

- 1) Nell'esercitazione 5 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`. Modificate il programma in modo da gestire le eventuali eccezioni lanciate.
- 2) Modificare la classe `Razionale` dell'esercitazione 5 per far in modo che lanci eccezioni quando si effettuano operazioni non valide (`ArithmeticException`) o si prova a creare un oggetto con denominatore nullo (`IllegalArgumentException`).
- 3) Modificare la classe `Data` dell'esercitazione 5 per far in modo che lanci eccezioni `IllegalArgumentException` quando si prova a creare un oggetto passando parametri non validi al costruttore o ai metodi `set`.
- 4) 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`.
- 5) Modificare le classi `Razionale` e `Complex` dell'esercitazione 9 per far in modo che lancino opportune eccezioni quando si prova a fare operazione aritmetiche non consentite (somma di `Complex` con `Razionale`, etc...)
- 6) Modificare la classe `Stack` dell'esercitazione 8 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) Modificate la classe `Stack` dell'esercizio precedente in modo che possa contenere oggetti di qualsiasi tipo (usate i generics)

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

<code>javac -d . es10src/nomeClasse.java</code>	<i>compila e genera il bytecode</i>
<code>java nomePackage.nomeClasse</code>	<i>esegue il bytecode sulla JVM</i>