PROGRAMMAZIONE (9 CFU) – A.A. 2022-2023	ES. 08

1) Definire una classe **Stack** che implementi una pila di 100 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.

2) Nella gerarchia descritta nell'esercitazione 6 e relativa a **Shape**, **Circle**, **Rectangle**, **Square**, introdurre una interfaccia **Scalable** che dichiara un metodo *scale(double factor).* Il metodo consente di modificare le dimensioni dei lati/raggio di una figura in base al fattore di proporzionalità *factor*.
   Introdurre anche l'interfaccia **Drawable** che dichiara un metodo *draw()* per visualizzare i dati dell'oggetto.
   Rendere **Shape** classe astratta e verificare che per questa classe non si possono istanziare oggetti.

3) Si costruisca una gerarchia di classi per rappresentare veicoli su terra considerando le seguenti entità: **Veicolo, VeicoloAMotore, Ciclomotore, Automobile, Bicicletta**.
   Un **Veicolo** è caratterizzato da una posizione, una velocità iniziale e un'accelerazione (bidimensionali).
   In base ai valori di velocità e accelerazione, un **Veicolo** segue la legge di moto uniformemente accelerato:

$$x = x_0 + v_{0x} * t + a_x * t * t$$
$$y = y_0 + v_{0y} * t + a_y * t * t$$

   (si utilizzi il metodo *muovi(double t)* dove *t* rappresenta la variazione di tempo durante cui il veicolo si muove che aggiorna la posizione del veicolo).
   VeicoloAMotore è un **Veicolo** caratterizzato da un motore con una cilindrata predefinita.
   Ciclomotore è un **VeicoloAMotore** caratterizzato dal numero di telaio (*long*).
   Automobile è un **VeicoloAMotore** caratterizzato dal numero di targa (*String*).
   Bicicletta è un **Veicolo** caratterizzato dal modello (*String*).
   Scrivere un'applicazione che consenta di testare la gerarchia delle classi e di simulare il movimento nel tempo di una **Bicicletta**, un**'Automobile** e un **Ciclomotore** avviati tutti allo stesso istante di tempo.

   Si modifichi la gerarchia in modo da implementare classi astratte dove possibile. Rendere abstract il metodo *muovi()* della classe **Veicolo**. Si supponga che i **veicoliAMotore** si muovano di moto uniformemente accelerato come visto nell'esercizio precedente, mentre le **Biciclette** seguano la seguente legge di moto:

$$x = x_0 + v_{0x} * t + a_x * t * t$$
$$y = cos(x)$$

   Si utilizzi una classe Main per testare le nuove classi.

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

`javac -d . es08src/nomeClasse.java`	*compila e genera il bytecode*
`java nomePackage.nomeClasse`	*esegue il bytecode sulla JVM*