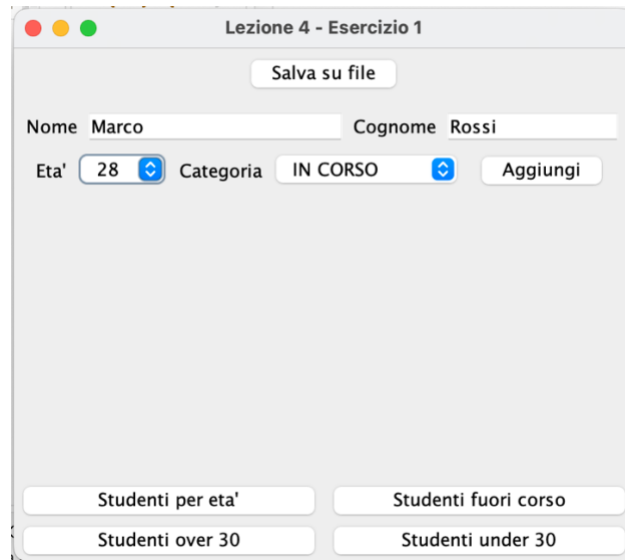


POO – Lezione 4

Si realizzi la seguente GUI per la gestione delle informazioni di Studenti universitari.



In particolare, si garantiscano i seguenti requisiti e/o funzionalità:

- a) Aggiunta di uno studente tramite appositi campi input.
- b) Stampa degli studenti, ordinati per età, tramite espressione lambda (Suggerimento: Come fatto per l'esercizio 2 del 20/12/2022).
- c) Stampa degli studenti fuori corso tramite un'espressione lambda equivalente al metodo `actionPerformed(ActionEvent e)`. (Suggerimento: si ricordi che la relativa interfaccia è per definizione un'interfaccia funzionale).
- d) Definizione di un'eccezione **NON controllata** `CampiVuotoException` che viene lanciata se almeno uno dei campi non è stato compilato.
- e) Definizione di una espressione lambda per la stampa degli studenti fuori corso.
- f) Definizione di una espressione lambda tramite metodo statico per la stampa di tutti gli studenti over 30 (età > 30) (Suggerimento: Come fatto per l'esercizio 3 del 20/12/2022).
- g) Utilizzo della stessa espressione lambda (senza ridefinire un nuovo metodo statico) per la stampa di tutti gli studenti under 30 (età < 30).
- h) Salvataggio su file di tutti gli studenti presenti sulla lista.
- i) Definizione di un'eccezione **CONTROLLATA** `SalvataggioControllatoException` che sia gestita opportunamente dal metodo public `void saveOnFile(String fileName)` della classe manager.
- j) Si utilizzi opportunamente, sul pannello principale della GUI, un `BorderLayout`.
- k) Modularizzare la creazione dei sotto-pannelli (Suggerimento: si provi prima tramite la definizione di metodi per ogni pannello e poi tramite la definizione di classi concrete).