

# Laboratorio di programmazione

9 novembre 2016 (Trump Day!)

- Lo scheletro dei vostri programmi deve essere

```
import java.io.*;
import java.util.*;

public class ... {
    public static void main( String[] args ) throws Exception {
        ...
    }
}
```

dove i due gruppi di puntini devono essere sostituiti dal nome scelto per la classe e dal suo codice. Per il nome potete usare lettere maiuscole, minuscole e numeri, purché il nome inizi con una lettera. Ricordatevi che il nome del file sorgente *deve* essere identico al nome della classe.

Per leggere dati da tastiera dovete creare uno `Scanner`:

```
Scanner scanner = new Scanner( System.in );
```

Una volta creato lo `Scanner` (va creato *una sola volta*) potete leggere l'input da tastiera dell'utente in vari modi:

- Per leggere una riga: `String line = scanner.nextLine();`
- Per leggere una parola: `String line = scanner.next();`
- Per leggere un intero: `int x = scanner.nextInt();`
- Per leggere un long: `long l = scanner.nextLong();`
- Per leggere un double: `double d = scanner.nextDouble();`

- Per stampare, potete usare `System.out.println(...)`, che va a capo, e `System.out.print(...)`, che non va a capo.
- Se avete più stringhe, le potete concatenare con l'operatore `+`, che si occupa anche di convertire in stringa qualunque altro oggetto o variabile:

```
Scanner scanner = new Scanner( System.in );
System.out.print( "Scrivi qualcosa: " );
String input = scanner.nextLine();
System.out.println( "Hai scritto " + input );
```

- Se volete leggere un numero:

```

Scanner scanner = new Scanner( System.in );
System.out.print( "Scrivi un numero: " );
int x = scanner.nextInt();
int quadrato = x * x;
System.out.println( "Hai scritto " + x + " che ha quadrato " + quadrato );

```

1. Scrivete un programma che, dopo aver richiesto in input un numero intero  $n$ , stampi la corrispondente tabellina, moltiplicando  $n$  per i numeri interi da 1 a 10, come indicato nel seguente esempio di esecuzione:

```

Inserisci un numero: 9
1 x 9 = 9
2 x 9 = 18
3 x 9 = 27
4 x 9 = 36
5 x 9 = 45
6 x 9 = 54
7 x 9 = 63
8 x 9 = 72
9 x 9 = 81
10 x 9 = 90

```

2. Create una classe Java che legge un intero inserito dall'utente e ne stampa i divisori.

```

Inserisci un numero: 12
1 2 3 4 6 12

```

3. Create una classe Java che legge un intero inserito dall'utente e dice se è primo.

```

Inserisci un numero: 12
Composto

```

```

Inserisci un numero: 13
Primo

```

4. Considerate la seguente regola: dato un numero intero positivo  $n$ , se  $n$  è pari lo si divide per 2, se è dispari lo si moltiplica per 3 e si aggiunge 1 al risultato. Quando  $n$  è 1 ci si ferma.

Questa semplice regola permette di costruire delle sequenze: la sequenza che si costruisce a partire dal numero  $n$  è detta *sequenza di Collatz di  $n$* . Ad esempio, la sequenza di Collatz di 7 è:

```
7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

È un noto problema aperto stabilire se ogni sequenza di Collatz termina (cioè, se arriva a 1).

Scrivete un programma che chiede all'utente un numero e mostra la sequenza di Collatz del numero (con tanto di lunghezza).

### Esempio di funzionamento

```
Numero: 7
7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
Lunghezza: 17
```

```
Numero: 9
9 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
Lunghezza: 20
```

5. Create una classe Java che legge un intero inserito dall'utente e disegna un quadrato di asterischi con il lato uguale all'intero.

```
Inserisci un numero: 4
****
****
****
****
```

```
Inserisci un numero: 3
***
***
***
```

Provate ora a disegnare il quadrato con l'interno vuoto.

```
Inserisci un numero: 4
****
*  *
*  *
****
```

```
Inserisci un numero: 3
***
*  *
***
```

6. Scrivete un programma che, letto un numero intero  $n$ , emetta in output una "figura" di  $n$  righe costituite ciascuna da  $n$  caratteri, alternando righe costituite solo da  $*$  e righe costituite solo da  $+$ .

### Esempio di funzionamento

```
Inserisci un numero: 5
*****
+++++
*****
+++++
*****
```

7. Scrivete altri quattro programmi, simili a quello dell'esercizio precedente ma che disegnano figure diverse. Tutti questi programmi prendono in input un intero  $n$  e producono in output una figura di  $n$  righe e  $n$  colonne, costituita da \* e +. La differenza fra un programma e l'altro è il modo in cui sono alternati i caratteri, come indicato negli esempi che seguono.

**Attenzione:** nel caso dell'ultimo programma, potete disegnare la figura come indicato solo se  $n$  è pari. Se l'utente inserisce valore dispari, fate in modo che il programma termini stampando un messaggio d'errore.

Inserisci un numero: 5

```
*+*+*
*+*+*
*+*+*
*+*+*
*+*+*
```

Inserisci un numero: 6

```
*+*+*+*
+*+*+*
*+*+*+*
+*+*+*
*+*+*+*
+*+*+*
```

Inserisci un numero: 7

```
*+++++++
* *+++++
* *+++++
* *+++++
* *+++++
* *+++++
* *+++++
* *+++++
```

Inserisci un numero: 10

```
++++*++++
++++*++++
++*++++*++
+*++++*+*
*++++*+*+
*++++*+*+
+*++++*+*
++*++++*++
++++*++++
++++*++++
```

8. Data una sequenza di numeri reali  $x_1, x_2, \dots, x_n$ , sono dette rispettivamente

(a) *media aritmetica* il valore  $(x_1 + x_2 + \dots + x_n)/n$ ,

(b) *media geometrica* il valore  $\sqrt[n]{x_1 x_2 \dots x_n} = \exp(\log(x_1 x_2 \dots x_n)/n)$ ,

(c) *media quadratica* il valore  $\sqrt{(x_1^2 + x_2^2 + \dots + x_n^2)/n}$ ,

(d) *media armonica* il valore  $n / \left( \frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n} \right)$ .

Scrivete un programma che legga una sequenza di numeri reali positivi in input, terminati da -1, e ne stampi le quattro medie. Per scrivere il programma, potete usare i metodi `Math.sqrt(double)`, `Math.exp(double)` e `Math.log(double)`, che calcolano, rispettivamente, la radice quadrata, l'esponenziale e il logaritmo naturale del loro argomento.

**Suggerimento.** Naturalmente, per leggere la sequenza di numeri reali dovete usare un ciclo. Durante questo stesso ciclo potete tenere traccia dei valori che vi servono per calcolare le varie medie. Per esempio, per poter calcolare la media aritmetica vi servirà una variabile `n` in cui tenere il *numero* di valori letti fino a quel momento e una variabile `tot` contenente la *somma* dei valori letti fino a quel momento (di che tipo devono essere queste due variabili?). Ogni volta che viene letto un nuovo valore dovrete incrementare di uno la variabile `n` e incrementare la variabile `tot` del valore letto (entrambe le variabili devono essere inizializzate a zero prima del ciclo). State attenti a non incrementare né `n` né `tot` quando leggete il valore -1, che ha come unico scopo quello di terminare la sequenza degli input.

#### Esempio di input

```
2 4 5 -1
```

#### Esempio di output

```
Media aritmetica: 3.666667
Media geometrica: 3.419952
Media quadratica: 3.872983
Media armonica: 3.157895
```