

# Programmazione Orientata agli Oggetti in Linguaggio Java

## Strumenti di Sviluppo: Conclusioni

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: Conclusioni >> Sommario



## Sommario

- Ricapitolazione
- Altri Strumenti
- API di Java
  - ⇒ tools.jar



## Ricapitolazione

- Ambiente di sviluppo
  - ⇒ IDE
- Gli IDE di riferimento per Java
  - ⇒ NetBeans
  - ⇒ Eclipse
- Il sistema di costruzione di riferimento
  - ⇒ Ant



## Ricapitolazione

- Strumenti raggruppati
  - ⇒ editor con funzioni avanzate
  - ⇒ strumento per il refactoring
  - ⇒ compilatore
  - ⇒ debugger
  - ⇒ framework per l'esecuzione dei test
  - ⇒ strumenti per la documentazione
  - ⇒ strumenti per l'impacchettamento e la distribuzione (zip, jar ecc.)



## Altri Strumenti

- In realtà
  - ⇒ esistono altri strumenti utili di sviluppo
  - ⇒ due categorie
- Strumenti di analisi statica del codice
  - ⇒ metriche, verifica di stile
- Strumenti di analisi dinamica del codice
  - ⇒ profiler
  - ⇒ strumenti per l'analisi di copertura



## Altri Strumenti

- Strumenti di analisi statica
  - ⇒ metriche (es: Breakdown – <http://sourceforge.net/projects/ibatisbreakdown>)
  - ⇒ aderenza a convenzioni di stile (es: Checkstyle, <http://checkstyle.sourceforge.net>)
  - ⇒ aspetti “strutturali” dell'organizzazione dei componenti (es: IBM Structural Analysis for Java – <http://www.ibm.com>)



## Altri Strumenti

- Nota
  - ⇒ molte di queste funzionalità sono fornite anche da RefactorIt
- “Audit”
  - ⇒ comando che analizza il codice per riscontrare errori di programmazione
- “Code metrics”
  - ⇒ calcola un certo numero di metriche sul codice



## Altri Strumenti

- Profilatore (“profiler”)
  - ⇒ strumento attraverso cui è possibile raccogliere informazioni sul funzionamento di un’applicazione durante l’esecuzione
  - ⇒ per poi analizzarne le prestazioni ed eventualmente ottimizzarle
  - ⇒ tipicamente si basa su un’API della piattaforma che consente di raccogliere informazioni durante l’esecuzione del codice



## Altri Strumenti

- Java Virtual Machine Tools Interface
  - ⇒ interfaccia della macchina virtuale Java che consente a strumenti esterni scritti in Java (come il profiler) di ottenere informazioni sull'esecuzione di un'applicazione
- Tipologie di dati raccolti dal profiler
  - ⇒ dati sull'utilizzo della CPU e dello stack
  - ⇒ dati sull'utilizzo dello heap



## Altri Strumenti

- CPU e Stack
  - ⇒ viene analizzato continuamente o periodicamente ("campionato") lo stack
  - ⇒ viene contato il numero di esecuzioni di un metodo e i tempi di esecuzione
- Heap
  - ⇒ viene analizzato lo heap per verificare quanti oggetti sono stati creati e da quali metodi
  - ⇒ e l'attività del garbage collector



## Altri Strumenti

- Il profiler standard di Java
  - ⇒ hprof.exe, distribuito con l'SDK
  - ⇒ si tratta di un profiler molto rudimentale
- Per eseguirlo
  - ⇒ `java -Xrunhprof:<opzione>,<opzione>,... <classe>`
- Esempi
  - ⇒ `java -Xrunhprof:cpu=samples,interval=20,depth=3 <classe>`
  - ⇒ `java -Xrunhprof:heap=sites <classe>`



## Altri Strumenti

- Risultati del profiling
  - ⇒ i dati sono prodotti in un file di testo (nel caso di hprof hprof.txt)
  - ⇒ e poi devono essere analizzati
  - ⇒ esistono strumenti grafici per l'analisi dei dati prodotti dal profiler
  - ⇒ esempio: HPJmeter



## Altri Strumenti

- NetBeans Profiler
  - ⇒ il profiler della nuova generazione
  - ⇒ profiler integrato con l'IDE e completamente grafico
  - ⇒ è necessario installarlo come plugin
  - ⇒ fornisce dati su tempi della CPU, utilizzo dello heap e molto altro ancora



## Altri Strumenti

- Strumenti di analisi della copertura
  - ⇒ analizzano l'esecuzione dell'applicazione per verificare quante istruzioni sono effettivamente eseguite e quante no
  - ⇒ utili a capire quali sono le sezioni "morte" del codice o a capire quanta parte del codice è coperta effettivamente dai test
- Uno strumento di copertura open-source
  - ⇒ Emma – <http://emma.sourceforge.net>



## API di Java

- I task di Ant
  - ⇒ sono implementati attraverso classi Java
- I task fondamentali
  - ⇒ corrispondono a classi distribuite con la piattaforma
- Caratteristica di Java
  - ⇒ tutti gli strumenti distribuiti con la piattaforma sono in realtà realizzati in Java



## API di Java

>> tools.jar

- tools.jar
  - ⇒ disponibile in %JAVA\_HOME%\lib
  - ⇒ fornisce tutti i package corrispondenti ai principali strumenti
  - ⇒ deve essere aggiunto al classpath
- In particolare
  - ⇒ è abbastanza semplice, dall'interno di una classe Java, compilarne un'altra o generarne la documentazione





## Riassumendo

- Ricapitolazione
- Altri Strumenti
- API di Java
  - ⇒ tools.jar



## 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.