PROGRAMMAZIONE (9 CFU) – A.A. 2023-2024	ES. 9

1) In alcune esercitazioni passate è stato chiesto di leggere dei double da tastiera con uno Scanner. Se il dato inserito non è del formato corretto viene lanciata una java.util.InputMismatchException. Scrivete un programma che legge 5 double da tastiera, gestendo eventuali eccezioni, e li inserisce in una lista.

2) Modificare la classe Razionale dell'esercitazione 5 per far in modo che lanci eccezioni quando si effettua una divisione per zero (ArithmeticException) o si prova a creare un oggetto con denominatore nullo (IllegalArgumentException).

3) Modificare la classe IntegerSet dell'esercitazione 6.1 per far in modo che lanci eccezioni quando si passano parametri non compresi fra 0 e 99 al metodo insertElement.

4) Definire una classe **Stack** che implementi una pila di 10 String tramite un *ArrayList*<*String*>. Le funzioni membro della classe devono essere:
   void push(String s)
   String pop()
   boolean isEmpty()
   boolean isFull().
   Scrivete un programma che crea un oggetto **Stack** e, tramite un menu testuale, verifica il corretto funzionamento della classe.
   Implementate infine i metodi *toString* e *equals* e verificatene il corretto funzionamento.

5) Modificare la classe Stack dell'esercitazione precedente 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.

6) Basandovi sulla classe Stack precedente scrivete una nuova classe Stack che possa contenere oggetti di qualsiasi tipo (usate i generics).

7) Modificate la classe Razionale dell'esercizio 2 aggiungendo l'implementazione della classe Comparable. Create quindi una lista con 10 razionali e riordinatela utilizzando il metodo Collections.sort().
   Implementate quindi due Comparator che confrontano due numeri razionali semplicemente confrontando i numeratori e i denominatori. Ordinate nuovamente la lista con Collections.sort() e verificate che il comportamento sia quello atteso.

8) Scrivete un programma che, utilizzando il metodo *split* su una stringa contenente il testo di questo esercizio (lo potete incollare direttamente nel codice quando create la stringa), determina il numero totale di parole presenti nel testo e la parola che compare con maggiore frequenza. Potreste anche pensare di utilizzare una **Map<String, Integer>** per memorizzare la frequenza di ciascuna parola utilizzando la parola stessa come chiave. Stampate, infine, la frequenza di ciascuna parola mostrando le parole in ordine alfabetico.

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

`javac -d . es09src/nomeClasse.java`	*compila e genera il bytecode*
`java nomePackage.nomeClasse`	*esegue il bytecode sulla JVM*