



Laboratorio di Programmazione

Laurea in Bioinformatica

Web: <http://www.scienze.univr.it/fol/main?ent=oi&id=28023&lang=it>

Docente: *Carlo Drioli*

Email: *drioli@sci.univr.it*

Nicola Drago

Lucidi a cura di

Carlo Drioli

Federico Fontana

Lezione 3

Sommario

- Input da tastiera
- I tipi
- Esercizi

Input da tastiera

- Si vuole assegnare un valore a una variabile, inserendolo da tastiera in fase di esecuzione del programma.
- L'input da tastiera richiede operazioni complicate o l'utilizzo di una libreria con funzioni estese. Noi useremo la classe semplificata "**ConsoleInputManager**" contenuta nella libreria "**prog**" fornita a corredo del teso di riferimento.

Input da tastiera

Istallazione della libreria “prog” e altre modifiche delle variabili d’ambiente

- Copiare il file “prog.tgz” da <https://amarena.sci.univr.it/edu/pub/bscw.cgi/3034346> in una directory della propria home (ad es. local_java_dir)
- Decomprimere il file mediante il comando `tar -xzf prog.tgz`
- Modificare il file `.bashrc` nella propria home aggiungendo le seguenti righe:

```
JAVAPATH=/usr/local/jdk1.5.0_09
CLASSPATH=$JAVAPATH/lib:~/home/accounts/.../<ACCOUNT>/<LOCAL_JAVA_DIR>
PATH=$JAVAPATH/bin:$PATH
export CLASSPATH
export PATH
```

Input da tastiera

Uso della classe di input da tastiera della libreria “prog”

- Direttiva di inclusione della libreria

```
import prog.io.*;
```

- Costruzione di un oggetto `ConsoleInputManager` di nome *tastiera*

```
ConsoleInputManager tastiera = new ConsoleInputManager(),
```

- Invocazione del metodo *readInt* dell'oggetto *tastiera* per la lettura di un intero

```
int numero = tastiera.readInt();
```

Input da tastiera

Un semplice programma si test della libreria:

```
import prog.io.*;

class Input{
    public static void main(String[] args) {
        ConsoleInputManager tastiera new ConsoleInputManager();
        int numero;

        System.out.print("Digita un numero: ");
        numero = tastiera.readInt();
        System.out.print("Hai inserito il numero: ");
        System.out.println(numero);
    }
}
```

Input da tastiera: esercizio

- Scrivere ed eseguire un programma che chieda di inserire da tastiera la base e l'altezza di un triangolo, ne calcoli l'area e visualizzi il risultato:

Calcolo dell'area del Triangolo

Inserire la base: ...

Inserire l'altezza: ...

L'area è = ...

