

Programmazione Orientata agli Oggetti in Linguaggio Java

Test e Correzione:

C#

Parte b

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



Test e Correzione: C# >> Sommario



Sommario

- Riflessione
- Il File AssemblyInfo
- Global Assembly Cache



Riflessione

- Anche .NET
 - ⇒ è una famiglia di linguaggi basati sulla riflessione
- La classe analoga di `java.lang.Class`
 - ⇒ `System.Type`
 - ⇒ un pò più generale, dal momento che .NET utilizza un sistema unificato di tipi
 - ⇒ che racchiude tutti i tipi del linguaggio



Riflessione

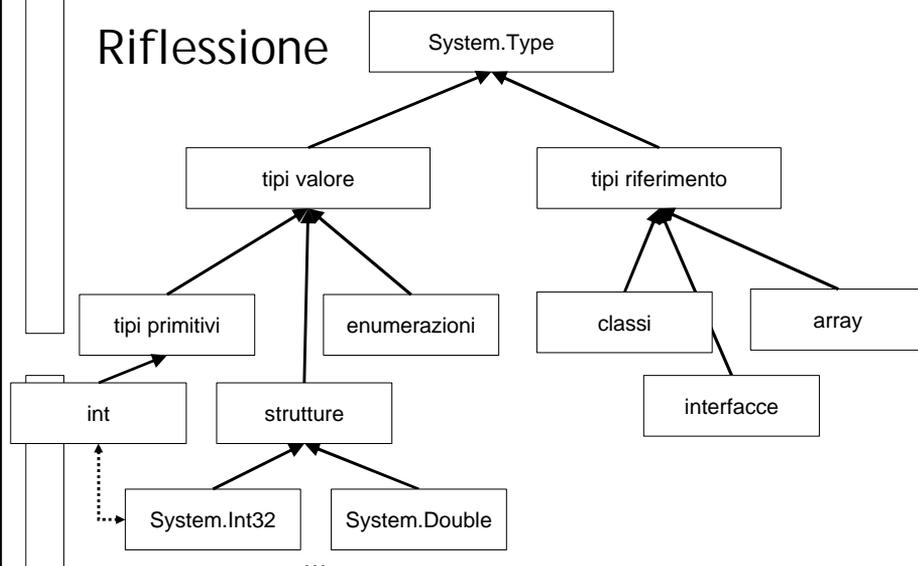
- La distinzione di C#
 - ⇒ tipi riferimento e tipi valore
- Tipi riferimento
 - ⇒ tipi i cui valori vengono allocati nello heap (es: oggetti delle classi)
 - ⇒ vengono manipolati attraverso variabili riferimento
 - ⇒ il cui valore è l'OID del dato (es: oggetto)

Riflessione

o Tipi valore

- ⇒ tipi i cui valori vengono allocati nello stack (es: dati primitivi)
- ⇒ non vengono manipolati attraverso i riferimenti
- ⇒ il valore della variabile coincide con il valore del dato (es: int)
- ⇒ sono tipi valore i tipi di base, le strutture e le enumerazioni

Riflessione





Riflessione

- Come ottenere il riferimento al tipo
 - ⇒ vari modi
 - ⇒ il metodo `System.Type GetType()` di `System.Object` restituisce il riferimento all'oggetto `Type` corrispondente all'oggetto
es: `System.Type classe = oggetto.GetType();`
 - ⇒ la funzione `typeof()` applicata ad un tipo
es: `System.Type tipo = typeof(double);`



Riflessione

- `Unibas.Utilita.Riflessione.Ispezionatore`
 - ⇒ un ispezionatore basato sulla riflessione per C#
- Il namespace `System.Reflection`
 - ⇒ contiene tutte le classi ed i metodi per la riflessione
 - ⇒ al solito, le operazioni sono orientate agli assembly e non alle classi singole



Riflessione

- Le classi fondamentali

- ⇒ System.Reflection.MethodInfo
- ⇒ System.Reflection.ConstructorInfo
- ⇒ System.Reflection.FieldInfo
- ⇒ System.Reflection.PropertyInfo

- Una particolarità

- ⇒ i metodi consentono l'accesso esclusivamente ai membri pubblici



Riflessione

>> Ispezionatore.cs
>> Ispezionatore.exe

- Il funzionamento di Ispezionatore

- ⇒ deve essere compilato in un assembly eseguibile
- ⇒ deve essere eseguito specificando il nome di un assembly da ispezionare
- ⇒ analizza tutti i tipi contenuti nell'assembly e riporta le informazioni relative

