

- 1) All'interno del **package prg.es03** scrivere una classe **SommaVettori** dotata di un metodo **main** che: generi attraverso la **Math.random()** (restituisce un **double** fra 0.0 e 1.0) due array di **double** compresi fra 0 e 1000, **vettore1** e **vettore2**, di dimensione 10; utilizzi un array di interi **risultato** per memorizzare la somma di **vettore1** e **vettore2**; stampi il risultato della somma.
  - 2) Scrivere una classe **SommaMatrici** dotata di un metodo **main** che: generi attraverso la **Math.random()** due matrici di **double** fra 0 e 100, **matrice1** e **matrice2**, di dimensione 3x5; utilizzi una matrice di **double** **risultato** per memorizzare la somma di **matrice1** e **matrice2**; stampi il risultato della somma.
  - 3) Scrivere una classe **Calcolatrice** che contenga al suo interno i seguenti metodi:
    - a. **double[] generaVettore(int L)** che genera un array di **double** casuali fra 0 e 1000 di dimensione L, con L parametro in input al metodo;
    - b. **double[] sommaVettori(double[] a, double[] b)** che riceve in input due array di **double** e, una volta controllato che le dimensioni dei due array siano uguali, restituisce un array che contiene la somma dei vettori in input. Se le dimensioni sono diverse restituisce un vettore uguale ad a;
    - c. **double[] concatenaVettori(double[] a, double[] b)** che riceve in input due array di **double** e restituisca un array che contiene la concatenazione dei vettori in input;
    - d. **void stampaVettore(double[] a)** che riceve in input un vettore di **double** e lo stampa.
    - e. **double[][] generaMatrice(int R, int C)** che genera una matrice di **double** fra 0 e 100 di dimensione RxC, con R e C parametri in input al metodo;
    - f. **double[][] sommaMatrici(double[][] a, double[][] b)** che riceve in input due matrici di interi e, una volta controllato che le dimensioni delle due matrici siano uguali, restituisca in output una matrice che contiene la somma delle matrici in input;
    - g. **void stampaMatrice(double[][] m)** che riceve in input una matrice di **double** e la stampa.
- Utilizzare una classe **TestMatematica** con un metodo **main** che:
- istanzi un oggetto **miaCalcolatrice** della classe **Calcolatrice**;
  - utilizzi l'oggetto **miaCalcolatrice** per generare 3 vettori: **v1** e **v2** di dimensione 3 e **v3** di dimensione 5.
  - utilizzando i metodi dell'oggetto **miaCalcolatrice**, sommi **v1** e **v2**, sommi **v1** e **v3**, concateni **v1** e **v3** e stampi i risultati ottenuti.
  - utilizzi l'oggetto **miaCalcolatrice** per generare 3 matrici **m1**, **m2** e **m3** di dimensione 3x5.
  - utilizzando i metodi dell'oggetto **miaCalcolatrice**, sommi **m1** e **m2**, e **m1** e **m3** e stampi i risultati ottenuti.
- 5) Creare una classe **Dado**. Utilizzando la funzione di libreria **Math.random()** definire il metodo **lancia()** che restituisce un numero intero da 1 a 6.  
 Creare una classe **TestDado** dotata del metodo **main()** in cui si istanziano due oggetti della classe **Dado**. Simulare il lancio dei due dadi per 100 volte e stampare un messaggio tutte le volte che esce 12.  
 Estendere il programma memorizzando in un array il numero di volte che esce ciascun numero. Alla fine stampate l'array. Qual è la combinazione più frequente?
  - 6) Scrivere un programma con i seguenti requisiti.
    - Utilizza una classe **Persona** che contiene le variabili *nome*, *cognome*, *eta* (età) e implementa tutti i costruttori e i metodi che si ritengono necessari. Si dichiara inoltre un metodo *toString* che restituisca in una stringa le informazioni sulla persona e il metodo *equals* che verifica se due oggetti rappresentano la stessa persona. Ricordarsi di utilizzare le convenzioni e le regole per i nomi di classi, metodi e variabili descritte a lezione.
    - Utilizza una classe **Principale** che, nel metodo **main()**, istanzia due oggetti chiamati **persona1** e **persona2** della classe **Persona**, inizializzando per ognuno di essi i relativi campi **nome**, **cognome**, **eta** adoperando l'operatore **dot**.
    - Dichiarare un terzo reference (**persona3**) che punti **persona1**. Controllare che effettivamente **persona3** punti all'oggetto voluto, stampando i campi di **persona3** sempre mediante l'operatore **dot** e che modificando **persona3** si modifica **persona1**.

NOTE PER COMPILAZIONE E TEST A RIGA DI COMANDO IN AMBIENTE LINUX:

**javac -d . nomeClasse.java**  
**java nomePackage.nomeClasse**

*compila e genera il bytecode*  
*esegue il bytecode sulla JVM*