

# Programmazione Orientata agli Oggetti in Linguaggio Java

## Sintassi e Semantica: Conclusioni

versione 2.2

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



Sintassi e Semantica: Conclusioni >> Sommario



## Sommario

- Riepilogo della Metafora
- La Piattaforma Java
  - ⇒ Boxing e Unboxing
  - ⇒ java.lang
  - ⇒ javadoc
- Errori Frequenti



## Riassunto della Metafora

### ○ Applicazione a oggetti

- ⇒ collezione di componenti >> piccoli robot antropomorfi
- ⇒ classe >> componente femmina
- ⇒ oggetto >> componente maschio
- ⇒ heap >> città dei robot (zona in cui vivono i componenti)
- ⇒ package >> condominio di componenti (determina indirizzo e visibilità)



## Riassunto della Metafora

### ○ Compiti dei Componenti

- ⇒ proprietà >> conoscere informazione
- ⇒ eseguire metodi >> fornire servizi eseguendo compiti

### ○ Comunicazione tra i componenti

- ⇒ chiamate di metodi o richieste di utilizzo delle proprietà >> richieste di prestare servizi



## Riassunto della Metafora

### ○ Utilizzo dei componenti

- ⇒ esecuzione dei costruttori >> la classe “fabbrica” gli oggetti che ne sono istanza
- ⇒ descrizione della struttura degli oggetti nel codice (proprietà e metodi) >> istruzioni di montaggio
- ⇒ riferimento >> telecomando per “pilotare” un oggetto (differenza rispetto alle classi)



## Riassunto della Metafora

### ○ Utilizzo dei componenti

- ⇒ per chiedere servizi ad una classe >> è sufficiente “urlarne” il nome
- ⇒ per chiedere servizi ad un oggetto >> è necessario farlo attraverso un telecomando
- ⇒ oggetto inutilizzabile >> oggetto per il quale non ci sono più telecomandi disponibili
- ⇒ garbage collector >> netturbino degli oggetti inutilizzabili



## Riassunto della Metafora

### ○ Ruoli e responsabilità

- ⇒ le classi non possono fare tutto >> fabbricare oggetti “stanca”
- ⇒ classi fabbricatrici e classi di servizio
- ⇒ utilizzo dei metodi statici >> la classe come supervisore dei suoi oggetti
- ⇒ associazione tra oggetti >> squadra di lavoro
- ⇒ massimizzare la coesione >> società organizzata con compiti specializzati



## La Piattaforma Java

### ○ In queste lezioni

- ⇒ principalmente classi del package java.lang

### ○ Il package java.lang

- ⇒ package che contiene tutte le classi considerate indispensabili per la programmazione
- ⇒ è automaticamente “importato” in tutte le classi scritte dall’utente per economia



## Il Package java.lang

### ○ Le classi principali

- ⇒ java.lang.System: include due proprietà statiche pubbliche, System.out e System.in
- ⇒ java.lang.String: la classe per la manipolazione di stringhe immutabili
- ⇒ java.lang.Math: numerosi metodi statici per lo svolgimento delle principali operazioni matematiche



## Boxing e Unboxing

### ○ Altre classi fondamentali di java.lang

- ⇒ le classi “wrapper”
- ⇒ una per ciascun tipo di dato primitivo
- ⇒ Integer, Float, Double, Character, Boolean

### ○ Due funzioni principali

- ⇒ effettuare operazioni sui valori del tipo di base corrispondente
- ⇒ trasformare un dato primitivo in un oggetto

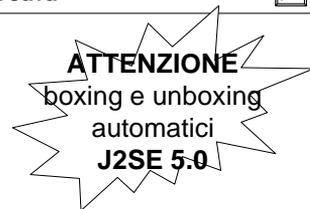


## Boxing e Unboxing

- Oggetto di tipo integer
  - ⇒ oggetto che “incapsula” un valore di tipo int
- Operazione di “boxing” (inscatolamento)
  - ⇒ trasforma un valore int in un oggetto Integer
  - ⇒ `int x = 10; Integer y = new Integer(x);`
- Operazione di “unboxing”
  - ⇒ trasforma un oggetto Integer in un valore int
  - ⇒ `Integer y = new Integer(10); int x = y.intValue();`



