

ALESSANDRO FALETTI

Java per principianti

*Copyright © 2025 by Alessandro Faletti*

*All rights reserved. No part of this publication may be reproduced, stored or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise without written permission from the publisher. It is illegal to copy this book, post it to a website, or distribute it by any other means without permission.*

*First edition*

*This book was professionally typeset on Reedsy.*

*Find out more at [reedsy.com](https://reedsy.com)*

# Contents

|   |    |
|---|----|
| <i>Introduzione a Java</i>                | iv |
| 1 Hello World!                            | 1  |
| 2 Input e variabili                       | 6  |
| 3 Condizioni logiche                      | 13 |
| 4 Recap                                   | 17 |
| 5 Iterazioni                              | 21 |
| 6 Array                                   | 27 |
| 7 Matrici                                 | 33 |
| 8 Procedure                               | 38 |
| 9 Funzioni                                | 42 |
| 10 Esercizio recap (Prenota posti cinema) | 45 |
| 11 Soluzione esercizi                     | 47 |

# Introduzione a Java

Benvenguto/a, prima di partire con la spiegazione voglio rispondere a delle semplici e comuni domande su Java.

1. Cosa è Java? Java è un linguaggio di programmazione inventato nel 1995 da James Gosling. I linguaggi di programmazioni sono divisi in alto livello e basso livello. Alto livello sono i linguaggi che cercano di imitare il più possibile il linguaggio umano. Al contrario i linguaggi di basso livello sono simili al linguaggio macchina e difficili da decifrare e imparare. Il nostro Java è un linguaggio ad alto livello.
2. A cosa serve? Java trova molte applicazioni nello sviluppo di: applicazioni android, programmi dekstop, applicazioni web, sistemi aziendali e software gestionali, intelligenza artificiale, robotica, apprendimento della programmazione e automazione di problemi comuni.
3. Perché scegliere Java rispetto ad altri linguaggi? Si può scegliere Java perchè è un linguaggio solido e maturo che ha oltre 25 anni di età, si trova nella maggior parte dei programmi che utilizziamo, facilita l'apprendimento di altri linguaggi di programmazione, è molto richiesto nel lavoro ed è molto facile da imparare.
4. Vale la pena imparare un linguaggio di programmazione con l'avvento dell'AI? Imparare un linguaggio di pro-

grammazione migliora il nostro apprendimento, il nostro ragionare e amplia le nostre possibilità di carriera, in alcuni settori ad esempio: medico, aeronautico, militare ecc... l'intervento umano è essenziale e l'utilizzo dell'AI è quasi vietato.

## DOVE ESEGUIRLO

Dopo queste 4 domande e 4 risposte iniziamo a capire dove possiamo scrivere

il nostro programma. Quest'ultimo viene scritto in un IDE (Integrated

Development Environment o Ambiente di Sviluppo Integrato), un software che

fornisce un insieme di strumenti per scrivere, modificare, testare e debuggare

il codice, risolvendo eventuali problemi in modo efficiente. L'IDE può essere

online o offline, per comodità noi useremo un IDE online (GDB online per

esempio).

## DOVE SALVARE IL CODICE

Possiamo salvarlo direttamente nell'IDE oppure possiamo caricarlo su GitHub

creando il nostro portfolio per mostrare le nostre skill.



Su GitHub possiamo anche collaborare a progetti OpenSource mettendo in pratica ciò che abbiamo imparato.

# 1

## Hello World!

Dopo aver fatto una panoramica di questo linguaggio iniziamo a scrivere codice. Quando si inizia a imparare un qualsiasi linguaggio di programmazione, il primo programmino è stampare la frase HELLO WORLD!. Spiegato il problema dobbiamo cliccare sulla barra di ricerca e cercare un IDE online Java (o offline se preferite, però è più lungo perchè bisogna installarlo e fare il setup).

## JAVA PER PRINCIPIANTI

Ide online Java



Try for free · Get Started for Free · 40% Off Starter Suite · Watch the Demo Now



<https://www.online-java.com> ::

### Online Java - IDE, Code Editor, Compiler

Build, Run & Share Java code online using online-java's IDE for free. It's one of the quick, robust, powerful online compilers for java language. Don't worry about setting up java environment in your local. Now Run the java code in your favorite browser instantly. Getting...

[Terms & Conditions](#) · [Online Java Compiler](#) · [Online Java Editor](#) · [Online Java IDE](#)



<https://www.onlinergdb.com> > [online\\_java\\_compiler](#) ::

### Online Java Compiler - online editor

OnlineGDB is online IDE with java compiler. Quick and easy way to run java program online.



<https://www.jdoodle.com> > [online-java-compiler](#) ::

### Online Compiler and Editor/IDE for Java, C, C++, PHP, Python,...

JDoodle is an Online Compiler, Editor, IDE for Java, C, C++, PHP, Perl, Python, Ruby and many more. You can run your programs on the fly online, and you can save and share them with others. Quick and Easy way to compile and run programs online.

Per comodità io utilizzo online GDB perchè ho imparato con quello, oppure con Programiz (non nell'elenco), comunque vanno bene tutti. Dopo essere entrati in GDB Compiler, questa è la vostra schermata.

```
Main.java : 1  ****
2
3
4
5
6
7
8
9
10
11
12
13
14
```

Online Java Compiler.  
Code, Compile, Run and Debug java program online.  
Write your code in this editor and press "Run" button to execute it.

```
public class Main
{
    public static void main(String[] args) {
    }
}
```

Fatto questo setup esaminiamo la struttura base del codice Java uguale per tutti i programmi.

- **Il codice scritto in giallo** è facoltativo perchè è un commento. Un commento si fa anteponendo alla linea di codice un doppio // (forward slash) o nel caso di commenti su più righe, (come in questo caso) si ingloba la frase in un /\* \*/.
- **public class Main{}** è una classe pubblica: class serve a definire una classe, mentre public indica che può essere usata da altre parti del programma. Il nome della classe è Main, e tutto il nostro codice deve essere scritto al suo interno. Lo possiamo immaginare come un contenitore.
- **public static void main(String[] args){}** si trova all'interno della classe pubblica ed è la funzione principale (main), per i nostri primi programmi scriveremo solo al suo interno, poi più avanti creeremo noi altre funzioni, procedure.

## STAMPARE HELLO WORLD!

Dopo aver spiegato la struttura base del codice e dove scrivere, impariamo a

stampare una stringa. Nell'apprendimento di un linguaggio di

programmazione il primo programma che si crea è stampare la frase Hello

Word!. In Java la funzione stampa è **System.out.println()** o **System.out.print()**.

La differenza tra i due è che il primo stampa e dopo va a capo in automatico

mentre il secondo no. Per stampare una frase bisogna racchiuderla in dei

doppi apici ““, per esempio “Hello world!”. Mentre per i

numeri no. Fatta

questa introduzione teorica procediamo a stampare la frase Hello world!, ci

mettiamo nel public static void main e inseriamo:

```
System.out.println("Hello world!");
```

Una cosa **MOLTO** importante da specificare è che in Java alla fine di una riga bisogna sempre mettere il punto e virgola (;). Il risultato dovrebbe essere stato il seguente:

The screenshot shows a Java code editor with the following code:

```

8
9 public class Main
10 {
11     public static void main(String[] args) {
12         System.out.println("Hello world!");
13     }
14 }
15

```

Below the code, a terminal window displays the output:

```

Hello world!
...Program finished with exit code 0
Press ENTER to exit console.

```

Come possiamo vedere, Hello world! viene mostrato sulla console.

## ESERCIZI

Tengo a precisare che la soluzione degli esercizi si trova alla fine del libro e che

non devono essere per forza uguali ma devono avere lo stesso risultato.

Stampare il tuo nome

Stampare il tuo nome e a capo il tuo cognome

HELLO WORLD!

Stampare il tuo nome, sulla stessa linea il tuo cognome e a capo la tua età.

Se te la senti puoi provare a inventarti degli esercizi o a cercare online.

# 2

## Input e variabili

Dopo aver visto la funzione stampa, un'altra funzione fondamentale è quella riguardante l'input. Questa funzione prende in input (in ingresso) delle variabili, ma cosa sono le variabili?

### VARIABILI

Le variabili sono contenitori dove possiamo salvare dei dati di varia tipologia

(numeri, stringhe caratteri, booleani ecc...) per poi riutilizzarli nel

programma. In Java prima di utilizzare una variabile bisogna dichiararla nel

seguente modo:

```
Tipo_variabile nome_variabile;
```

Nel caso dei numeri interi:

```
int numero;
```

Nel caso di stringhe:

```
String nome;
```

Nel caso di numeri decimali:

```
double numero;
```

Nel caso di caratteri:

```
char carattere;
```

Nel caso di booleani (true/false):

```
boolean condizione;
```

Qui ci sono i principali tipi di variabili ma ce ne sono di più.

Dopo aver dichiarato una variabile bisogna assegnarle un valore.

```
!!Si poteva farlo anche direttamente nella  
dichiarazione ma nei casi più complessi l'input lo  
decide l'utente!!
```

Per assegnare un valore direttamente nella dichiarazione si fa nel seguente modo (per esempio nei numeri interi ma la modalità è la stessa):

```
int numero=1;
```

Nelle stringhe :

```
String nome="Alessandro";
```

Nel caso di caratteri:

```
char carattere='a';  
!!Importante che le stringhe devono essere contenute  
nelle doppio virgolette (""), mentre i caratteri nelle  
singole virgolette ('').!!
```

Nel caso di numeri decimali:

```
double numero=6.01;
```

Nel caso di booleani:

```
boolean condizione=true;
```

Se volevamo solamente dichiarare le variabili e poi assegnare un valore allora non dovevamo più indicare il tipo di variabile ma solo il nome della variabile l'uguale e il nuovo dato con alla fine il punto virgola, per esempio:

```
int numero;  
numero=1;
```

## INPUT

Dopo aver fatto una panoramica sui tipi di variabili, come dichiararle e come assegnarle i valori, impariamo come si può prendere dall'utente un deter-

minato valore.

