



UNIVERSITÀ DEGLI STUDI DI SALERNO

*Corso di Laurea Triennale in Informatica, Classe Resto 1*

**CORSO DI  
PROGRAMMAZIONE OBJECT ORIENTED**  
Prof. MASSIMO FICCO

**Prova di Programmazione**  
**2 febbraio 2023**

*Cognome..... Nome..... Matr...../.....*

**Traccia Esercizio 1 – 25 Punti**

Sviluppare un programma per la creazione di un gioco di fantasy. Il programma dovrà essere così costituito:

Una interfaccia **Pers-Master** esporta i seguenti metodi:

- *Ruota(int g)*: tale metodo ruota il personaggio a destra di 'g' gradi;
- *Avanti(int s)*: tale metodo sposta avanti il personaggio di 's' unità di misure; può generare un'eccezione se la posizione supera il limite del quadro di gioco, pari a 100 unità di misura.
- *Indietro(int s)*: tale metodo sposta indietro il personaggio di 's' unità di misure; può generare un'eccezione se la posizione supera il limite del quadro di gioco, pari a 100 unità di misura.

Una classe astratta **Personaggio** implementa l'interfaccia di cui sopra.

- Tale classe possiede i seguenti attributi: identificativo univoco personaggio, tipo di personaggio, energia, posizione iniziale nel quadro di gioco, posizione durante il gioco, etc.
- La classe implementa il seguente metodo astratto:
  - *Colpisci(Personaggio pers)*: tale metodo colpisce il Personaggio *pers* riducendo la sua energia di un unità;

Due classi **Cavaliere** e **Orco** derivano dalla classe **Personaggio**, ereditando tutti i metodi su descritti.

Ogni Cavaliere e Orco ha un identificato univoco che viene generato da un contatore che s'incrementa ogni volta che viene creato un nuovo Personaggio.

Una classe **Gestore** che gestisce la posizione dei personaggi all'interno del quadro di gioco.

Tale classe deve creare una lista di Personaggi a caso e fornire un metodo per salvare su un file lo stato di tutti i personaggi durante il gioco.

Inoltre, tale classe implementa i seguenti metodi:

- *get\_Pos(Personaggio pers)*: restituisce la posizione del personaggio *pers*;
- *get\_Pos\_all()*: restituisce la posizione e l'identificativo di tutti i personaggi presenti nel quadro di gioco;
- *get-Type(Personaggio pers)*: restituisce il tipo di personaggio (Orco, Cavaliere);

Implementare un'interfaccia grafica che permette all'utente di indicare all'interno di una form il numero di Personaggi che compongono la lista Personaggi.

**Traccia Esercizio 2 – 2 Punti**

Dato l'applicativo implementato nel primo esercizio, creare un metodo con espressioni lambda, per stampare l'identificativo univoco di ogni Orco presente nella lista di Personaggi.

**Traccia Esercizio 3 – 3 Punti**

Impiegando lo Stream API e le lambda espressioni siano realizzati dei metodi per l'elaborazione dei dati nell'applicazione implementata nel primo esercizio:

- Calcolare il numero dei Cavalieri che posseggono energia maggiore di 50 e hanno posizione minore di 50 unità di misura.
- Estrarre l'identificativo univoco di ogni Cavaliere che possiede energia maggiore di 70 e stamparlo a video.