

Inserire nel package **it.unipa.community.nomecognome.prg.n10.esX** le seguenti applicazioni.

- 1) Implementate il metodo **clone()** per le classi **Razionale**, **Complex** e **Date** e verificatene il corretto funzionamento con un programma di test.
- 2) Si utilizzi il metodo **setProperty()** per settare le seguenti proprietà al corrispondente valore (Key, Value): (database, localhost), (dbuser, <vostroNome>), (dbpassword, <vostrapw>). Si utilizzi il metodo **getProperty()** per ottenere il valore associato ad ogni proprietà precedentemente settata.
- 3) Si scriva un'applicazione che, attraverso l'utilizzo di un oggetto della classe **Runtime**, consenta di stampare la memoria ancora libera in JVM (**freeMemory()**) e l'esecuzione del comando "*gedit <nome file da aprire>*". Il nome del file da aprire deve essere passato come parametro all'esecuzione del programma (String args[] del main).
- 4) Si scriva un'applicazione che, dato un numero **double** x, stampi a video:
  - a. valore assoluto di x;
  - b. arrotondamento di x;
  - c. arrotondamento per difetto di x;
  - d. arrotondamento per eccesso di x;
  - e. il massimo e il minimo tra il numero dato x ed un secondo valore y;
  - f. il prodotto tra il numero dato x ed un numero **random** tra 0 e 1;
  - g. il numero  $e^x$  ed il numero  $y^x$ ;
  - h. la radice quadrata di x.
- 5) Utilizzando l'interfaccia **Comparable**, si scriva un'applicazione per ordinare un **array** di oggetti della classe **Date**. Scrivete quindi una classe che implementa l'interfaccia **Comparator** per confrontare due oggetti **Date** e si scriva un'applicazione per ordinare un **ArrayList** di oggetti della classe **Date** utilizzando questa classe.
- 6) Definite una classe **VectorInteger** per la gestione di un array dinamico di oggetti **Integer**. La dimensione del **VectorInteger** viene definita come parametro del costruttore (di default sarà 10) e il **VectorInteger** viene inizializzato con il valore 0. L'accesso ai membri deve avvenire mediante metodi **set** e **get** che verificano le dimensioni del **VectorInteger** ed eventualmente lanciano un'eccezione. Prevedete metodi per la somma, la differenza e il prodotto scalare che restituiscono un nuovo **VectorInteger** contenente il risultato e che verificano che i **VectorInteger** su cui fare l'operazione siano della stessa dimensione (eventualmente lanciate un'eccezione). Prevedete anche un metodo che restituisce un **VectorInteger** ottenuto moltiplicandolo per uno scalare e un metodo che restituisce il modulo (double). Utilizzate un'ArrayList per memorizzare internamente i dati e implementate l'interfaccia **Comparable** col metodo **compareTo** che restituisce -1, 0 oppure 1 in base al confronto dei moduli dei **VectorInteger**. Scrivete infine un programma per testare la classe.

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

*Creare una cartella col proprio cognome sulla scrivania e i relativi file sorgenti al suo interno.*

*Aprire una finestra di **terminale** e digitare:*

**cd Desktop/cognome** oppure **cd Scrivania/cognome** (si posiziona nella directory)

*Creare i file sorgente con **gedit** e salvarli nella propria directory.*

*Digitare:*

**javac nomeClasse.java** (compila e genera il bytecode)

**java nomeClasse** (esegue il bytecode sulla JVM)