Input da tastiera: soluzione esercizio

```
import prog.io.*;
class areaTriangolo2{
    public static void main(String[] args){

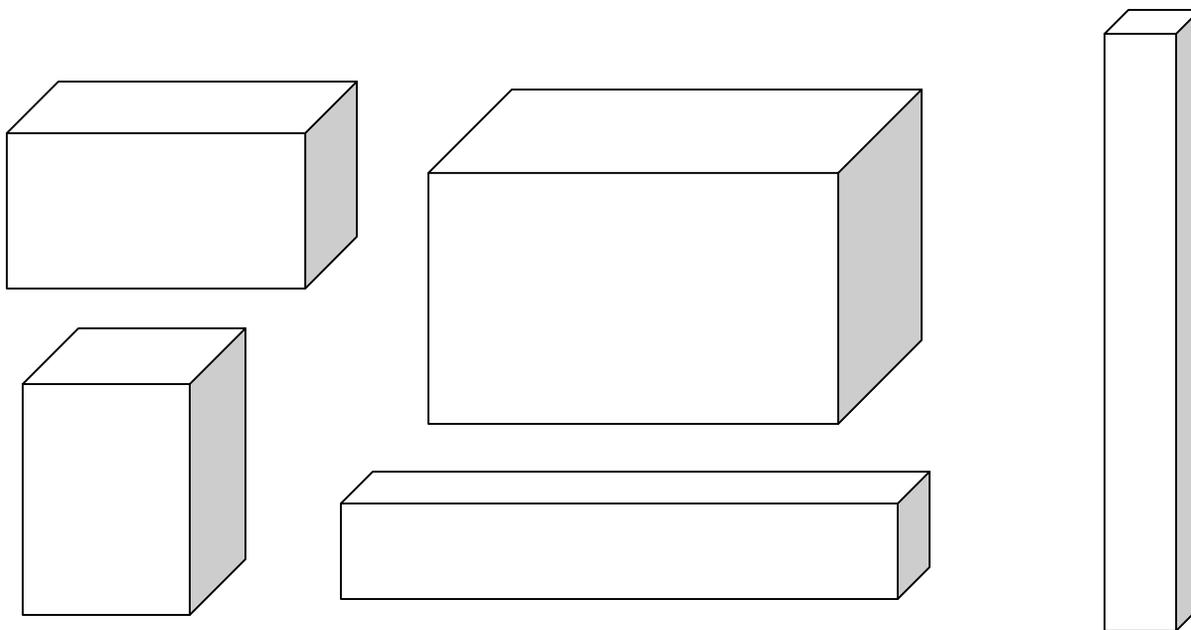
        ConsoleInputManager keyboard = new ConsoleInputManager();
        int baseTriangolo, altezzaTriangolo, areaTriangolo;

        System.out.println("Calcolo dell'AREA del TRIANGOLO ");
        System.out.print("Inserire la base: ");
        baseTriangolo = keyboard.readInt();
        System.out.print("Inserire l'altezza: ");
        altezzaTriangolo = keyboard.readInt();
        areaTriangolo = (baseTriangolo * altezzaTriangolo) / 2;
        System.out.print("L'area e': ");
        System.out.println(areaTriangolo);
    }
}
```

Tipi di dati

- Una variabile è un “**contenitore**”. ma di contenitori ne abbiamo di FORME e DIMENSIONI diversi:

TIPI



Tipi di dati in Java

- Tipi **primitivi**

Contenitori di valori di un tipo primitivo (es. int, double, char)
Le operazioni sui tipi primitivi sono predefinite nel linguaggio

- Tipi **riferimento** (\leftrightarrow classi)

Contenitori di riferimenti per accedere ad oggetti (strutture complesse con componenti, metodi, etc.)

I tipi riferimento sono virtualmente in numero infinito: il programmatore può crearne di nuovi a suo piacimento

Tipi primitivi

Nome tipo	Natura valore	Dimensione
char	carattere	2 bytes
byte	intero	1 byte
short	intero	2 bytes
int	intero	4 bytes
long	intero	8 bytes

Il tipo char

- Usato per rappresentare **caratteri**, cioè lettere dell'alfabeto (sia maiuscole che minuscole), cifre decimali (0, 1, ..., 9), simboli speciali
- Spesso per rappresentare i caratteri si usano 8 bit (1 byte) e codice ASCII
- In Java vengono usati **16 bit** e codice UNICODE, per un totale di $2^{16}=65535$ caratteri rappresentabili (lettere usate negli alfabeti occidentali, simboli speciali, ideogrammi, etc.)
- Esempio:

```
char ch1 = 'a' ;
```

```
char ch2 = 'A' ;
```

I tipi per rappresentare n.i interi

- Codifica di segno e valore

N bytes = 8 * N bits

X bits rappresentano 2^X valori

1 bit viene usato per il segno :

2^{X-1} valori negativi e $2^{X-1}-1$ valori non negativi

- Esempio

1 byte = 8 bits

1 bit viene usato per il segno :

7 bits rappresentano 256 valori,
cioè l'intervallo [-128,127]

I tipi per rappresentare n.i interi

- Range di valori rappresentabili con i tipi byte, short, int e long:

byte:	$[- 2^7 , 2^7 - 1]$	1 byte = 8 bits
short:	$[- 2^{15} , 2^{15} - 1]$	2 bytes = 16 bits
int:	$[- 2^{31} , 2^{31} - 1]$	4 bytes = 32 bits
long:	$[- 2^{63} , 2^{63} - 1]$	8 bytes = 64 bits