Prima di tutto al di fuori del main bisogna importare la libreria Scanner (utilizzata per l'input).

```
Main.java  :
1- import java.util.Scanner;
2 public class Main
3 {
4-     public static void main(String[] args) {
5-         }
6 }
7
```

Fatto questo all'interno della funzione main dichiariamo lo Scanner con il nome scanner.

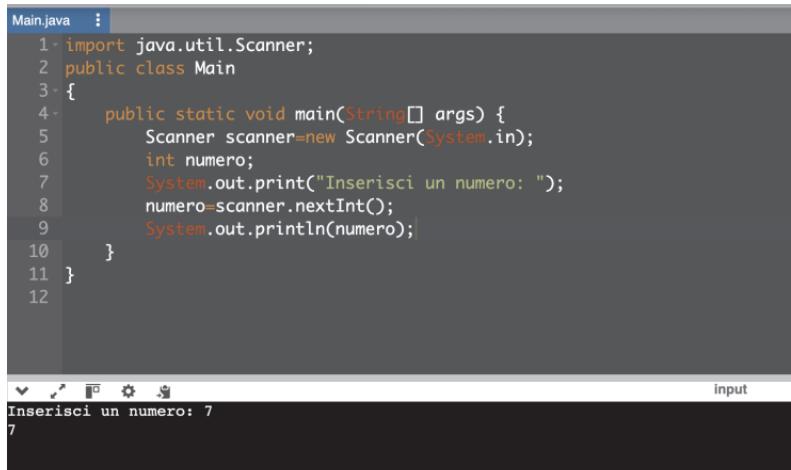
```
Main.java  :
1- import java.util.Scanner;
2 public class Main
3 {
4-     public static void main(String[] args) {
5-         Scanner scanner=new Scanner(System.in);
6     }
7 }
8
```

Grazie a questo possiamo adesso prendere in input un valore e assegnarlo ad una variabile.

A seconda del tipo di variabile lo scanner è diverso, in generale: scanner.nextInt(); —> per gli interi.

scanner.nextLine();—>per le stringhe.  
 scanner.next().charAt(0);—>per i caratteri.  
 E altri ancora, ma per gli esercizi che faremo serviranno solo questi!

Fatto questo possiamo fare un esempio di un programma che prende in input un numero e stampi quel numero.



```
Main.java
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         int numero;
7         System.out.print("Inserisci un numero: ");
8         numero=scanner.nextInt();
9         System.out.println(numero);
10    }
11 }
12
```

Inserisci un numero: 7  
input

Per farlo un pò più bello e includere nell'output finale anche una frase bisogna utilizzare la **concatenazione**. Per stampare variabili e frasi nella stessa funzione stampa dobbiamo utilizzare il segno di operazione +.

Per esempio se avessimo voluto scrivere nell'output finale Il numero che hai scelto è il 7,  
 dobbiamo fare nel seguente modo:

```
System.out.println("Il numero che hai scelto è il:  
" +numero);
```

## OPERAZIONI TRA VARIABILI

Con le variabili si possono fare anche delle operazioni, per esempio somma (+), sottrazione (-), divisone (/), moltiplicazione (\*), trovare resto(%), elevamento a potenza (Math.pow()) con una precedente importazione della libreria) ecc...

Per esempio se volessimo fare un programma che date due variabili a e b si fa la somma, la differenza, la moltiplicazione e la divisione si fa in questo modo:

```
Main.java : |  
1- import java.util.Scanner;  
2 public class Main  
3 {  
4-     public static void main(String[] args) {  
5         Scanner scanner=new Scanner(System.in);  
6         int a;  
7         int b;  
8         System.out.println("Inserisci un numero: ");  
9         a=scanner.nextInt();  
10        System.out.println("Inserisci un numero: ");  
11        b=scanner.nextInt();  
12        int somma=a+b;  
13        int differenza=a-b;  
14        int moltiplicazione=a*b;  
15        int divisione=a/b;  
16        System.out.println(somma+" "+differenza+" "+moltiplicazione+" "+divisione);  
17    }  
18 }  
19
```

Inserisci un numero:  
2  
Inserisci un numero:  
3  
5 -1 6 0

## ESERCIZI

Prendi in input 1 numero int e stampalo.

Prendi in input 1 numero int e stampalo con la concatenazione, hai scelto il numero.

Prendi in input un nome e un cognome e stampali con la concatenazione.

# 3

## Condizioni logiche

In java, in tutti i linguaggi di programmazione ma come anche nella vita, ci sono delle decisioni da prendere. In informatica queste decisioni le decidiamo noi ma è il computer a controllarle, per esempio una condizione logica è la seguente: se l'utente inserisce un numero maggiore di 6 scrivi promosso altrimenti bocciato. In java le condizioni logiche possono essere l'if o lo switch.

### **IF-STATEMENT**

La struttura di base è:

```
if(condizione){  
    fai qualcosa;  
}  
else{  
    fai qualcosa;  
}
```

## JAVA PER PRINCIPIANTI

The screenshot shows a Java code editor with the file 'Main.java' open. The code contains a single if-else block that prints 'Promosso.' if the user inputs a number greater than 6, and 'Bocciato.' otherwise. The output window below shows the program's execution: it asks for a number, receives '7' as input, and then prints 'Promosso.'.

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         System.out.println("Inserisci un numero: ");
7         int numero=scanner.nextInt();
8         if(numero>6){
9             System.out.println("Promosso.");
10        }
11        else{
12            System.out.println("Bocciato.");
13        }
14    }
15 }
16
```

Inserisci un numero:  
7  
Promosso.

Se ci fossero state invece più condizioni allora dovevamo inserire else-if.

The screenshot shows a Java code editor with the file 'Main.java' open. The code uses a nested if-else-if structure to determine if a number is positive, zero, or negative. The output window shows the program asking for a number, receiving '-2' as input, and then printing 'Negativo'.

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         System.out.println("Inserisci un numero: ");
7         int numero=scanner.nextInt();
8         if(numero>0){
9             System.out.println("Positivo");
10        }
11        else if(numero==0){
12            System.out.println("Zero");
13        }
14        else if(numero<0){
15            System.out.println("Negativo");
16        }
17    }
18 }
```

Inserisci un numero:  
-2  
Negativo

In questo caso abbiamo fatto inserire un numero e se questo era

maggiore di 0 stampava che era positivo, uguale a zero stampava uguale a zero, se era minore di zero stampava che era negativo.

Come si può notare nel (numero==0) non abbiamo utilizzato il singolo = perchè indica l'assegnazione (come nelle variabili) ma il doppio = indica il controllo.

Qui sorge un problema, se volessimo stampare nel caso dello 0 che è positivo? Nella condizione dobbiamo specificare allora che è maggiore e uguale ( $\geq$ ) stessa cosa nel caso se fosse minore e uguale ( $\leq$ ).

## SWITCH-CASE

Un altro esempio di condizione logica è lo switch-case, questo si usa soprattutto quando

hai più scelte da valutare in base a un valore preciso (come numeri o stringhe fisse).

Un esempio sarebbe dati i numeri da 0-5 stampare il numero scelto dall'utente.

La struttura di base è:

```
switch (variabile){
    case numero:
        fai qualcosa;
        break;
    case altro_numero:
        fai qualcosa;
        break;
}
```

```

1- import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         System.out.println("Inserisci un numero: ");
7         int numero=scanner.nextInt();
8         switch (numero){
9             case 0:
10                 System.out.println("Il numero è lo 0.");
11                 break;
12             case 1:
13                 System.out.println("Il numero è l'1.");
14                 break;
15             case 2:
16                 System.out.println("Il numero è il 2.");
17                 break;
18             case 3:
19                 System.out.println("Il numero è il 3.");
20                 break;
21             case 4:
22                 System.out.println("Il numero è il 4.");
23                 break;
24             case 5:
25                 System.out.println("Il numero è il 5.");
26                 break;
27         }
28     }
29 }
```

input

Inserisci un numero:  
5  
Il numero è il 5.

Molto importante è aggiungere alla fine di ogni case il break, altrimenti si eseguono tutte le altre condizioni.

## ESERCIZI

Prendi in input 1 numero e scrivi se è pari o dispari.  
 Prendi in input 1 numero e scrivi se è positivo o negativo.  
 Prendi in input 1 numero in un intervallo tra 1-5 e scrivi il numero scelto.  
 Prendi in input due numeri e scrivi quale è il maggiore.

# 4

## Recap

Spiegate le basi del linguaggio e dell'informatica prima di passare a cose un pò più complesse facciamo qualche esercizio di recap.

