

Gocce di Java

Tipi di dato primitivi (Modulo 2)

Paolo Lollini

DIMAI – Università degli Studi di Firenze

basato sulle slide del Prof. Crescenzi

- ▶ Concetti base per scrivere un programma
 - ▶ Dati
 - ▶ Variabile
 - ▶ Tipo
 - ▶ Istruzioni
 - ▶ Istruzioni base
 - ▶ Strutture di controllo
 - ▶ Sotto-programmi

- ▶ **Variabili:** usate per immagazzinare dati come numeri e lettere
 - ▶ **Nome:** identificatore con cui riferirsi alla variabile
 - ▶ **Valore:** dato contenuto in variabile
 - ▶ **Tipo:** genere di dati che variabile può contenere

A ogni variabile viene assegnata una locazione di memoria, ma il programmatore usa il nome della variabile per accedere al suo valore.

► Istruzioni di base

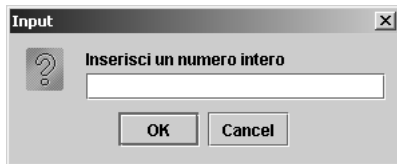
- **Istruzione di assegnamento:** permettono di assegnare valore a variabile
 - Esempio

```
somma = primoNumero+secondoNumero;
```

- **Istruzioni di input e output:** permettono di inserire dati da file o da tastiera e scrivere su disco, monitor o stampante

► Input e output in JAVA--

```
int x=Input.getInt("Inserisci un numero intero");  
System.out.println( "Numero inserito: "+x );
```



- Stampa nella vista di Eclipse Console
- Anche getLong, getDouble e getChar

- ▶ **Dichiarazione:** consente di conoscere nome di variabile, memoria da riservare per variabile e come memorizzare
 - ▶ Consiste di tipo, seguito da lista di nomi (separati da virgole) terminata con punto e virgola
- ▶ In Java ogni variabile deve essere **dichiarata** prima di essere usata!
- ▶ Sintassi
- ▶ Esempio

```
int punteggio;  
char lettera;  
double larghezza, altezza;
```

- ▶ **Identificatore:** nome di variabili (e di altro)
 1. Non possono cominciare con una cifra
 2. Contengono lettere, cifre, simbolo `_` (e simbolo `$`)
 3. Sono *sensibili alle maiuscole*
- ▶ Buone norme
 1. Devono avere un significato e suggerire utilizzo di variabile
 2. Devono iniziare con lettera minuscola e seguire notazione “a cammello”
 - ▶ esempio `sommaPunteggioGiocatori`
 3. Non devono contenere simbolo `$`
- ▶ Errori di ortografia rilevati da compilatore

- └ Tipi di dato primitivi
- └ Tipi di dato primitivi

NOME	TIPO	MEMORIA
byte	intero	1 byte
short	intero	2 byte
int	intero	4 byte
long	intero	8 byte
float	reale	4 byte
double	reale	8 byte
char	carattere	2 byte
boolean	vero/falso	1 byte

Numeri in virgola mobile

- ▶ Rappresentazione **floating point** di x : *mantissa* m ed *esponente* e tali che $x = m \cdot B^e$
 - ▶ Notazione scientifica: $B = 10$, anche indicata con E
 - ▶ Esempio: $2.34E+2$ è 234.0 e $1.234E-3$ è 0.001234
- ▶ Mantissa ed esponente occupano **quantità fissata di memoria** e sono limitati inferiormente e superiormente
 - ▶ Numero cifre decimali in mantissa indica precisione rappresentazione
 - ▶ Numeri `double`: numeri in virgola mobile a precisione doppia

- └ Tipi di dato primitivi
- └ Assegnare valori alle variabili

Istruzione di assegnazione

► Sintassi

variabile = espressione;

► Esempio

```
punteggio = giuste-sbagliate;  
area = larghezza*altezza;  
punteggio = punteggio+1;
```

► Variabile a sinistra segno uguale posta uguale a valore espressione a destra

- Segno di uguale detto *operatore di assegnazione*
- *Espressione*: variabile, costante o espressione costituita da operatori aritmetici

- └ Tipi di dato primitivi
- └ Assegnare valori alle variabili

Valore di ritorno

- ▶ Un'espressione come $(base * altezza) / 2$ produce un valore
 - ▶ Se il valore di `base` è 20 e quello di `altezza` è 9 allora il valore prodotto è 90
- ▶ questo valore è detto **valore di ritorno** dell'espressione
- ▶ Quindi non diremo che il “valore prodotto è 90”
- ▶ ma che “il **valore restituito** è 90”

Useremo la stessa terminologia quando vedremo i “metodi”.

Prima viene calcolato il valore dell'espressione alla destra del `=` e tale valore viene poi usato per impostare il valore della variabile alla sinistra del `=`

```
double base;  
double altezza;  
double area;  
base = 4.0;  
altezza = 8.0;  
area = (base*altezza)/2;  
System.out.println( "Area: "+area );
```

- └ Tipi di dato primitivi
- └ Assegnare valori alle variabili

Formula di Gauss

- Somma primi n numeri interi

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$



- └ Tipi di dato primitivi
 - └ Assegnare valori alle variabili

Formula di Gauss

- Somma primi n numeri interi

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

```
int n;  
n = Input.getInt( "Inserisci il numero n" );  
int somma;  
somma = n*(n+1)/2;  
System.out.println( "Somma: "+somma );
```

- └ Tipi di dato primitivi
 - └ Assegnare valori alle variabili

- ▶ Stesso identificatore può apparire in due lati operatore di assegnazione

```
int contatore;  
contatore = 5;  
System.out.println( "Contatore: "+contatore );  
contatore = contatore-1;  
System.out.println( "Contatore: "+contatore );  
contatore = contatore-1;  
System.out.println( "Contatore: "+contatore );
```

- ▶ Sottrae uno a valore di contatore e usa risultato per impostare nuovo valore di contatore