- Esempio

- ⇒ Ispezionatore.exe e:\codice\lib\Unibas.Utilita.dll



Riflessione

○ NUnit

- ⇒ come JUnit, anche NUnit è pesantemente basato sulla riflessione
- ⇒ in particolare, analizza l'assembly dei test per estrarre gli attributi
- ⇒ utilizzando i metodi della classe System.Attribute
- ⇒ in particolare GetCustomAttributes()



Il File AssemblyInfo

>> ildasm.exe Unibas.Utilita.dll

○ Gli assembly di .NET

- ⇒ archivi di classi compilate
- ⇒ contengono codice oggetto

○ Analogo

- ⇒ simili ai file .jar di Java

○ Infatti

- ⇒ anche gli assembly di .NET hanno un proprio manifesto, visibile con ildasm.exe



Il File AssemblyInfo

>> utilita\AssemblyInfo.cs

- Metainformazioni per l'assembly
 - ⇒ possono essere specificate attraverso attributi di .NET
- Una tecnica standard
 - ⇒ associare ad un assembly un file chiamato AssemblyInfo.cs
 - ⇒ contenente solo metainformazioni sotto forma di attributi
 - ⇒ es: numero di versione



Il File AssemblyInfo

- Distribuzione e installazione dell'assembly
 - ⇒ il framework consulta le metainformazioni aggiunte attraverso gli attributi
 - ⇒ utilizzando la riflessione
 - ⇒ es: firma digitale
- Un assembly di tipo netmodule
 - ⇒ un assembly senza metainformazioni
 - ⇒ viene creato per essere aggiunto ad altri assembly con l'opzione /target:module



Global Assembly Cache

- Una caratteristica di C# diversa da Java
 - ⇒ il modo in cui viene gestita la visibilità
- In particolare
 - ⇒ in .NET non esiste un concetto analogo al concetto di classpath in Java
- Di conseguenza
 - ⇒ le regole di visibilità sono completamente diverse



Global Assembly Cache

- Per il compilatore
 - ⇒ i riferimenti vengono risolti attraverso l'opzione /ref
- Per la macchina virtuale
 - ⇒ ci sono due diverse opzioni
 - ⇒ prima opzione: assembly contenuti nella stessa cartella
 - ⇒ seconda opzione: "global assembly cache"



Global Assembly Cache

○ Prima opzione

- ⇒ tutti gli assembly dell'applicazione sono contenuti nella stessa cartella
- ⇒ in questo caso si parla di "assembly privati" – non condivisibili con altre applicazioni
- ⇒ è l'opzione fortemente consigliata dalla Microsoft
- ⇒ ma in alcuni casi è necessario condividere lo stesso assembly tra due applicazioni



Global Assembly Cache

○ Seconda opzione

- ⇒ installazione dell'assembly nella GAC

○ Global Assembly Cache

- ⇒ collezione di assembly .NET condivisibili da applicazioni diverse
- ⇒ corrisponde al contenuto della cartella c:\windows\assembly

○ ATENZIONE: è una cartella di sistema

- ⇒ non può essere trattata come le altre



Global Assembly Cache

- Come visualizzarne il contenuto
 - ⇒ l'utility gacutil.exe (da console)
 - ⇒ lo strumento "Configurazione .NET Framework" in Strumenti di Amministrazione
- Un esempio di assembly condiviso
 - ⇒ nunit.framework.dll
 - ⇒ viene installato da NUnit nella GAC
 - ⇒ è possibile che ce ne siano diverse versioni



Global Assembly Cache

>> utilita\AssemblyInfo.cs

- Caratteristiche della GAC
 - ⇒ gestione delle versioni: sono consentite versioni diverse dello stesso assembly
 - ⇒ sicurezza: sono gestiti solo gli assembly "firmati" (procedura complessa)
 - ⇒ la firma viene specificata in AssemblyInfo.cs
- In generale
 - ⇒ si tratta di uno strumento abbastanza difficile da usare



Global Assembly Cache

- Risoluzione dei riferimenti
 - ⇒ durante la compilazione, tutti gli assembly a cui viene fatto riferimento vengono registrati nel manifesto dell'assembly creato (nome e versione) come "extern"
 - ⇒ durante l'esecuzione, gli assembly esterni vengono cercati per cominciare nella GAC
 - ⇒ se non vengono trovati, vengono cercati nella stessa cartella dell'assembly



Riassumendo

- Riflessione
- Il File AssemblyInfo
- Global Assembly Cache



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.