**Chiedi all'utente il suo nome e la sua età. Stampa una frase del tipo:**

**Ciao Marco, hai 20 anni.**

The screenshot shows a Java code editor with a file named Main.java. The code uses Scanner to read input from the user and prints it back. Below the editor is a terminal window showing the execution of the program and its output.

```
Main.java
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         System.out.println("Inserisci il tuo nome: ");
7         String nome=scanner.nextLine();
8         System.out.println("Inserisci la tua età: ");
9         int eta=scanner.nextInt();
10        System.out.println("Ciao "+nome+" hai: "+eta+" anni.");
11    }
12 }
13
```

```
Inserisci il tuo nome:
alessandro
Inserisci la tua età:
99
Ciao alessandro hai: 99 anni.
```

**Prendi due numeri interi in input. Stampa la somma, la differenza, il prodotto e la divisione.**

The screenshot shows a Java code editor with a file named Main.java. The code uses Scanner to read two integers from the user and then calculates their sum, difference, multiplication, and division. Below the editor is a terminal window showing the execution of the program and its output.

```
Main.java
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         int a;
7         int b;
8         System.out.println("Inserisci un numero: ");
9         a=scanner.nextInt();
10        System.out.println("Inserisci un numero: ");
11        b=scanner.nextInt();
12        int somma=a+b;
13        int differenza=a-b;
14        int moltiplicazione=a*b;
15        int divisione=a/b;
16        System.out.println(somma+" "+differenza+" "+moltiplicazione+" "+divisione);
17    }
18 }
19
```

```
Inserisci un numero:
2
Inserisci un numero:
3
5 -1 6 0
```

**Chiedi all'utente un numero intero e stampa se è pari o dispari.**

The screenshot shows a Java code editor with a file named Main.java. The code is as follows:

```
1- import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         System.out.println("Inserisci un numero: ");
7         int a=scanner.nextInt();
8         if(a%2==0){
9             System.out.println("Il numero è pari.");
10        }
11        else{
12            System.out.println("Il numero è dispari.");
13        }
14    }
15 }
16
```

Below the code editor, there is a terminal window titled "Input". It shows the following interaction:

```
Inserisci un numero:
4
Il numero è pari.
```

**Crea una mini calcolatrice.**

## JAVA PER PRINCIPIANTI

```
Main.java :  
1 import java.util.Scanner;  
2 public class Main  
3 {  
4     public static void main(String[] args) {  
5         Scanner scanner=new Scanner(System.in);  
6         System.out.println("Inserisci un numero: ");  
7         int a=scanner.nextInt();  
8         System.out.println("Inserisci un numero: ");  
9         int b=scanner.nextInt();  
10        System.out.println("Inserisci 1-somma,2-differenza,3-moltiplicazione,4-divisione");  
11        int risultato;  
12        int operazione=scanner.nextInt();  
13        switch (operazione){  
14            case 1:  
15                risultato=a+b;  
16                System.out.println(risultato);  
17                break;  
18            case 2:  
19                risultato=a-b;  
20                System.out.println(risultato);  
21                break;  
22            case 3:  
23                risultato=a*b;  
24                System.out.println(risultato);  
25                break;  
26            case 4:  
27                risultato=a/b;  
28                System.out.println(risultato);  
29                break;  
30        }  
31    }  
32}  
33
```

```
Inserisci un numero:  
1  
Inserisci un numero:  
2  
Inserisci 1-somma,2-differenza,3-moltiplicazione,4-divisione  
1  
3
```

# 5

## Iterazioni

In Java, così come in tutti i linguaggi di programmazione, possiamo servirci delle iterazioni, chiamate anche cicli, per ripetere uno o più blocchi di codice. Questa ripetizione può avvenire per un numero prestabilito di volte oppure finché una determinata condizione è vera. Le iterazioni sono fondamentali per rendere il codice più leggero, più efficiente e molto più facile da gestire. Inoltre, ci permettono di automatizzare operazioni ripetitive oppure di mantenerle attive finché l'utente lo desidera. Pensiamoci un attimo: se volessimo creare una calcolatrice, dopo aver concluso un'operazione, molto probabilmente ne vorremmo fare subito un'altra, senza riavviare tutto da capo e dalla parte del programmatore dovrebbe incollare infinitamente quel blocco di codice (impossibile). Ed è proprio grazie alle iterazioni che possiamo farlo, mantenendo attivo un menu che si ripete finché l'utente non decide di uscire. Esistono 3 tipi di cicli:

- For-loop: si utilizza quando si sanno in anticipo quante volte bisogna ripetere.

- While-loop: si utilizza per ripetere fino a quando una condizione è vera.
- Do-while-loop: è simile al while ma la differenza è che esegue il blocco all'interno del ciclo anche se la condizione è falsa (il controllo è alla fine).

Per uscire dal ciclo se c'è una determinata condizione per esempio nell'if si deve aggiungere la parola (**break;**).

```
for (int i = 1; i <= 10; i++) {
    if (i == 5) {
        break;
    }
    System.out.println("i = " + i);
}
```

## FOR-LOOP

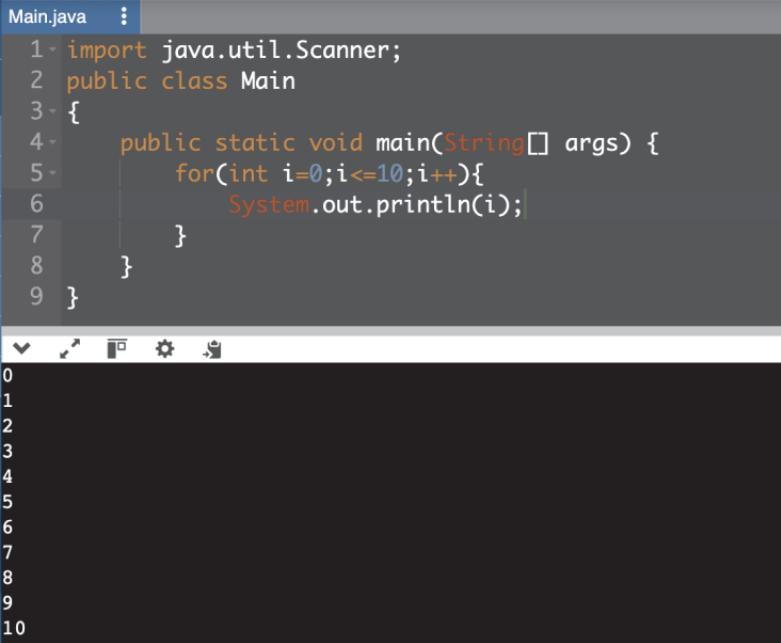
La struttura di base è la seguente:

```
for(int i=0;i<numero;i++){
    fai qualcosa;
}
```

Quindi quando si entra nel ciclo si dichiara un contatore (i), a cui viene attribuito un numero, che incrementa ( $i++ \rightarrow i=i+1$ ). Ogni volta viene controllata la condizione, e se è falsa esce dal ciclo.

Un esempio con il ciclo for è stampare tutti i numeri tra 0 e 10.

Se non ci fosse stato il ciclo sarebbe stato relativamente lungo e faticoso.



```
Main.java  :
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         for(int i=0;i<=10;i++){
6             System.out.println(i);
7         }
8     }
9 }
```

```
0
1
2
3
4
5
6
7
8
9
10
```

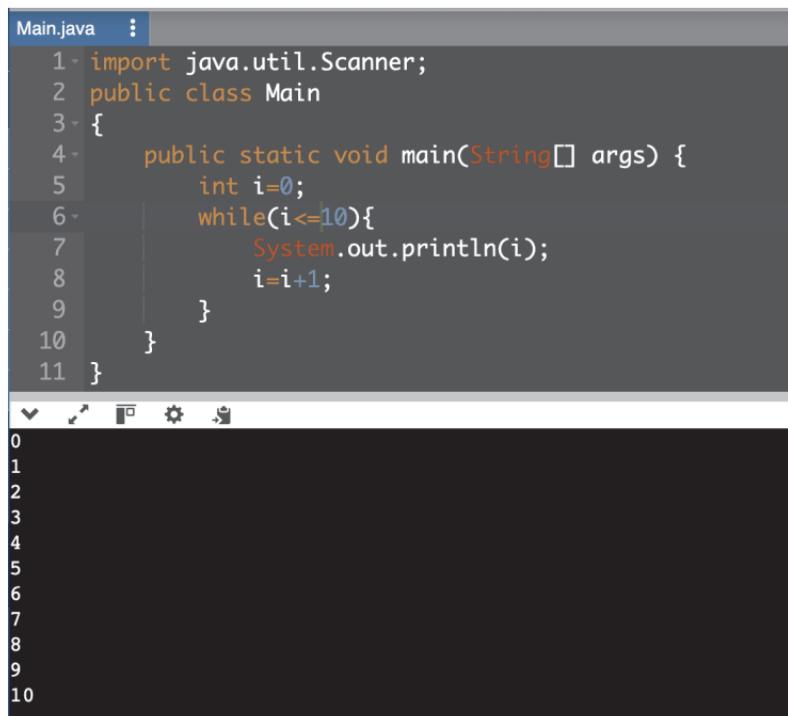
## WHILE LOOP

La struttura di base è la seguente:

```
while(condizione){
    fai qualcosa;
}
```

La struttura del ciclo while è relativamente molto più semplice da ricordare rispetto a quella del for. Nel while il ciclo viene ripetuto il (fai qualcosa) fino a quando la condizione diventa

falsa. Il contatore nel while può essere aggiunto manualmente all'interno con il suo relativo incremento oppure nel caso della condizione sempre vera non è presente. Un esempio del while è stampare tutti i numeri minori di 10 a partire da 0.



```
Main.java ::

1- import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         int i=0;
6         while(i<=10){
7             System.out.println(i);
8             i=i+1;
9         }
10    }
11 }
```

