a differenza di Java, per eseguire l'assembly non è necessario invocare esplicitamente la macchina virtuale (il sistema operativo la esegue automaticamente)

G. Mecca - Programmazione Orientata agli Oggetti 38



La Piattaforma .NET

- Per eseguire codice .NET
 - ⇒ una macchina Windows deve avere installato il .NET Framework
- .NET Framework
 - ⇒ CLR
 - ⇒ API di .NET
 - ⇒ analogo di JRE



La Piattaforma .NET

ATTENZIONE
alla differenze tra
le piattaforme

- Di conseguenza
 - ⇒ alcune fortissime analogie con la piattaforma Java
 - ⇒ in particolare l'architettura basata sulla macchina virtuale
- La differenza principale
 - ⇒ un linguaggio su molte piattaforme
 - ⇒ tanti linguaggi su una piattaforma (di fatto)



La Piattaforma .NET

○ Il progetto Mono

- ⇒ tentativo di sviluppare un'implementazione open source della macchina virtuale .NET e del compilatore di C#
- ⇒ sponsorizzato dalla Xymian
- ⇒ <http://www.go-mono.org>
- ⇒ attualmente una implementazione parziale del framework versione 1.1



Caratteristiche del Linguaggio Java

○ Principali caratteristiche del linguaggio

- ⇒ I. completamente orientato agli oggetti
- ⇒ II. sintassi basata sul C/C++
- ⇒ III. semplice e robusto
- ⇒ IV. collegamento dinamico
- ⇒ V. caratteristiche avanzate (multi-thread, riflessione, ...)



Caratteristiche del Linguaggio Java

- I. Completamente orientato agli oggetti
 - ⇒ non è consentita programmazione ibrida
 - ⇒ tutta la programmazione è basata su classi e oggetti
- II. Sintassi basata sul C/C++
 - ⇒ similarità nei costrutti del linguaggio
 - ⇒ es: espressioni, if, cicli ecc.



Caratteristiche del Linguaggio Java

- III. Semplice e robusto
 - ⇒ drastica semplificazione rispetto al C++
 - ⇒ gestione automatica della memoria (“garbage collection”)
 - ⇒ fortemente tipato
 - ⇒ gestione rafforzata delle eccezioni
 - ⇒ ereditarietà singola (>>)



Caratteristiche del Linguaggio Java

- IV. Collegamento dinamico
 - ⇒ le librerie non devono essere collegate staticamente al codice
 - ⇒ il collegamento avviene dinamicamente (appena viene richiesto l'uso della libreria)
 - ⇒ grosso vantaggio: consente di collegare versioni aggiornate delle librerie senza dover ricollegare il codice
- V. Altre caratteristiche avanzate



Caratteristiche del Linguaggio C#

- Principali caratteristiche di Java e C#
 - ⇒ I. completamente orientato agli oggetti
 - ⇒ II. sintassi basata sul C/C++
 - ⇒ III. semplice e robusto (ATTENZIONE)
 - ⇒ IV. collegamento dinamico
 - ⇒ V. caratteristiche avanzate (multi-thread, riflessione, ...)



Caratteristiche del Linguaggio C#

- Semplice e robusto
 - ⇒ C# eredita molte delle scelte fatte per Java
 - ⇒ gestione automatica della memoria, tipizzazione, ereditarietà singola, eccezioni...
- Ma
 - ⇒ è meno restrittivo nell'imporre queste scelte al programmatore
 - ⇒ es: gestione della memoria, eccezioni
 - ⇒ questa è una delle principali differenze



Utilizzo degli Strumenti

- Per sviluppare in Java
 - ⇒ è necessario installare il J2SE Development Kit (JDK)
 - ⇒ scaricabile liberamente da <http://www.java.sun.com> con la relativa documentazione (zip con file html)
 - ⇒ attenzione alle dimensioni
 - ⇒ jdk-1_5_0-windows.exe >> circa 52 Mega
 - ⇒ jdk-1_5_0-doc.zip >> circa 44 Mega



Utilizzo degli Strumenti

○ Post-Configurazione

- ⇒ creare una nuova variabile di ambiente `JAVA_HOME` e assegnargli il valore della cartella in cui è installato l'SDK
- ⇒ Pannello di Controllo >> Sistema >> Avanzate >> Variabili di Ambiente
- ⇒ es: `JAVA_HOME=C:\Programmi\Java\jdk1.5.0_01`
- ⇒ aggiornare la variabile `PATH` aggiungendo il valore `%JAVA_HOME%\bin`