Altri tipi primitivi

Nome tipo	Natura valore	Dimensione
float	reale	4 bytes
double	reale	8 bytes
boolean	true o false	1 bit

I tipi per rappresentare n.i reali

- I numeri reali costituiscono un insieme non numerabile, impossibile da rappresentare in modo esatto in un calcolatore \Rightarrow si usano rappresentazioni **approssimate**
- Rappresentazione con virgola mobile (floating point). Range dei valori float/double:

<i>type</i>	<i>Size</i>		<i>Range</i>	<i>Precision</i>
<i>name</i>	<i>bytes</i>	<i>bits</i>	<i>approximate</i>	<i>in decimal digits</i>
float	4	32	+/- 3.4 * 10 ³⁸	6-7
double	8	64	+/- 1.8 * 10 ³⁰⁸	15

■ Esempio

```
float f = 3.0f;  
double d = 3.0d;  
double r1 = 1.2345e5; // r1=123450.0  
double r2 = 1.2345e-5; // r1=0.000012345
```

I tipi primitivi: assegnamento ed espressioni

- E' possibile assegnare un valore a una variabile attraverso una costante o un'espressione:

```
int f = 3;  
int d = 5;  
int fd = f+d-4;
```

- Dichiarazione di piu' variabili dello stesso tipo e assegnamento di valore:

```
tipo  variabile_1 = Espressione_1,  
      variabile_2 = Espressione_2,  
      ....      ..... ;
```

I tipi primitivi: assegnamenti speciali

- Prodotto

`triplo*=3;` **equivale a** `triplo=triplo*3;`

- Incremento

`contatore+=1;`
`contatore++;` **equivalgono a** `contatore = contatore +1;`

- Decremento

`contatore-=1;`
`contatore--;` **equivalgono a** `contatore = contatore -1;`

I tipi primitivi: assegnamento a priori e a posteriori

- Incremento a priori

```
int n=3, m=4;  
int ris = n*(++m); //m=5, ris=15
```

- Incremento a posteriori

```
int n=3, m=4;  
int ris = n*(m++); //ris=12, m=5
```

- Decremento ha lo stesso comportamento

I tipi: conversione di tipo

- Si può assegnare un valore a una variabile dello stesso tipo o di tipo più **generale** senza perdita di informazione:

```
double numero = 80 ; //80 → 80.0
```

```
int numeroIntero = 7.7 ; ERRATO
```

- I tipi numerici rispettano la seguente gerarchia:

byte → short → int → long → float → double

- Nell'assegnamento con espressioni, se due operandi hanno tipo diverso, il risultato è del tipo più generale

Esempi: $9.0/2 \rightarrow 4.5$ $9/2 \rightarrow 4$ (**troncamento**)

I tipi: conversione esplicita di tipo

- Il type cast (conversione esplicita di tipo) permette di assegnare ad una variabile il valore di un tipo piu' generale

```
tipo var = (tipo) espressione;
```

- Esempi:

```
double dist = 9.0;
```

```
int punti = dist; errato
```

```
int punti = (int)dist; corretto, punti = 9
```

```
double dist = 9.8;
```

```
int punti = (int)dist; punti = 9 (troncamento!)
```

I tipi: precedenza degli operatori

■ Regole di precedenza

Prima: gli operatori unari: + - ++ -- !

Poi: gli operatori binari moltiplicativi : * / %
(%: resto della divisione intera)

Infine: gli operatori binari additivi : + -

■ Esempio

base+peso*valore \neq (base + peso)*valore

Esercizi

- Calcolare il doppio di un numero fornito da tastiera
- Dato il lato, calcolare il perimetro e l'area del quadrato
- Dati due numeri A, B da tastiera, assegnare ad una variabile booleana il valore del confronto $A > B$
- Dati i pesi di tre oggetti, calcolare il peso medio
- Dato il prezzo di un prodotto e l'aliquota IVA, calcolare il prezzo totale