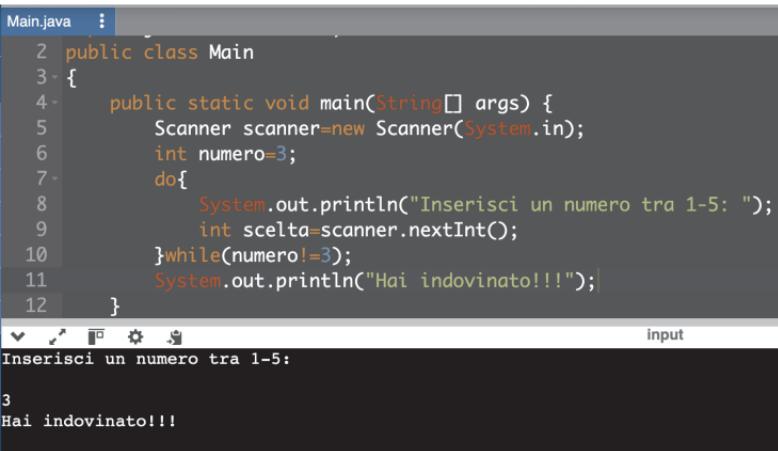
```
0
1
2
3
4
5
6
7
8
9
10
```

## DO-WHILE LOOP

La struttura di base è la seguente:

```
do{
    fai qualcosa;
}while(condizione);
```

Il ciclo do-while in Java esegue il blocco di codice almeno una volta, indipendentemente dalla condizione. La condizione viene verificata dopo l'esecuzione del codice, a differenza del ciclo while dove la condizione viene controllata prima. Questo lo rende utile quando si desidera che un'azione venga eseguita almeno una volta, come nel caso di richieste di input da parte dell'utente. Se la condizione è falsa, il ciclo termina dopo la prima esecuzione, ma se è vera, continua a ripetersi. Bisogna notare che bisogna aggiungere alla parola do all'inizio e il ; alla fine dopo la condizione. Un esempio con il ciclo do-while è indovinare il numero tra 1-5.



```
Main.java ::

1  public class Main
2  {
3      public static void main(String[] args) {
4          Scanner scanner=new Scanner(System.in);
5          int numero=3;
6          do{
7              System.out.println("Inserisci un numero tra 1-5: ");
8              int scelta=scanner.nextInt();
9          }while(numero!=3);
10         System.out.println("Hai indovinato!!!");
11     }
12 }
```

Inserisci un numero tra 1-5: input

3  
Hai indovinato!!!

## ESERCIZI

- Stampa numeri 1-10 (for)
- Stampa numeri 1-10 (while)
- Somma i primi 5 numeri (while)
- Chiedi all'utente un numero finché non inserisce 0 (do-while)

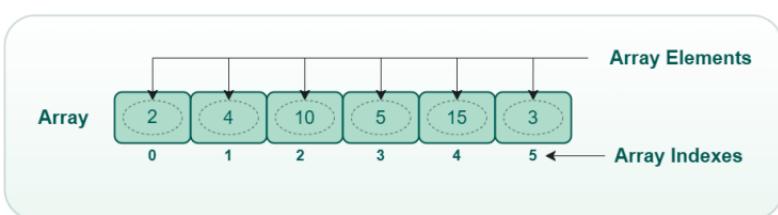
Stampa numeri solo pari fino a 20 (for)

Stampa numeri solo pari fino a 20 (while)

# 6

## Array

L'array è una delle strutture dati, utilizzata per organizzare, memorizzare e gestire gruppi di dati in modo ordinato. Esistono molti tipi di strutture dati ma noi ci limitiamo all'array (il più conosciuto e il più semplice, ma molto molto utile). L'array serve per esempio per memorizzare e gestire i voti di uno studente, infatti si possono aggiungere, modificare o stampare. La visualizzazione di un'array nella nostra testa potrebbe essere la seguente:



Come vediamo è un insieme di quadratini che formano una riga, all'interno di ogni quadratino troviamo un dato (in questo caso si vogliono memorizzare dei numeri). Ogni quadratino ha una

posizione specifica, il primo ha posizione 0 (molto importante perchè spesso chi è alle prime parti la confonde con la prima posizione) e poi si aumenta di 1. Per ssere precisi il contenuto di ogni quadratino si chiama elemento dell'array, mentre la posizione si chiama indice. Per esercitarti potresti disegnarti sulla carta un array e a voce nominare gli indici (le posizioni) degli elmenti anche in disordine per costruirti una logica nella mente. Dopo questa introduzione teorica passiamo alla pratica...

### DICHIARAZIONE ARRAY

Un array di numeri si dichiara così:

```
int array[] = new int[numero di elementi];
```

int indica la tipologia di dato mentre array è il nome dell'array e numero di elementi indica quanti elementi deve contenere l'array (la sua dimensione). Per esempio se noi volessimo dichiarare un array di caratteri con il nome alfabeto e con 21 lettere:

```
char alfabeto[] = new char[21];
```

### POPOLAMENTO ARRAY

Dopo averlo dichiarato si può o manualmente aggiungere gli elementi o si possono far inserire dall'utente. Nel gergo tecnico aggiungere elementi ad un array si chiama (popolare gli array). Aggiungerli manualmente si puo fare in 2 modi diversi: o aggiungerli direttamente nella dichiarazione o di volta in volta in ogni posizione. Il primo metodo è il seguente:

```
1 int[] numeri = {10, 20, 30, 40, 50};
```

In questo modo l'altra parte dell'inizializzazione non c'è perché la dimensione la non è la variabile ma il computer conta direttamente gli elementi.

Il secondo metodo è il seguente:

```
1 int[] numeri = new int[5];
2     numeri[0] = 10;
3     numeri[1] = 20;
4     numeri[2] = 30;
5     numeri[3] = 40;
6     numeri[4] = 50;
```

Qui dichiariamo l'array normalmente e la dimensione (numero 5) e poi manualmente in ogni posizione si aggiunge un numero. Questo metodo è la versione base del metodo che faremo adesso con il ciclo for. In questo esempio dichiariamo un array di 5 elementi e lo facciamo popolare all'utente.

```

1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner= new Scanner(System.in);
6         System.out.println("Popolamento array: ");
7         int array[]=new int[5];
8         for(int i=0;i<5;i++){
9             System.out.println("Inserisci un numero: ");
10            array[i]=scanner.nextInt();
11        }
12    }
13 }

```

Popolamento array:  
Inserisci un numero:  
1  
Inserisci un numero:  
2  
Inserisci un numero:  
3  
Inserisci un numero:  
4  
Inserisci un numero:  
5

Come possiamo vedere abbiamo fatto il set-up con lo scanner, abbiamo fatto una piccola introduzione abbiamo dichiarato l'array con il nome array, con 5 elementi e poi con l'aiuto del ciclo for abbiamo popolato l'array. Nel ciclo for il contatore l'abbiamo inizializzato a 0 (prima posizione dell'array) e ogni volta che ricominciava il ciclo aumentava di 1 fino a che diventava 5 ed usciva dal ciclo. Il ruolo del contatore è molto importante perchè viene utilizzato come assegna posizione (array[i]=scanner.nextInt());, se ci ricordiamo sopra abbiamo fatto un esempio simile però noi dovevamo a mano aggiungere gli elementi, qui fa tutto in automatico e il numero lo sceglie l'utente. Se avessimo messo  $\leq 5$  la condizione avrebbe aggiunto l'elemento anche alla posizione 5 che non esiste e sarebbe crashato il programma. Adesso che abbiamo aggiunto 5 numeri al nostro array, vogliamo

vedere il contenuto e se ha memorizzato correttamente.

### STAMPARE ARRAY

Questa operazione è molto semplice, bisogna utilizzare ancora un altro ciclo for e stampare ogni elemento in ciascuna posizione.

```
12 |     System.out.println("Adesso ti stampo l'array: ");
13 |     for(int i=0;i<5;i++){
14 |         System.out.println(array[i]);
15 |     }
16 | }
17 }
```

**Adesso ti stampo l'array:**

1  
2  
3  
4  
5

Abbiamo utilizzato la stessa logica di prima (array[i] indica l'elemento nella posizione i) e la posizione i è data dal ciclo for.

### ESERCIZI

Dichiara array di 10 voti e falli inserire dall'utente e poi stampali.

Dichiara array registro di 10 temperature e falli

inserire dall'utente e poi stampale.  
Qusto potrebbe essere un esercizio un pò più  
articolato: Dichiara array di 10 voti e falli  
inserire dall'utente e poi stampali, stampa la media  
e di ogni voto scrivi se è sufficiente o  
insufficiente (Cerca di fare di testa tuo, nel caso  
tu non riesca puoi prendere spunto dalla soluzione in  
fondo al libro).

# 7

## Matrici

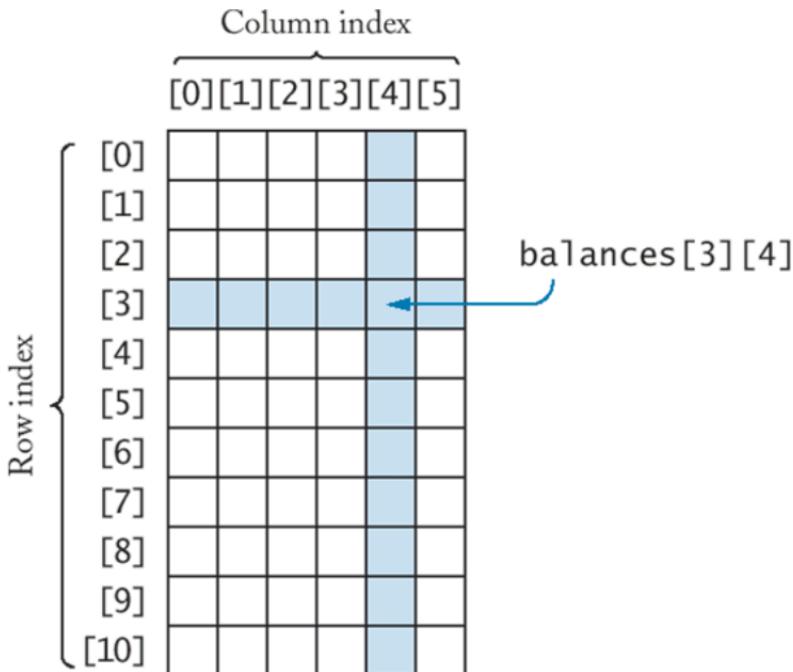
Le matrici o array-multidimensionali sono delle vere e proprie tabelle array. Queste matrici sono fondamentali da conoscere per saper svolgere il gioco tris (matrice 3x3). Queste sono molto simili all'array sia nella dichiarazione, sia nel popolamento, sia nella stampa.