Utilizzo degli Strumenti

○ Gli strumenti principali

- ⇒ il compilatore: `javac.exe`
- ⇒ la macchina virtuale: `java.exe`

○ Il passo

- ⇒ compilare il sorgente (`.java`) con `javac`
- ⇒ eseguire il bytecode (`.class`) con `java`



Un Semplice Programma Java

```
/*
   Questo e' il primo programma Java
   file: Prima.java
*/
package prima;

public class Prima {

    public static void main(String[] args) {
        // stampa una scritta sullo schermo
        System.out.println("Prima prova");
    }
}
```

>>



```
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\mecca>E:
E:\>cd temp
E:\temp>dir prima
Il volume nell'unit  E   APPOGGIO
Numero di serie del volume: 20C6-D0FB

Directory di E:\temp\prima
18/03/2004 12.26 <DIR>          .
18/03/2004 12.26 <DIR>          ..
18/03/2004 08.54                239 Prima.java
                1 File          239 byte
                2 Directory  1.706.311.680 byte disponibili

E:\temp>javac prima\Prima.java

E:\temp>dir prima
Il volume nell'unit  E   APPOGGIO
Numero di serie del volume: 20C6-D0FB

Directory di E:\temp\prima
18/03/2004 12.26 <DIR>          .
18/03/2004 12.26 <DIR>          ..
18/03/2004 12.26                421 Prima.class
18/03/2004 08.54                239 Prima.java
                2 File          660 byte
                2 Directory  1.706.311.680 byte disponibili

E:\temp>java prima.Prima
Prima prova

E:\temp>_
```



Utilizzo degli Strumenti

○ Per sviluppare in C#

- ⇒ è necessario installare il
.NET Redistributable Package e il
.NET Framework SDK v. 1.1
- ⇒ scaricabili liberamente dal sito
<http://www.gotdotnet.com>
- ⇒ attenzione alle dimensioni
- ⇒ dotnetfx.exe >> ca. 23 Mega
- ⇒ NETSDKSetup1.1.exe >> circa 112 Mega



Utilizzo degli Strumenti

○ Post-Configurazione

- ⇒ cambiare il PATH aggiungendo il valore
C:\Windows\Microsoft.NET\Framework\v1.1.4322\
⇒ ATTENZIONE: il nome effettivo della cartella
può cambiare con la versione dell'SDK

○ Lo strumento principale

- ⇒ il compilatore csc.exe
- ⇒ la macchina virtuale viene eseguita automaticamente



Un Semplice Programma C#

```
/*
   Questo e' il primo programma in C#
   file: primo.cs
*/
namespace primo {

public class Prima {

    public static void Main(string[] args) {
        // stampa una scritta sullo schermo
        System.Console.WriteLine("Prima Prova");
    }
}
}
```

>>



```
8 Prompt dei comandi
E:\appoggio\csharp>dir
Il volume nell'unit  E   APPOGGIO
Numero di serie del volume: 20C6-D0FB

Directory di E:\appoggio\csharp

26/10/2003 16.15 <DIR>          .
26/10/2003 16.15 <DIR>          ..
26/10/2003 16.13                253 primo.cs
                1 File          253 byte
                2 Directory  4.212.260.864 byte disponibili

E:\appoggio\csharp>csc primo.cs
Compilatore Microsoft (R) Visual C# .NET versione 7.10.3052.4
per Microsoft (R) .NET Framework 1.1.4322
Copyright (C) Microsoft Corporation 2002. Tutti i diritti riservati.

E:\appoggio\csharp>dir
Il volume nell'unit  E   APPOGGIO
Numero di serie del volume: 20C6-D0FB

Directory di E:\appoggio\csharp

26/10/2003 16.17 <DIR>          .
26/10/2003 16.17 <DIR>          ..
26/10/2003 16.13                253 primo.cs
26/10/2003 16.17                3.072 primo.exe
                2 File          3.325 byte
                2 Directory  4.212.256.768 byte disponibili

E:\appoggio\csharp>primo.exe
Prima Prova

E:\appoggio\csharp>
```



Riassumendo

- Linguaggi di Riferimento
- La Piattaforma Java
- La Piattaforma .NET
- Caratteristiche del Linguaggio Java
- Caratteristiche del Linguaggio C#
- Utilizzo degli Strumenti



Termini della Licenza

- This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
- Questo lavoro viene concesso in uso secondo i termini della licenza "Attribution-ShareAlike" di Creative Commons. Per ottenere una copia della licenza, è possibile visitare <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviare una lettera all'indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.