

Inserire nel package **it.unipa.community.nomecognome.prg.n08.es0X** le seguenti applicazioni.

- 1) Si consideri la classe **Razionale** (esercitazione n07) e la si modifichi in modo da lanciare opportune eccezioni quando il denominatore è nullo (ArithmeticException) o quando si tenta di sommare ad un **Razionale** un **Numero** che non è **Razionale** (IllegalArgumentException). Si scriva un'applicazione che istanzi oggetti della classe **Razionale** e gestisca le eventuali eccezioni.
- 2) Si consideri la gerarchia di classi **Shape, Circle, Rectangle, Square** (esercitazione n07). Si modifichino le classi in modo da lanciare opportune eccezioni quando gli oggetti delle varie classi vengono inizializzati con valori negativi (IllegalArgumentException). Si scriva un'applicazione che istanzi oggetti delle suddette classi e gestisca le eventuali eccezioni.
- 3) Si modifichino le classi **Image** and **Roi** dell'esercitazione n06 in modo da lanciare e gestire opportune eccezioni laddove necessario (per esempio, IllegalArgumentException se i valori inseriti per le dimensioni delle immagini non sono positivi, oppure se la Roi non è valida o, se quando si calcola la ROIImage, la ROI non è interamente contenuta nell'immagine). Si scriva un'applicazione che consente di testare le modifiche fatte.
- 4) Si consideri la classe **Date** (esercitazione n07) e la si modifichi in modo da lanciare opportune eccezioni qualora la si inizializzi un oggetto con una data non valida (esempio 32/01/2016 oppure 31/13/2015). A tal fine, si scriva una eccezione personalizzata MyIllegalDateException. Si scriva un'applicazione che istanzi oggetti della classe **Date** e gestisca le eventuali eccezioni.
- 5) Si consideri la classe **Line** (esercitazione n05) e la si modifichi in modo da lanciare un'eccezione personalizzata MyIllegalLineException quando si tenta di definire una retta attraverso due punti coincidenti. Si scriva un'applicazione che istanzi oggetti della classe **Line** e gestisca le eventuali eccezioni.
- 6) Si consideri la classe **Plane** (esercitazione n05) e la si si modifichi in modo da lanciare un'eccezione personalizzata MyIllegalPlaneException quando si tenta di definire un piano attraverso tre punti collineari (si usi un oggetto della classe **Line** dell'esercizio precedente per verificare se i punti sono collineari. Si gestiscano le eccezioni lanciate da **Line**.). Si scriva un'applicazione che istanzi oggetti della classe **Plane** e gestisca le eventuali eccezioni.
- 7) Si consideri un **Mazzo** di carte siciliane. Ogni **Carta** è caratterizzata da un seme (ORO, COPPE, BASTONI, SPADE) ed un valoreCarta (ASSO, DUE, ... , SETTE, DONNA, CAVALLO, RE). Un **Mazzo** è composto da 40 carte. Utilizzare delle enumerazioni per rappresentare semi e valori di ogni carta. All'interno dell'enumerazione valoreCarta, prevedere un metodo punti() che restituisce i punti di ogni carta nel gioco della briscola (ASSO=11 punti, TRE=10 punti, DONNA=2 punti, CAVALLO=3 punti e RE=4 punti, tutte le altre valgono 0 punti).
- 8) Modificare la classe **Date** dell'esercizio 1 in modo da utilizzare una enumerazione per i mesi dell'anno.

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

Creare una cartella col proprio cognome sulla scrivania con all'interno le directory **classes** e **src**.

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

**cd Desktop/cognome/classes** oppure **cd Scrivania/cognome/classes**

Creare i file sorgente con **gedit** e salvarli nelle sottodirectory di **src** corrispondenti al package.

Digitare:

**javac -d . ../src/it/unipa/.../prg/n08/es0X/\*.java** (compila e genera il bytecode)  
**java it.unipa.community.....prg.n08.es0X.Main** (esegue il bytecode sulla JVM)