### DICHIARAZIONE MATRICI

Una matrice di interi si dichiara così:

```
int matrix[][]=new int[dimensione riga][dimensione  
colonna];
```

Come per gli array, anche nelle matrici in Java il tipo di dato (int) indica che tipo di dati conterranno, mentre matrix è il nome della matrice. La dimensione si scrive come [righe][colonne]. Una matrice 3x3 ha quindi 3 righe e 3 colonne. Per ricordarlo: ogni riga è come un singolo array, quindi la prima dimensione rappresenta le righe, e la seconda le colonne.



Stessa cosa se si vuole dichiarare una matrice di caratteri formata da 3 righe e 3 colonne:

```
char matrix[][]=new char[3][3];
```

## POPOLAMENTO MATRICI

Come gli array dobbiamo riferirci ad un elemento con la sua posizione data in questo caso dalla riga e dalla colonna, (`matrix[riga][colonna]`). Abbiamo detto che la posizione di un elemento è data da 2 posizioni (riga e colonna) quindi per popolarla utilizzeremo 2 cicli `for` con 2 incrementi diversi.

```
for(int i=0;i<riga;i++){
    for(int j=0;j<colonna;j++){
        matrix[i][j]=scanner.nextInt();
    }
}
```

Quando utilizziamo due cicli for uno dentro l'altro, si parla di cicli annidati. Questi vengono usati spesso per popolare o leggere matrici, cioè array bidimensionali. Esistono 2 diversi tipi di popolamenti:orizzontale (riga per riga) o verticale (colonna per colonna). Nel caso del popolamento orizzontale, cioè riga per riga, il ciclo esterno scorre le righe mentre quello interno scorre le colonne. Entrando nel ciclo esterno, si esegue completamente quello interno, riempiendo così una riga per volta da sinistra a destra. Dopo aver compilato la prima riga, si torna al ciclo esterno che passa alla riga successiva e ripete il processo fino all'ultima riga. Questo metodo permette di riempire la matrice una riga alla volta, in ordine crescente. Ragionamento analogo per popolamento verticale.

Esempio di ordinamento orizzontale:

```

1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         int matrix[][]=new int[3][3];
7         for(int i=0;i<3;i++){
8             for(int j=0;j<3;j++){
9                 System.out.println("Inserisci un numero: ");
10                matrix[i][j]=scanner.nextInt();
11            }
12        }
13    }
14 }
```

Inserisci un numero:  
1  
Inserisci un numero:  
2  
Inserisci un numero:  
3  
Inserisci un numero:  
4  
Inserisci un numero:  
5  
Inserisci un numero:  
6  
Inserisci un numero:  
7  
Inserisci un numero:  
8  
Inserisci un numero:  
9

Guardando con attenzione è molto simile al popolamento di un array, leggermente più complicato. Vale sempre l'attribuzione `matrix[riga][colonna]` alla posizione specifica della matrice. Dopo aver inserito riga per riga tutti gli elementi dell'array adesso proviamo a stamparli.

### **STAMPARE MATRICI**

Lo svolgimento è identico a quello dell'array tranne il ciclo for che deve essere annidato per le colonne e le righe (bidimensionale).

```
System.out.println("Adesso ti stampo la matrice: ");
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            System.out.print(matrix[i][j]);
        }
        System.out.println("");
    }
```

```
Adesso ti stampo la matrice:
123
456
789
```

Come possiamo vedere l'ordinamento fatto è stato orizzontale, (inserendo tutti i numeri da 1-9 abbiamo completato tutte le righe).

Per stampare quindi abbiamo ancora utilizzato il ciclo anidato e all'interno del ciclo interno non abbiamo utilizzato `println` (stessa riga) alla fine della riga quindi fuori dal ciclo interno lo abbiamo utilizzato per passare alla riga inferiore.

## ESERCIZI

```
Dichiara matrice(3x3), falla riempire dall'utente e
stampala.
Dichiara matrice(3x3) e simula il gioco tris
(riempiendo te gli spazi con X-O tramite input).
Dichiara 2 matrici 3x3 e fai la moltiplicazione
membro membro (alla stessa posizione).
```

## Procedure

In Java, una procedura (chiamata tecnicamente *metodo void*) è un insieme di istruzioni racchiuse in un blocco di codice, che svolge una determinata operazione ma non restituisce alcun valore, (a differenza delle funzioni). Le procedure sono molto utili e smart, esse permettono di strutturare il codice in blocchi più semplici, rendono il codice più ordinato e quindi molto più comprensibile e permettono il riutilizzo di blocchi di codice. Se noi ci pensiamo una procedura si trova all'interno della public class main.



The screenshot shows a dark-themed code editor window titled "Main.java". The code is a simple Java program:

```
1  -----
2
3  |           | Online Java Compiler.
4  |           | Code, Compile, Run and Debug java program online.
5  Write your code in this editor and press "Run" button to execute it.
6
7 -----
8
9 public class Main
10 {
11     public static void main(String[] args) {
12     }
13 }
```

The code editor has a status bar at the bottom with the text "Java 11" and "File Edit View Insert Run Help".

## PROCEDURE

Dove scriviamo il codice attualmente è una procedura chiamata main (public static void main(String[] args){}). La differenza principale tra una procedura e una funzione è come abbiamo già detto che una procedura non ritorna niente ma la possiamo riconoscere dalla parola **void**, senza di quella sarebbe una funzione.

Quindi dopo questa introduzione teorica iniziamo a capire come scrivere la procedura.

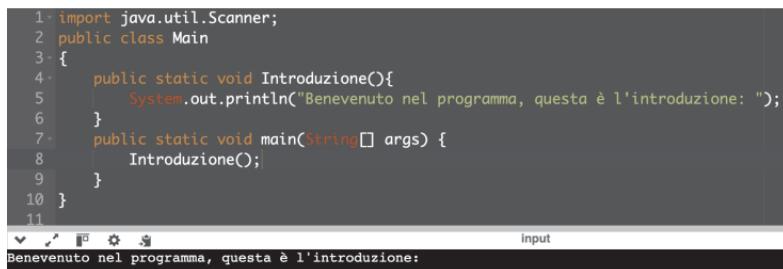
Per prima cosa bisogna ricordarsi di scriverla all'interno della class main ma al di fuori della procedura main, poi si scrive così:

```
public static void nome_procedura(eventuali  
parametri){  
}
```

Dopo averla dichiarata bisogna chiamarla all'interno della procedura main.

```
nome_procedura(eventuali parametri);
```

Un esempio di procedura sarebbe per esempio stampare l'introduzione di un programma:



The screenshot shows a Java code editor with the following code:

```
1- import java.util.Scanner;  
2 public class Main  
3 {  
4     public static void Introduzione(){  
5         System.out.println("Benevenuto nel programma, questa è l'introduzione: ");  
6     }  
7     public static void main(String[] args) {  
8         Introduzione();  
9     }  
10 }  
11
```

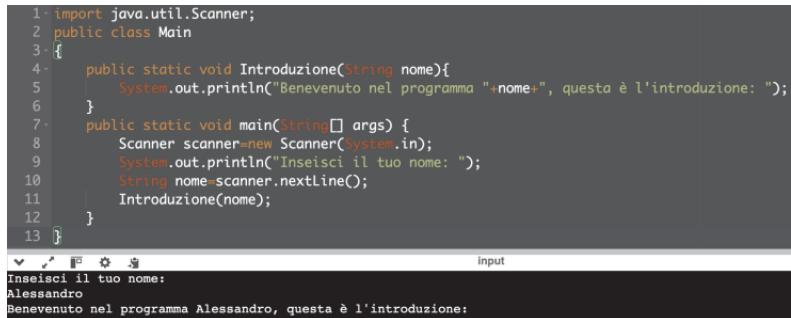
Below the code, there is a terminal window showing the output:

```
Benevenuto nel programma, questa è l'introduzione: input
```

Per farlo un pò più complicato si potrebbero inserire dei parametri. Il parametro si deve dichiarare negli eventuali parametri dentro le parentesi tonde.

```
public static void Introduzione(String nome){
```

In questo esempio vogliamo prendere in input un nome e stampare benvenuto nel programma + nome + questa è l'introduzione.



```

1- import java.util.Scanner;
2- public class Main
3- {
4-     public static void Introduzione(String nome){
5-         System.out.println("Benvenuto nel programma "+nome+", questa è l'introduzione.");
6-     }
7-     public static void main(String[] args) {
8-         Scanner scanner=new Scanner(System.in);
9-         System.out.println("Inserisci il tuo nome: ");
10-        String nome=scanner.nextLine();
11-        Introduzione(nome);
12-    }
13- }
```

The screenshot shows a Java code editor with the above code. Below it, a terminal window titled 'input' shows the output of running the program. The user types 'Inserisci il tuo nome:' followed by 'Alessandro'. The program then prints 'Benvenuto nel programma Alessandro, questa è l'introduzione:'.