## Boxing e Unboxing



- **Attenzione**
  - ⇒ a partire da J2SE 5.0 il boxing e l'unboxing sono diventati automatici
- In particolare, è possibile scrivere

```
// in J2SE 5.0
int x = 10;
Integer y = x; // boxing automatico
Integer z = new Integer(10);
int w = z; // unboxing automatico
```

```
// prima di J2SE 5.0
int x = 10;
Integer y = new Integer(x);
Integer z = new Integer(10);
int w = z.intValue();
```



## Boxing e Unboxing

- Un'operazione tipica con Integer
  - ⇒ estrarre un valore intero a partire da una stringa letta dalla tastiera
  - ⇒ il metodo `public static int parseInt(String s)` di `Integer`
  - ⇒ analogamente per `Float` (`parseFloat()`) e `Double` (`parseDouble()`)
  - ⇒ in `Boolean`:  
`public static boolean getBoolean(String s)`



## JavaDoc

- La forma standard della documentazione
  - ⇒ nome del package, elenco delle classi con breve descrizione e collegamento
  - ⇒ per ogni classe: nome e descrizione
  - ⇒ elenco delle eventuali proprietà pubbliche
  - ⇒ elenco dei metodi pubblici
  - ⇒ altre informazioni (>>)
  - ⇒ questo tipo di documentazione è chiamato `JavaDoc`



## JavaDoc

- JavaDoc

- ⇒ due cose diverse
- ⇒ un formato per la documentazione del codice
- ⇒ uno strumento per produrre la documentazione

- Lo strumento javadoc.exe

- ⇒ fornito a corredo della piattaforma in %JAVA\_HOME%\bin



## JavaDoc

- Utilizzo dello strumento

- ⇒ eseguito su una classe o su un package, crea la documentazione nel formato javadoc
- ⇒ tante pagine HTML

- Sintassi

- ⇒ `javadoc -d cartellaDest nomePackage`
- ⇒ `javadoc -d cartellaDest nomeClasse.java`



## JavaDoc

```
>> it.unibas.utilita.Console  
>> javadoc -d javadoc Console.java
```

- Come vengono specificate le descrizioni
  - ⇒ attraverso commenti particolari immersi nel codice java
- Commento javadoc
  - ⇒ inizia con `/**` e finisce con `*/`
  - ⇒ è possibile indicare informazioni particolari per la classe, le proprietà e i metodi
  - ⇒ es: `@author`, `@return`, `@parameter` ecc.



## JavaDoc

```
>> javadoc -d javadoc calcolatrice
```

- Se nel codice non ci sono commenti
  - ⇒ javadoc genera comunque la documentazione senza descrizioni
  - ⇒ nome delle classi, nome delle proprietà, nomi dei metodi
- In generale
  - ⇒ è uno strumento molto utile anche nel caso in cui non si scrivano commenti espliciti



## Errori Frequenti

- Gli errori più frequenti in Java
  - ⇒ errori nell'utilizzo degli strumenti (compilatore e macchina virtuale) >> CLASSPATH
  - ⇒ errori logici nell'uso degli oggetti
- Errori nell'uso degli strumenti
  - ⇒ attenzione a verificare sempre che la struttura dei file su disco sia corretta e che consenta la visibilità di tutte le classi



## Errori Frequenti

- Errori logici nell'uso degli oggetti
  - ⇒ NullPointerException: si cerca di eseguire un metodo su un riferimento null >> verificare che il riferimento sia stato inizializzato
  - ⇒ ArrayIndexOutOfBoundsException: sconfinamento nell'utilizzo di un array >> verificare l'uso degli indici



## Errori Frequenti

- Un errore sintattico frequente
  - ⇒ “Variable might not have been initialized”:  
mancata inizializzazione di una variabile:  
viola la regola di inizializzazione obbligatoria  
>> inizializzare alla dichiarazione



## Riassumendo

- Riepilogo della Metafora
- La Piattaforma Java
  - ⇒ Boxing e Unboxing
  - ⇒ java.lang
  - ⇒ javadoc
- Errori Frequenti



## 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.