

# Programmazione Orientata agli Oggetti in Linguaggio Java

## Strumenti di Sviluppo: IDE

versione 1.0

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



### Strumenti di Sviluppo: IDE >> Sommario

## Sommario

- Introduzione
- Tipiche Funzionalità di un IDE
- Un Ulteriore Esempio
- PlugIn



## Introduzione

### ○ IDE

- ⇒ Integrated Development Environment, ovvero Ambiente Integrato di Sviluppo
- ⇒ un'applicazione che assiste nello sviluppo di progetti software

### ○ Tipicamente

- ⇒ un'applicazione molto "pesante"
- ⇒ con requisiti molto onerosi in termini di risorse hardware

>> Task Manager - NetBeans



## Introduzione

### ○ Alcune conseguenze

- ⇒ l'IDE non è adatto allo sviluppo su macchine datate o di scarse prestazioni
- ⇒ l'IDE non è uno strumento adatto a sviluppare piccole applicazioni o singole classi

### ○ In questi casi

- ⇒ l'editor di testi e il compilatore da linea di comando restano lo strumento più adatto



## Introduzione

- Gli IDE open source più diffusi per Java
  - ⇒ NetBeans
  - ⇒ Eclipse
- Gli IDE commerciali
  - ⇒ IntelliJ IDEA
  - ⇒ Borland JBuilder
  - ⇒ Oracle JDeveloper



## Introduzione

- NetBeans
  - ⇒ progetto open source supportato dalla Sun
  - ⇒ disponibile su <http://www.netbeans.org> sotto forma di installabile (netbeans-xxx.exe)
  - ⇒ l'IDE commerciale della Sun – Sun One Studio è basato sul codice di NetBeans
  - ⇒ viene considerato dalla Sun l'IDE standard per Java; è distribuito in forma integrata con l'SDK su <http://www.java.sun.com>



## Introduzione

### ○ Eclipse

- ⇒ progetto open source supportato dalla IBM
- ⇒ disponibile su <http://www.eclipse.org> sotto forma di file zip da scompattare sul disco
- ⇒ è una piattaforma di sviluppo “neutra”, non è pensato specificamente per Java
- ⇒ a differenza di NetBeans non è particolarmente “ortodosso” rispetto alla piattaforma



## Tipiche Funzionalità di un IDE

### ○ La modalità di lavoro

- ⇒ centrata attorno alla nozione di progetto

### ○ Progetto dell'IDE

- ⇒ collezione di file che compongono un'applicazione
- ⇒ tipicamente corrisponde al contenuto di una cartella del disco
- ⇒ e produce come risultato un .jar



## Tipiche Funzionalità di un IDE

- Definizione di un nuovo progetto
  - ⇒ tipicamente l'IDE consente di scegliere tra vari modelli di progetto predefiniti
  - ⇒ è necessario specificare il nome
  - ⇒ è necessario specificare la cartella del disco che conterrà i file del progetto
  - ⇒ l'IDE crea uno scheletro di cartelle e file secondo il modello scelto



## Tipiche Funzionalità di un IDE

**ATTENZIONE**

alla definizione del CLASSPATH

- Definizione dei riferimenti
  - ⇒ collezione di cartelle e jar
  - ⇒ definiscono il classpath valido per il progetto
- Di conseguenza
  - ⇒ il CLASSPATH di sistema non viene tipicamente utilizzato dall'IDE
  - ⇒ ciascun progetto ha il proprio classpath
  - ⇒ può averne più di uno (es: compilazione e test)



## Tipiche Funzionalità di un IDE

### ○ Progetti e Sottoprogetti

- ⇒ un progetto può avere dei sottoprogetti
- ⇒ progetti di cui è composto o da cui dipende
- ⇒ nella configurazione del progetto è possibile specificare i sottoprogetti da cui dipende
- ⇒ es: applicazione eseguibile e librerie

