

## **Relazione progetto JAVA**

### **Programmazione II**

#### **Prof. Levi**

#### *Implementazione*

Le classi NewAr e NewAr2 sostituiscono per “nome” le classi Board

In entrambe le implementazioni quando andiamo a creare un contenuto se inseriamo una categoria che non abbiamo definito in precedenza il sistema la aggiunge automaticamente alla lista delle categorie disponibili, se esiste già ovviamente non viene inserito di nuovo nella lista dedicata

NewAr implementa DataBoard e si comporta come una collezione di Data che rappresenta un utente

Ogni utente quindi dovrà avere una password di tipo alfanumerica e dei contenuti che potrà inserire

Tutte le funzioni che permettono di interagire sui dati necessitano della password per essere eseguite

Abbiamo definito due ArrayList di stringhe uno per contenere una lista di amici ed uno una lista di nomi di categorie da noi inserite

Poi un ArrayList di tipo category che contiene entrambi, in modo da associare gli amici che vogliamo a delle determinate categorie

Per la gestione dei contenuti ho usato sempre ArrayList, ma in questo caso che prendesse come valori oggetti di tipo Data

Tramite dei metodi pre stabiliti possiamo interagire con la lista in modo da inserire , rimuovere o modificare i dati

NewAr2 implementa DataBoard e si comporta come una collezione di Data che rappresenta un utente

Ogni utente quindi dovrà avere una password e dei contenuti che potrà inserire

In questo caso la password veniva modificata tramite una funzione dedicata.

Tutte le funzioni che permettono di interagire sui dati necessitano della password per essere eseguite

Questa volta abbiamo definito una lista Vector di tipo category per la gestione intera degli amici e della categoria senza l'ausilio extra di due arraylist di stringe

Per la gestione dei contenuti abbiamo optato per HashMap che avesse come valori un tipo intero che rappresentasse un codice identificativo e Data per la gestione dei contenuti

Data al suo interno avevo tutti gli elementi caratteristici che ci venivano richiesti da testo, come la categoria , l'autore, il testo e il numero di like, e anche una lista che permetteva di vedere chi avesse messo like

Control serve per implementare delle funzioni che facilitano i controlli vari

### *Gestione delle Eccezioni*

Per la gestione delle eccezioni sono stati definiti principalmente 2 gruppi di eccezioni, uno che comprendesse eccezioni specifiche di diverso tipo, per la gestione di errori considerati più “gravi” rispetto ad altri, come per esempio:

- DataNotFound
- DataDuplicate
- FailLogin
- InvalidData

Alcune di queste eccezioni venivano gestite dalla classe Control con delle funzioni dedicate. L'altro gruppo di Eccezioni veniva rappresentato da InternalError che gestiva eccezioni generiche, specificando però ogni volta di che cosa si trattasse l'errore.

### *Batteria di Test*

Per generare una batteria di test nel file Start troviamo il main con due diversi test, uno per la prima implementazione di NewAr ed uno per la seconda.

In entrambi i test vengono verificate almeno una volta tutte le funzioni implementate, cercando di lanciare delle eccezioni per verificare il corretto funzionamento.

Alcuni dati, usati più volte per tutto il programma vengono definiti da delle variabili, invece altri meno significativi vengono inseriti direttamente.