Bisogna ricordarsi anche di inserire il parametro dentro la parentesi della chiamata.

# Introduzione(nome);

## ESERCIZI

## PROCEDURE

Scrivi una procedura che prende in input un nome e stampa ciao + nome e chiamala nel main.  
Dichiara una procedura somma e stampa la somma di 2 numeri presi in input.

# 9

## Funzioni

A differenza delle procedure le funzioni possono ritornare qualcosa e appunto non presentano la parola void ma presentano la parola del tipo di variabile che restituiscono. La dichiarazione è uguale alle procedure senza il void e si trovano sempre al di fuori della procedura main ma all'interno della classe main:

```
public static (tipo variabile che restituisce)
nomefunzione(eventuali parametri){

}
```

Dopo averla dichiarata bisogna chiamarla all'interno della procedura main:

```
nomefunzione(eventuali parametri);
```

Abbiamo detto quindi che le funzioni possono restituire un valore. Questo significa che, una volta eseguito il codice all'interno della funzione, il programma riceverà un risultato (un numero,

## FUNZIONI

un testo, ecc.) che può essere usato in altre parti del programma, ad esempio per stamparlo o per fare altri calcoli. Un esempio di funzione potrebbe essere dati 2 dati presi in input stampare la somma con una funzione.

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static int Somma(int numero1,int numero2){
5         int somma=numero1+numero2;
6         return somma;
7     }
8     public static void main(String[] args) {
9         Scanner scanner=new Scanner(System.in);
10        int numero1;
11        int numero2;
12        System.out.println("Inserisci un numero: ");
13        numero1=scanner.nextInt();
14        System.out.println("Inserisci un numero: ");
15        numero2=scanner.nextInt();
16        System.out.println(Somma(numero1,numero2));
17    }
18 }
19 }
```

Inserisci un numero:

2

Inserisci un numero:

3

5

## ESERCIZI

Scrivi una funzione che prende due numeri e restituisce il maggiore.  
Fai una piccola calcolatrice dove racchiudi all'interno (ciclo while) la presa in input di 2 numeri e fai scegliere l'operazione (addizione o sottrazione) che deve trovarsi al di fuori del main.

# 10

## Esercizio recap (Prenota posti cinema)

Dopo aver scoperto e assimilato tutti questi argomenti è ora di metterli in pratica per costruire un programma che prenoti i posti di un cinema. Dobbiamo immaginare che la piantina del cinema sia un array che contiene 10 posti e quando un utente vuole prenotare il posto dica se sia occupato o no. Per controllare se il posto è occupato prima di tutto inizializziamo tutti i posti a 0, quando una persona sceglie il posto lo setta ad 1 e se lo risceglie dice che è occupato. Per prendere in input la scelta facciamo una funzione che controlli se il numero inserito sia compreso tra 0 - 9, se è vero ritorna il posto. Nella funzione main dichiariamo un array di interi di 10 posti e poi settiamo tutti i numeri a 0. Poi facciamo il do-while dove chiamiamo la funzione input e la assegnamo a una variabile int posto, con un if controlliamo se è occupato o è libero, con un if controlliamo se è occupato o è libero. Dopo ogni occupazione aggiorniamo la somma che deve arrivare fino a 10, arrivati a 10 si esce dal ciclo e si scrive che tutti i posti sono occupati.

## JAVA PER PRINCIPIANTI

```
1- import java.util.Scanner;
2- public class Main
3 {
4-     public static int Input(){
5-         Scanner scanner=new Scanner(System.in);
6-         int posto;
7-         do{
8-             posto=scanner.nextInt();
9-             if(posto<0 || posto>10){
10-                 System.out.println("ERRORE ERRORE ERRORE");
11-             }
12-         }while(posto<0 || posto>10);
13-         return posto;
14-     }
15-     public static void main(String[] args) {
16-         int array[]=new int[10];
17-         int somma=0;
18-         for(int i=0;i<10;i++){
19-             array[i]=0;
20-         }
21-         do{
22-             System.out.println("Benvenuto nel cinema. I posti sono 10 e vanno dal 0-9, inserisci il posto che vuoi prenotare:");
23-             int posto=Input();
24-             if(array[posto]==0){
25-                 System.out.println("Il posto è libero benvenuto!!!");
26-                 array[posto]=1;
27-                 somma=somma+1;
28-             }
29-             else{
30-                 System.out.println("Il posto è occupato.");
31-             }
32-         }while(somma<10);
33-         System.out.println("I posti sono tutti occupati.");
34-     }
35- }
```

Sul terminale dovrebbe apparire così:

```
Benvenuto nel cinema. I posti sono 10 e vanno dal 0-9, inserisci il posto che vuoi prenotare:
1
Il posto è libero benvenuto!!!
Benvenuto nel cinema. I posti sono 10 e vanno dal 0-9, inserisci il posto che vuoi prenotare:
2
Il posto è libero benvenuto!!!
Benvenuto nel cinema. I posti sono 10 e vanno dal 0-9, inserisci il posto che vuoi prenotare:
3
Il posto è libero benvenuto!!!
Benvenuto nel cinema. I posti sono 10 e vanno dal 0-9, inserisci il posto che vuoi prenotare:
4
Il posto è libero benvenuto!!!
Benvenuto nel cinema. I posti sono 10 e vanno dal 0-9, inserisci il posto che vuoi prenotare:
5
Il posto è libero benvenuto!!!
Benvenuto nel cinema. I posti sono 10 e vanno dal 0-9, inserisci il posto che vuoi prenotare:
6
Il posto è libero benvenuto!!!
Benvenuto nel cinema. I posti sono 10 e vanno dal 0-9, inserisci il posto che vuoi prenotare:
7
Il posto è libero benvenuto!!!
Benvenuto nel cinema. I posti sono 10 e vanno dal 0-9, inserisci il posto che vuoi prenotare:
8
Il posto è libero benvenuto!!!
Benvenuto nel cinema. I posti sono 10 e vanno dal 0-9, inserisci il posto che vuoi prenotare:
9
Il posto è libero benvenuto!!!
Benvenuto nel cinema. I posti sono 10 e vanno dal 0-9, inserisci il posto che vuoi prenotare:
0
Il posto è libero benvenuto!!!
I posti sono tutti occupati.
```

# 11

## Soluzione esercizi

### CAPITOLO 1

Stampare il tuo nome.

```
1 public class Main
2 {
3     public static void main(String[] args) {
4         System.out.println("Alessandro");
5     }
6 }
7
```

Alessandro

Stampare il tuo nome e a capo il tuo cognome.

```
1 public class Main
2 {
3     public static void main(String[] args) {
4         System.out.println("Alessandro");
5         System.out.println("Faletti");
6     }
7 }
8
```

```
Alessandro
Faletti
```

Stampare il tuo nome, sulla stessa linea il tuo cognome e a capo la tua età.

```
1 public class Main
2 {
3     public static void main(String[] args) {
4         System.out.print("Alessandro ");
5         System.out.println("Faletti");
6         System.out.println("99");
7     }
8 }
```

# Alessandro Faletti

## 99

### CAPITOLO 2

Prendi in input 1 numero int e stampalo.

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         System.out.print("Inserisci un numero: ");
7         int numero=scanner.nextInt();
8         System.out.print(numero);
9     }
10 }
11
```

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

Prendi in input 1 numero int e stampalo con la concatenazione,  
hai scelto il numero.

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         System.out.print("Inserisci un numero: ");
7         int numero=scanner.nextInt();
8         System.out.print("Hai scelto il numero "+numero);
9     }
10 }
```

```
Inserisci un numero: 3
Hai scelto il numero 3
```

Prendi in input un nome e un cognome e stampali con la concatenazione.

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         System.out.print("Inserisci un nome: ");
7         String nome=scanner.nextLine();
8         System.out.print("Inserisci un cognome: ");
9         String cognome=scanner.nextLine();
10        System.out.print(nome+" "+cognome);
11    }
12 }
```

Inserisci un nome: Alessandro  
Inserisci un cognome: Faletti  
Alessandro Faletti

## CAPITOLO 3

Prendi in input 1 numero e scrivi se è pari o dispari.

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         System.out.println("Inserisci un numero: ");
7         int numero=scanner.nextInt();
8         if(numero%2==0){
9             System.out.println("Pari");
10        }
11        else{
12            System.out.println("Dispari");
13        }
14    }
15 }
```

Inserisci un numero:  
3  
Dispari

Prendi in input 1 numero e scrivi se è positivo o negativo.

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         System.out.println("Inserisci un numero: ");
7         int numero=scanner.nextInt();
8         if(numero>=0){
9             System.out.println("Positivo");
10        }
11        else{
12            System.out.println("Negativo");
13        }
14    }
15 }
```

Inserisci un numero:

4

Positivo

Prendi in input 1 numero in un intervallo tra 1-5 e scrivi il numero scelto.

