

# Programmazione Orientata agli Oggetti in Linguaggio Java

## Sintassi e Semantica Introduzione

versione 2.1

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: Introduzione >> Sommario



## Sommario

- Introduzione
- Elementi di Base
- Tipi di Dato
- Variabili e Costanti
- Espressioni e Operatori
- Istruzioni

G. Mecca - Programmazione Orientata agli Oggetti

2



## Introduzione

- Sintassi e semantica di Java
  - ⇒ due aspetti fondamentali
- Elementi comuni ai linguaggi procedurali
  - ⇒ variabili, tipi di dato, istruzioni
- Elementi della progr. con gli oggetti
  - ⇒ definizione delle classi e utilizzo degli oggetti
- In questa unità
  - ⇒ ci concentriamo sui primi



## Introduzione

- Elementi comuni ai linguaggi procedurali
  - ⇒ Java è un linguaggio basato sul C++
  - ⇒ molti elementi comuni
  - ⇒ alcune differenze sostanziali
- Differenze principali
  - ⇒ introdotte per favorire la “sicurezza” e la qualità del codice
  - ⇒ vediamo prima gli elementi comuni



## Elementi di Base

- La sintassi è sensibile alle maiuscole
  - ⇒ es: Prova è considerato diverso da prova
- Terminatore di istruzioni
  - ⇒ punto e virgola (;)
  - ⇒ obbligatorio dopo ciascuna istruzione
- Delimitatori dei blocchi
  - ⇒ parentesi graffe



## Elementi di Base

- Commenti
  - ⇒ su riga singola o su più righe
- Commenti su riga singola
  - ⇒ es: // questo è un commento
  - ⇒ es: int x = 1; // un altro commento
- Commenti su più righe
  - ⇒ es: /\* questo è un commento  
su più righe \*/



## Elementi di Base

<http://www.unicode.org>

- L'insieme di caratteri di Java: Unicode
  - ⇒ standard moderno per la rappr. dei caratteri
  - ⇒ obiettivo di internazionalizzazione
  - ⇒ rappresentati con 16 bit
  - ⇒ fino a 65.000 caratteri diversi (di ogni lingua)
  - ⇒ da 0 a 127: corrispondono ai car. ASCII-7
  - ⇒ da 0 a 255: corrispondono ai car. Latin-1

### ○ Esempi

carattere barra inversa

⇒ 'a', '1', '\*', '\n', '\t', '\\', '\u2122'

carattere Unicode di codice esadecimale 2122 (TM)



## Elementi di Base

### ○ Identificatori

- ⇒ sequenza di lettere e cifre
- ⇒ devono cominciare con una lettera
- ⇒ lunghezza arbitraria
- ⇒ possono contenere il carattere \_

### ○ Attenzione

- ⇒ in realtà il concetto di lettera e cifra in Unicode è molto più ampio che in ASCII
- ⇒ es: lettere cirilliche, lettere accentate...



## Tipi di Dato

- Dati in un programma Java
  - ⇒ due categorie
- Componenti
  - ⇒ classi e oggetti
- Dati dei tipi primitivi
  - ⇒ costanti, variabili, parametri
  - ⇒ del tutto analoghi ai corrispondenti nella programmazione procedurale



## Tipi di Dato

- Il sistema di tipi di base di Java
  - ⇒ molto simile a quello dei linguaggi di programmazione procedurale (es: C/C++)
- Ma c'è una differenza importante
  - ⇒ in Java c'è una maggiore enfasi sulla correttezza
  - ⇒ e di conseguenza Java è un linguaggio più fortemente tipato



## Tipi di Dato

tipo	valori	costanti
int	interi a 32 bit (complemento a 2)	234 -3 987345
double	virgola mobile a 64 bit	1.12 -1E-2 8.233D
float	virgola mobile a 32 bit	1.12F -1E-2F 8.233F
char	<b>caratteri Unicode (16 bit)</b>	'a' '0' '\n' '\t' '\\' '\'" '\u12de'
<b>boolean</b>	<b>true o false</b>	true o false

tipo	valori	costanti
byte	interi a 8 bit (complemento a 2)	234 -3
short	interi a 16 bit (complemento a 2)	234 -3 32000
long	interi a 64 bit (complemento a 2)	234L -3L 980002323L



