

- 1) Nell'esercitazione 4 è richiesto di leggere dei double da tastiera con uno Scanner. Se il dato inserito non è del formato corretto viene lanciata una `java.util.InputMismatchException`. Modificare il programma in modo da gestire le eventuali eccezioni lanciate.
- 2) Aggiungere alla classe Calcolatrice dell'esercitazione 3 la gestione delle eccezioni (i metodi lanceranno eccezioni `java.lang.IllegalArgumentException` quando opportuno invece di stampare messaggi di errore). Utilizzare una classe `TestMatematica` con un metodo `main` che ne verifichi il funzionamento.
- 3) Modificare la classe Razionale dell'esercitazione 4 per far in modo che lanci eccezioni quando si effettuano operazioni non valide (`java.lang.ArithmeticException`) o si prova a creare un oggetto con denominatore nullo (`java.lang.IllegalArgumentException`).
- 4) Modificare la classe Data dell'esercitazione 4 per far in modo che lanci eccezioni quando si prova a creare un oggetto passando parametri non validi al costruttore.
- 5) Modificare la classe `IntegerSet` dell'esercitazione 6.2 per far in modo che lanci eccezioni quando si passano parametri non compresi fra 0 e 99 al metodo `insertElement`.
- 6) Modificare la classe `Stack` dell'esercitazione 7 in modo che lanci opportune eccezioni quando si prova a estrarre da uno stack vuoto o a inserire in uno stack già al limite della capacità.
- 7) Modificare la classe `PositiveIntegerSet` della classe 8 in modo che lanci eccezioni quando si prova a inserire un elemento negativo o a creare un oggetto con elementi negativi.
- 8) Modificare le classi `Razionale` e `Complex` dell'esercitazione 8 per far in modo che lancino opportune eccezioni quando si prova a fare operazione aritmetiche non consentite (somma di `Complex` con `Razionale`, etc...)

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

```
javac -d ../classes -cp ../classes nomeClasse.java compila e genera il bytecode  
java -cp ../classes nomePackage.nomeClasse esegue il bytecode sulla JVM
```