## SOLUZIONE ESERCIZI

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         System.out.println("Inserisci un numero compreso tra 1-5: ");
7         int numero=scanner.nextInt();
8         switch(numero){
9             case 1:
10                 System.out.println("Il numero 1");
11                 break;
12             case 2:
13                 System.out.println("Il numero 2");
14                 break;
15             case 3:
16                 System.out.println("Il numero 3");
17                 break;
18             case 4:
19                 System.out.println("Il numero 4");
20                 break;
21             case 5:
22                 System.out.println("Il numero 5");
23                 break;
24         }
25     }
26 }
```

```
Inserisci un numero compreso tra 1-5:
3
Il numero 3
```

Prendi in input due numeri e scrivi quale è il maggiore.

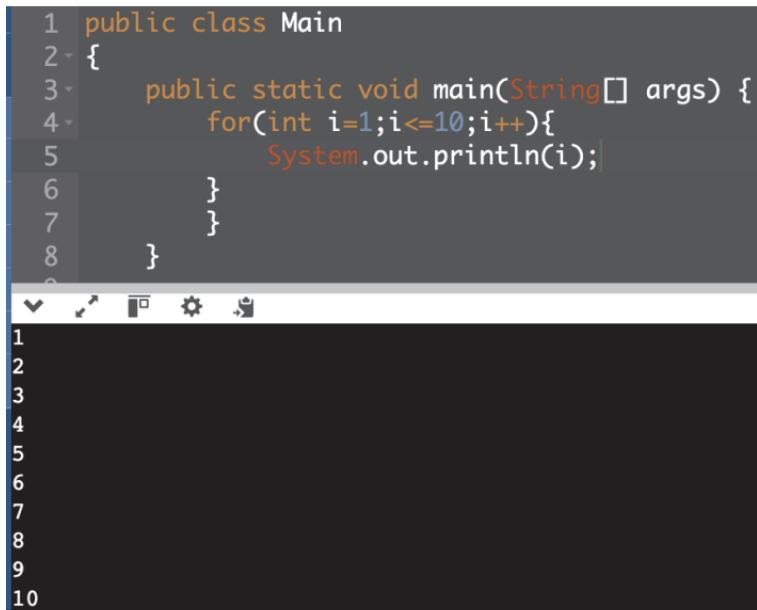
```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         System.out.println("Inserisci un numero: ");
7         int numero1=scanner.nextInt();
8         System.out.println("Inserisci un numero: ");
9         int numero2=scanner.nextInt();
10        if(numero1>numero2){
11            System.out.println(numero1+" è il maggiore.");
12        }
13        else{
14            System.out.println(numero2+" è il maggiore.");
15        }
16    }
17 }
18 }
```

```
Inserisci un numero:
2
Inserisci un numero:
1
2 è il maggiore.
```

## CAPITOLO 5

Stampa numeri 1-10 (for)

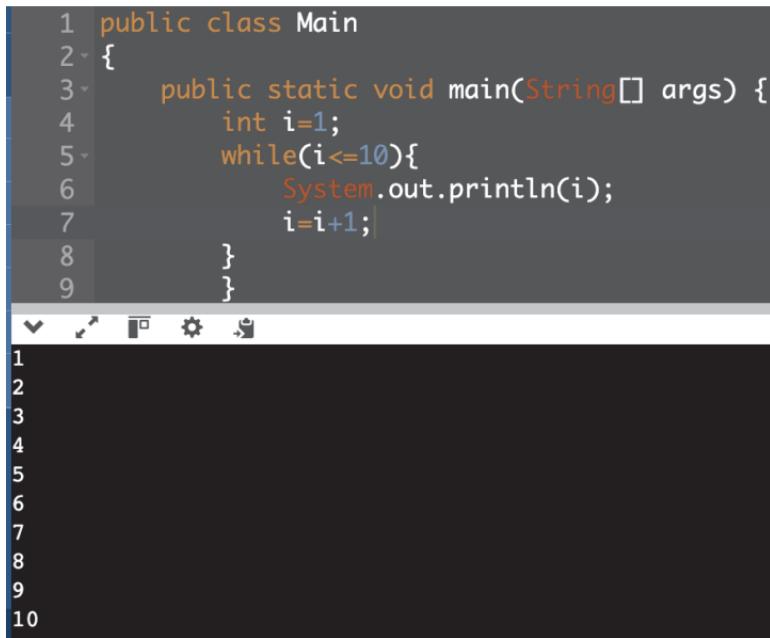
## SOLUZIONE ESERCIZI



```
1 public class Main
2 {
3     public static void main(String[] args) {
4         for(int i=1;i<=10;i++){
5             System.out.println(i);
6         }
7     }
8 }
```

```
1
2
3
4
5
6
7
8
9
10
```

Stampa numeri 1-10 (while)



A screenshot of a Java code editor showing a simple program. The code defines a class named Main with a main method. Inside the main method, there is a while loop that prints integers from 1 to 10 to the console. The code is as follows:

```
1 public class Main
2 {
3     public static void main(String[] args) {
4         int i=1;
5         while(i<=10){
6             System.out.println(i);
7             i=i+1;
8         }
9     }
10 }
```

The output window below the editor shows the numbers 1 through 10 printed sequentially.

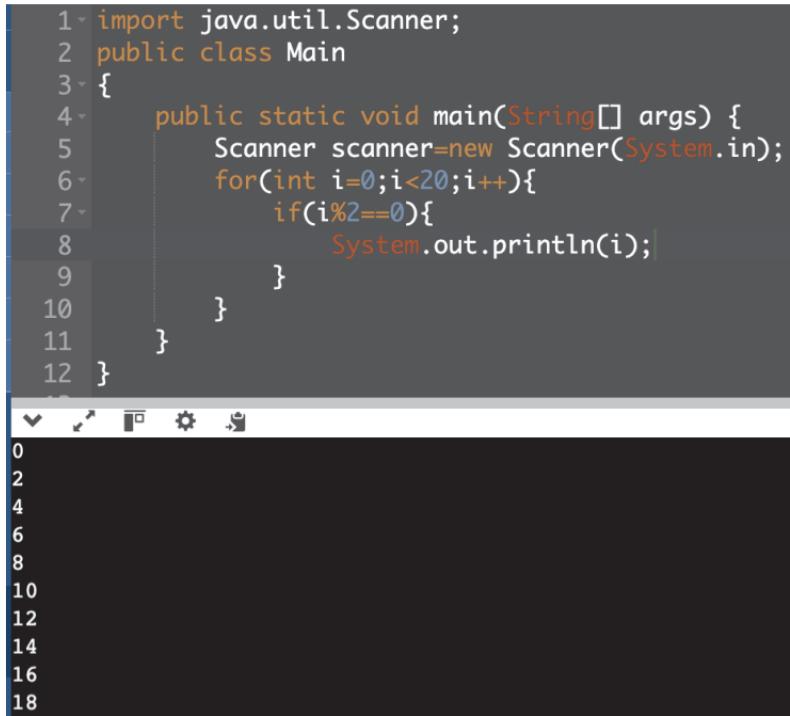
Chiedi all'utente un numero finchè non inserisce 0 (do-while).

## SOLUZIONE ESERCIZI

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         int n;
7         do{
8             System.out.println("Inserisci un numero: ");
9             n=scanner.nextInt();
10        }while(n!=0);
11        System.out.println("Il numero è 0.");
12    }
13 }
```

Inserisci un numero:  
3  
Inserisci un numero:  
1  
Inserisci un numero:  
0  
Il numero è 0.

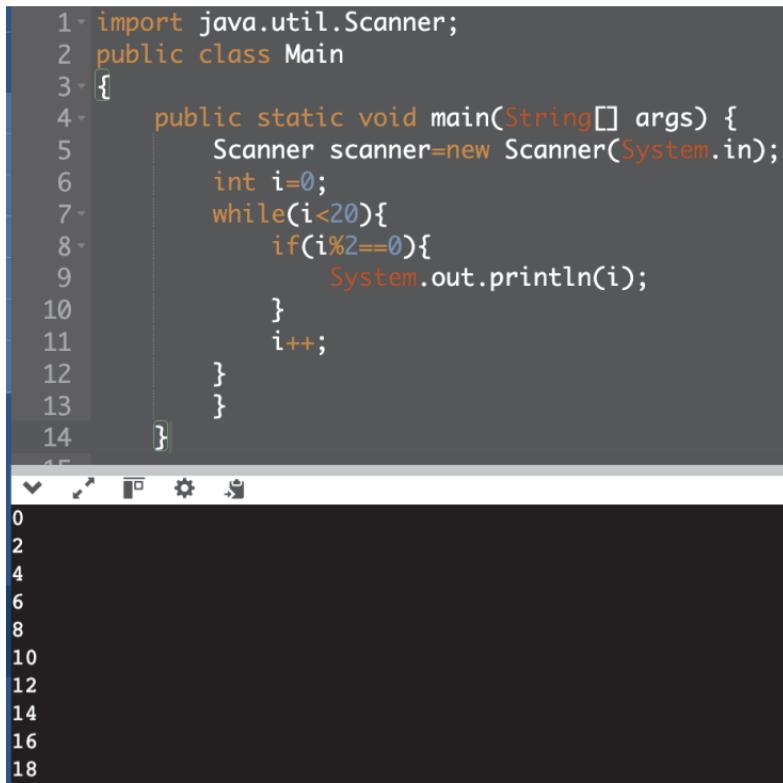
Stampa numeri solo pari fino a 20 (for).



```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         for(int i=0;i<20;i++){
7             if(i%2==0){
8                 System.out.println(i);
9             }
10        }
11    }
12 }
```

0  
2  
4  
6  
8  
10  
12  
14  
16  
18

Stampa numeri solo pari fino a 20 (while).



```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         int i=0;
7         while(i<20){
8             if(i%2==0){
9                 System.out.println(i);
10            }
11            i++;
12        }
13    }
14 }
```

The screenshot shows a Java code editor with the following code:

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         int i=0;
7         while(i<20){
8             if(i%2==0){
9                 System.out.println(i);
10            }
11            i++;
12        }
13    }
14 }
```

Below the code editor is a terminal window showing the output of the program:

```
0
2
4
6
8
10
12
14
16
18
```

## CAPITOLO 6

Dichiara array di 10 voti e falli inserire dall'utente e poi stampali.

## JAVA PER PRINCIPIANTI

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         int array[]=new int[10];
7         for(int i=0;i<10;i++){
8             System.out.println("Inserisci un voto: ");
9             array[i]=scanner.nextInt();
10    }
11    System.out.println("Adesso te li stampo: ");
12    for(int i=0;i<10;i++){
13        System.out.println(array[i]);
14    }
15 }
16 }
```

```
Inserisci un voto:  
3  
Inserisci un voto:  
4  
Inserisci un voto:  
5  
Inserisci un voto:  
4  
Inserisci un voto:  
3  
Inserisci un voto:  
2  
Inserisci un voto:  
1  
Inserisci un voto:  
2  
Inserisci un voto:  
3  
Inserisci un voto:  
4  
Adesso te li stampo:  
3  
4  
5  
4  
3  
2  
1  
2  
3  
4
```

Dichiara array registro di 10 temperature e falli inserire dall'utente e poi stampale.