### ○ In questo caso

- ⇒ i jar dei sottoprogetti vengono aggiunti al classpath
- ⇒ i sottoprogetti vengono costruiti ogni volta che viene costruito il progetto principale



## Tipiche Funzionalità di un IDE

### ○ Contenuto della cartella di progetto

- ⇒ file di codice sorgente
- ⇒ altri file che costituiscono l'applicazione (es: file xml, immagini, documentazione ecc.)
- ⇒ file di codice oggetto
- ⇒ file contenenti le "metainformazioni"

### ○ File di metainformazione

- ⇒ creati dall'IDE, contengono la configurazione del progetto
  - >> creazione progetto NetBeans + classpath
  - >> cartella di progetto



## Tipiche Funzionalità di un IDE

- Una volta creato il progetto
  - ⇒ è possibile procedere alla scrittura del codice
  - ⇒ ed alle operazioni di costruzione
- In sintesi
  - ⇒ l'IDE integra vari strumenti (editor, compilatore, macchina virtuale, debugger...)
  - ⇒ e consente di lavorare attraverso un ambiente a finestre multiple in cui vengono utilizzati i diversi strumenti



## Tipiche Funzionalità di un IDE

- Le principali finestre dell'IDE
  - ⇒ finestra dell'editor per la scrittura del codice
  - ⇒ finestra del progetto, in cui viene visualizzata la struttura del progetto
  - ⇒ finestra di esecuzione, in cui vengono visualizzati i risultati dell'esecuzione
  - ⇒ finestra del debugger, in cui vengono visualizzati i controlli del debugger
  - ⇒ finestra del filesystem, con cartelle e file



## Tipiche Funzionalità di un IDE

- Organizzazione delle finestre
  - ⇒ forniscono diverse “prospettiva sul progetto”, ovvero ad un punto di vista sul contenuto
- La finestra di progetto
  - ⇒ tipicamente fornisce una prospettiva logica (package, classi, metodi)
- La finestra del filesystem
  - ⇒ tipicamente fornisce una prospettiva fisica (cartelle e file)

>> NetBeans



## Tipiche Funzionalità di un IDE

- Aggiunta di file di codice sorgente
  - ⇒ l'IDE consente di aggiungere cartelle, package e file al progetto
  - ⇒ anche in questo caso l'IDE fornisce una serie di modelli predefiniti
  - ⇒ es: classe, classe con main, interfaccia ecc.
  - ⇒ viene creato uno scheletro che il programmatore deve riempire





## Tipiche Funzionalità di un IDE

- Scrittura del codice sorgente
  - ⇒ funzionalità evolute di supporto al programmatore
- Le principali funzionalità
  - ⇒ completamento automatico del codice
  - ⇒ compilazione interattiva
  - ⇒ gestione delle versioni
  - ⇒ refactoring (>>)



## Tipiche Funzionalità di un IDE

- Completamento automatico del codice
  - ⇒ durante la scrittura l'IDE suggerisce i possibili completamenti per le istruzioni
  - ⇒ tipicamente dopo aver scritto ogni .
- Compilazione interattiva
  - ⇒ l'IDE esegue una parziale compilazione del codice scritto "in linea"
  - ⇒ gli errori sintattici riscontrati vengono segnalati durante la scrittura



## Tipiche Funzionalità di un IDE

- Gestione delle versioni
  - ⇒ nei grandi progetti, il codice viene mantenuto in archivi (“repository”) centralizzati
- Version Control System
  - ⇒ archivio in cui vengono mantenute tutte le versioni dei file e delle cartelle
  - ⇒ l’IDE consente normalmente di collegarsi ad un VCS per scaricare l’ultima versione dei file del progetto (o una precedente)



## Tipiche Funzionalità di un IDE

- Inoltre, varie altre facilitazioni
- Una funzionalità molto utile
  - ⇒ visualizzazione delle “cose da fare” (“to do”)
  - ⇒ pressochè tutti gli IDE consentono di aggiungere al codice commenti con sintassi particolare (es: //TODO:....) in cui vengono descritte le operazioni in sospeso
  - ⇒ e poi raccoglierli tutti in una lista unica da utilizzare come promemoria