## Tipi di Dato

### ○ Riassumendo

- ⇒ tipi di base
- ⇒ classi e oggetti, tra cui String e array

### ○ Attenzione

- ⇒ in Java non ci sono i record (le strutture)
- ⇒ vengono sostituite e migliorate dalle classi e dagli oggetti
- ⇒ infatti: struttura = insieme di attributi
- ⇒ componente = insieme di attributi e metodi



## Variabili e Costanti

- Due tipi di variabili
  - ⇒ variabili dei tipi primitivi (es: int, float...)
  - ⇒ variabili riferimento
- Variabile di tipo primitivo
  - ⇒ spazio nella memoria identificato da un nome
  - ⇒ la cui natura è stabilita dal tipo
  - ⇒ inizialmente il valore è indefinito
  - ⇒ es: int i, j; float stipendio; char segnoSchedina;



## Variabili e Costanti

- Variabile riferimento
  - ⇒ variabile di tipo riferimento ad una classe
  - ⇒ viene utilizzata come intermediaria per gli oggetti della classe
  - ⇒ i riferimenti sono tipati
  - ⇒ es: java.lang.String s; Calcolatrice c;
  - ⇒ inizialmente il valore è indefinito
  - ⇒ successivamente: id di oggetto oppure null



## Variabili e Costanti

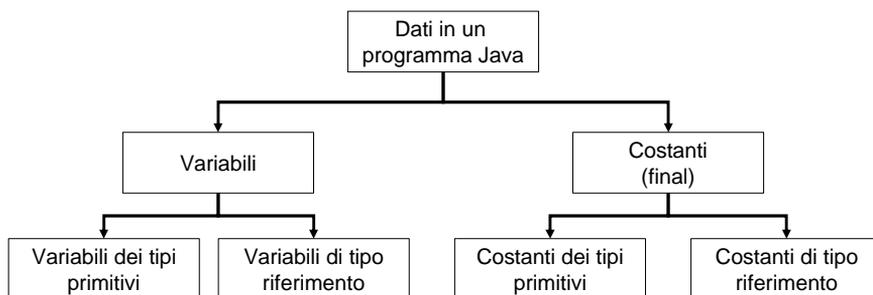
### ○ Costanti simboliche

- ⇒ dati indicati con il modificatore `final`
- ⇒ spazi nella memoria a cui viene assegnato un valore che non può essere cambiato
- ⇒ possono essere dei tipi di base oppure di tipo riferimento
- ⇒ `es: final double PIGRECO = 3.14;`
- ⇒ `es: final java.lang.String s = "Costante";`



## Variabili e Costanti

### ○ Riassumendo





## Espressioni e Operatori

- Espressioni
  - ⇒ composte di operandi e operatori
- Operandi
  - ⇒ variabili, costanti, costanti simboliche
- Operatori
  - ⇒ operatori ordinari
  - ⇒ funzioni predefinite



## Espressioni e Operatori

- Operatori aritmetici
    - ⇒ +, -, \*, /, % (modulo)
  - Operatori logici
    - ⇒ && (and), || (or), ! (not)
  - Operatori di confronto
    - ⇒ ==, !=, >, >=, <, <=
  - Operatori particolari
    - ⇒ ++, --, +=, -=, \*=, /=
- valutati con la tecnica del corto circuito
-



## Espressioni e Operatori

- Le funzioni predefinite

  - ⇒ metodi statici della classe `java.lang.Math`

- Esempi

  - ⇒ `java.lang.Math.sqrt` es: `Math.sqrt(x)`;

  - ⇒ `java.lang.Math.sin` es: `Math.sin(x)`;

  - ⇒ `java.lang.Math.cos` es: `Math.cos(x)`;