## JAVA PER PRINCIPIANTI

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         int array[]=new int[10];
7         for(int i=0;i<10;i++){
8             System.out.println("Inserisci una temperatura: ");
9             array[i]=scanner.nextInt();
10        }
11        System.out.println("Adesso te le stampo: ");
12        for(int i=0;i<10;i++){
13            System.out.println(array[i]);
14        }
15    }
16 }
17 }
```

```
Inserisci una temperatura:  
20  
Inserisci una temperatura:  
20  
Inserisci una temperatura:  
23  
Inserisci una temperatura:  
25  
Inserisci una temperatura:  
26  
Inserisci una temperatura:  
28  
Inserisci una temperatura:  
30  
Inserisci una temperatura:  
21  
Inserisci una temperatura:  
10  
Inserisci una temperatura:  
2  
Adesso te le stampo:  
20  
20  
23  
25  
26  
28  
30  
21  
10  
2
```

Dichiara array di 4 voti e falli inserire dall'utente e poi stampali, stampa la media e di ogni voto scrivi se è sufficiente o insufficiente

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         int array[]=new int[4];
7         for(int i=0;i<4;i++){
8             System.out.println("Inserisci un voto: ");
9             array[i]=scanner.nextInt();
10            if(array[i]>=6){
11                System.out.println("Sufficiente.");
12            }
13            else{
14                System.out.println("Insufficiente.");
15            }
16        }
17        System.out.println("Adesso te li stampo: ");
18        for(int i=0;i<4;i++){
19            System.out.println(array[i]);
20        }
21        int media;
22        int somma=0;
23        for(int i=0;i<4;i++){
24            somma=somma+array[i];
25        }
26        media=somma/4;
27        System.out.println("Questa è la media: "+media);
28    }
29 }
```

Inserisci un voto:

4

Insufficiente.

Inserisci un voto:

4

Insufficiente.

Inserisci un voto:

2

Insufficiente.

Inserisci un voto:

10

Sufficiente.

Adesso te li stampo:

4

4

2

10

Questa è la media: 5

## CAPITOLO 7

Dichiara matrice(3x3), falla riempire dall'utente e stampala.

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         int matrix[][]=new int[3][3];
7         for(int i=0;i<3;i++){
8             for(int j=0;j<3;j++){
9                 System.out.println("Inserisci un numero: ");
10                matrix[i][j]=scanner.nextInt();
11            }
12        }
13        for(int i=0;i<3;i++){
14            for(int j=0;j<3;j++){
15                System.out.print(matrix[i][j]);
16            }
17            System.out.println("");
18        }
19    }
20 }
21 }
```

```
Inserisci un numero:  
1  
Inserisci un numero:  
2  
Inserisci un numero:  
3  
Inserisci un numero:  
1  
Inserisci un numero:  
2  
Inserisci un numero:  
3  
Inserisci un numero:  
1  
Inserisci un numero:  
2  
Inserisci un numero:  
3  
123  
123  
123
```

Dichiara matrice(3x3) e simula il gioco tris (riempiendo te gli spazi con X-O tramite input).

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         char matrix[][]=new char[3][3];
7         for(int i=0;i<3;i++){
8             for(int j=0;j<3;j++){
9                 System.out.println("Inserisci o X o O: ");
10                matrix[i][j]=scanner.next().charAt(0);
11            }
12        }
13        for(int i=0;i<3;i++){
14            for(int j=0;j<3;j++){
15                System.out.print(matrix[i][j]);
16            }
17            System.out.println("");
18        }
19    }
20 }
21 }
```

Inserisci o X o 0:

X

Inserisci o X o 0:

O

Inserisci o X o 0:

x

Inserisci o X o 0:

o

Inserisci o X o 0:

o

Inserisci o X o 0:

x

Inserisci o X o 0:

X

Inserisci o X o 0:

o

Inserisci o X o 0:

o

X0x

oox

xoo

Dichiara 2 matrici 3x3 e fai la moltiplicazione membro membro (alla stessa posizione).

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String[] args) {
5         Scanner scanner=new Scanner(System.in);
6         int matrix_1[][]=new int[2][2];
7         int matrix_2[][]=new int[2][2];
8         int matrix_3[][]=new int[2][2];
9         for(int i=0;i<2;i++){
10             for(int j=0;j<2;j++){
11                 System.out.println("Inserisci un numero: ");
12                 matrix_1[i][j]=scanner.nextInt();
13             }
14         }
15         for(int i=0;i<2;i++){
16             for(int j=0;j<2;j++){
17                 System.out.println("Inserisci un numero: ");
18                 matrix_2[i][j]=scanner.nextInt();
19             }
20         }
21         int moltiplicazione;
22         for(int i=0;i<2;i++){
23             for(int j=0;j<2;j++){
24                 matrix_3[i][j]=matrix_1[i][j]*matrix_2[i][j];
25             }
26         }
27         for(int i=0;i<2;i++){
28             for(int j=0;j<2;j++){
29                 System.out.print(matrix_3[i][j]+" ");
30             }
31             System.out.println("");
32         }
33     }
34 }
35 }
36 }
```

```
Inserisci un numero:  
1  
Inserisci un numero:  
2  
1 4  
1 4
```

## CAPITOLO 8

Scrivi una procedura che prende in input un nome e stampa ciao + nome e chiamala nel main.

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void Introduzione(){
5         Scanner scanner=new Scanner(System.in);
6         System.out.println("Inserisci il tuo nome: ");
7         String nome=scanner.nextLine();
8         System.out.println("Ciao "+nome);
9     }
10    public static void main(String[] args) {
11        Introduzione();
12    }
13}
14
```

```
Inserisci il tuo nome:  
Alessandro  
Ciao Alessandro
```

Dichiara una procedura somma e stampa la somma di 2 numeri presi in input.

```
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void Somma(int numero1,int numero2){
5         int somma=numero1+numero2;
6         System.out.println("La somma è "+somma);
7     }
8     public static void main(String[] args) {
9         Scanner scanner=new Scanner(System.in);
10        int numero1;
11        int numero2;
12        System.out.println("Inserisci un numero: ");
13        numero1=scanner.nextInt();
14        System.out.println("Inserisci un numero: ");
15        numero2=scanner.nextInt();
16        Somma(numero1,numero2);
17    }
18 }
19 }
```

Inserisci un numero:

1

Inserisci un numero:

2

La somma è 3

## CAPITOLO 9

Scrivi una funzione che prende due numeri e restituisce il maggiore.

```
1- import java.util.Scanner;
2 public class Main
3 {
4     public static int Maggiore(int numero1,int numero2){
5         if(numero1>numero2){
6             return numero1;
7         }
8         else{
9             return numero2;
10    }
11 }
12 public static void main(String[] args) {
13     Scanner scanner=new Scanner(System.in);
14     System.out.println("Inserisci un numero: ");
15     int numero1=scanner.nextInt();
16     System.out.println("Inserisci un numero: ");
17     int numero2=scanner.nextInt();
18     System.out.println("Il numero maggiore è il: "+Maggiore(numero1,numero2));
19 }
20 }
21 }
```

```
Inserisci un numero:  
5  
Inserisci un numero:  
4  
Il numero maggiore è il: 5
```

Fai una piccola calcolatrice dove racchiudi all'interno (ciclo while) la presa in input di 2 numeri e fai scegliere l'operazione (addizione o sottrazione) che deve trovarsi al di fuori del main.

## SOLUZIONE ESERCIZI

```
1- import java.util.Scanner;
2 public class Main
3 {
4     public static int Somma(int numero1,int numero2){
5         return numero1+numero2;
6     }
7     public static int Differenza(int numero1,int numero2){
8         return numero1-numero2;
9     }
10    public static void main(String[] args) {
11        Scanner scanner=new Scanner(System.in);
12        while(true){
13            System.out.println("Scegli il tipo di operazione: 1-Somma, 2-Sottrazione");
14            int scelta=scanner.nextInt();
15            System.out.println("Inserisci un numero: ");
16            int numero1=scanner.nextInt();
17            System.out.println("Inserisci un numero: ");
18            int numero2=scanner.nextInt();
19            if(scelta==1){
20                System.out.println(Somma(numero1,numero2));
21            }
22            else{
23                System.out.println(Differenza(numero1,numero2));
24            }
25        }
26    }
27 }
```

```
Scegli il tipo di operazione: 1-Somma, 2-Sottrazione
1
Inserisci un numero:
2
Inserisci un numero:
3
5
Scegli il tipo di operazione: 1-Somma, 2-Sottrazione
2
Inserisci un numero:
3
Inserisci un numero:
1
2
Scegli il tipo di operazione: 1-Somma, 2-Sottrazione

```