>> NetBeans



## Tipiche Funzionalità di un IDE

- Accesso facilitato alle risorse
  - ⇒ il programmatore ha facilmente accesso a tutti i componenti del progetto e può rapidamente consultarli
  - ⇒ il programmatore ha accesso alla documentazione delle librerie di sistema
- Di conseguenza
  - ⇒ la produttività è decisamente aumentata



## Tipiche Funzionalità di un IDE

- Operazioni tipiche con l'IDE
  - ⇒ Build (Preparazione e Compile)
  - ⇒ Clean (elimina il codice compilato)
  - ⇒ Rebuild (Clean + Build)
  - ⇒ Run
  - ⇒ Debug (esecuzione del debugger integrato)
  - ⇒ JavaDoc
  - ⇒ Test



## Tipiche Funzionalità di un IDE

- Scrittura ed esecuzione dei test
  - ⇒ gli IDE per Java sono tipicamente integrati con JUnit
  - ⇒ consentono di sviluppare classi di test (in alcuni casi in modo semiautomatico)
  - ⇒ e di eseguire i test sviluppati periodicamente sul codice



## Un Ulteriore Esempio

- Nel seguito
  - ⇒ vediamo come realizzare le stesse operazioni viste con NetBeans utilizzando Eclipse

>> creazione di un progetto con Eclipse



## Plugin

- Una caratteristica comune ai due IDE
  - ⇒ si tratta di strumenti estensibili
  - ⇒ attraverso l'installazione di "plugin"
- Plugin
  - ⇒ modulo che viene aggiunto all'IDE per implementare una funzionalità aggiuntiva
  - ⇒ i siti degli IDE contengono un elenco dei plugin disponibili



## Plugin

- Esempi di plugin
  - ⇒ editor per UML
  - ⇒ strumenti per la formattazione del codice
  - ⇒ framework di test alternativi a JUnit
  - ⇒ strumenti per la gestione di file XML
  - ⇒ client per la gestione di sistemi di basi di dati
- In realtà
  - ⇒ molti dei moduli predefiniti dell'IDE sono essi stessi plugin distribuiti a corredo dell'IDE

## PlugIn

### ○ Plugin per NetBeans

- ⇒ distribuiti sotto forma di archivio di classi con estensione .nbm (“NetBeans module”)
- ⇒ si installa da Tools >> Update Center selezionando l’opzione “manuale”
- ⇒ per visualizzare e configurare i moduli installati: Tools >> Options >> IDE Configuration >> System >> Modules

## PlugIn

### ○ Plugin per Eclipse

- ⇒ distribuito sotto forma di archivio compresso (.zip) di file jar
- ⇒ è necessario decomprimere il file .zip del plugin nella cartella %ECLIPSE\_HOME%
- ⇒ vengono aggiunti file alle cartelle plugins e features
- ⇒ per visualizzare i plugin installati: Help >> About Eclipse Platform >> Plugin Detail



## PlugIn

- Update Center

- ⇒ ciascun IDE consente di collegarsi periodicamente alla rete per mantenere aggiornato l'IDE ed i relativi plugin

- Attenzione

- ⇒ i plugin sviluppati da terze parti possono rendere instabili l'IDE a causa di una scarsa qualità del codice o di poca aderenza alle regole di sviluppo fissate per i plugin



## PlugIn

- In altri termini

- ⇒ gli IDE moderni hanno due aspetti completamente diversi

- Aspetto tradizionale

- ⇒ ambiente interattivo di sviluppo

- Aspetto innovativo

- ⇒ framework (API, librerie, regole) per lo sviluppo di moduli con cui estendere l'ambiente interattivo



## Riassumendo

- Introduzione
- Tipiche Funzionalità di un IDE
- Un Ulteriore Esempio
- Plugin



## 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.