  - ⇒ `java.lang.Math.pow` es: `Math.pow(x, y)`; x alla y

  - ⇒ `java.lang.Math.log` es: `Math.log(x)`;



## Espressioni e Operatori

- Inoltre

  - ⇒ la classe `Math` definisce anche due costanti molto utili

  - ⇒ `Math.PI`: valore di pigreco

  - ⇒ `Math.E`: costante di Nepero

- Esempi

  - ⇒ `double` circonferenza;

  - ⇒ `circonferenza = 2 * Math.PI * raggio`;



## Istruzioni

- Istruzioni di Java

- ⇒ dichiarazioni
- ⇒ assegnazioni
- ⇒ strutture di controllo (condizionali e cicli)

- Dichiarazioni

- ⇒ servono a dichiarare costanti e variabili
- ⇒ semantica ordinaria
- ⇒ sintassi: <tipo> <identificatore>;



## Istruzioni

- Per i tipi di base

- ⇒ nulla di nuovo
- ⇒ es: `int i; float f; char segno;`

- Per i riferimenti

- ⇒ attenzione alla forma della dichiarazione
- ⇒ il tipo è (normalmente) il nome di una classe
- ⇒ es: `Calcolatrice c;`
- ⇒ es: `java.lang.String s;`



## Istruzioni

### ○ Assegnazioni

- ⇒ assegnano un valore ad una variabile
- ⇒ viene utilizzata un'espressione
- ⇒ semantica ordinaria

### ○ Sintassi

- ⇒ <variabile> = <espressione>;



## Istruzioni

### ○ Esempi

```
int i, j, k; boolean b; float f;  
i = 3; j = 7;  
k = (i * 17) - ((193 % 19) * Math.exp(0));  
k = ((4 / i) * 127) / Math.cos(0);  
b = (i + j) < 10;  
b = ((i < j) || (i == 7)) && (j != 10);  
f = (i + j) / 3;  
f = (i + j) / 3.0;
```



## Istruzioni

- Strutture di controllo
  - ⇒ strutture condizionali
  - ⇒ cicli
- Strutture condizionali
  - ⇒ if-then-else, if-then
  - ⇒ switch-case
- Sintassi e semantica
  - ⇒ ordinarie



## Istruzioni

### ○ Esempio

```
int i, j;  
i = 3; j = 4;  
if (i > j) {  
    System.out.println("i e' maggiore di j");  
} else {  
    System.out.println("i non è maggiore di j");  
}
```



## Istruzioni

- Cicli
  - ⇒ for e while
- Inoltre
  - ⇒ esiste l'istruzione do-while
  - ⇒ esistono le istruzioni break e continue
- Sintassi e semantica
  - ⇒ ordinarie anche in questo caso



## Istruzioni

- Esempio n. 1

```
for (int i = 0; i < 10; i++) {  
    System.out.println("Valore: " + i);  
}
```
- Esempio n. 2

```
int i = 0;  
while (i < 10) {  
    System.out.println("Valore: " + i);  
    i++;  
}
```



## Istruzioni

- Attenzione
  - ⇒ le istruzioni sono sostanzialmente le stesse dei linguaggi procedurali
  - ⇒ ma servono ad uno scopo totalmente diverso
- Nei linguaggi procedurali
  - ⇒ a definire i sottoprogrammi
- Nei linguaggi a oggetti
  - ⇒ a definire il comportamento dei componenti



## Istruzioni

- Dove si scrivono le istruzioni Java ?
  - ⇒ nei metodi degli oggetti o delle classi
- Modello di esecuzione
  - ⇒ l'esecuzione di una applicazione Java corrisponde all'esecuzione di una sequenza di metodi che si chiamano a vicenda
  - ⇒ a partire da un metodo main
  - ⇒ l'esecuzione dei metodi rispetta il modello basato sulla pila di attivazione



## Riassumendo

- Introduzione
- Elementi di Base
- Tipi di Dato
- Variabili e Costanti
- Espressioni e Operatori
- Istruzioni



## 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.