PROGRAMMAZIONE II - A. A. 2019 – 20
Primo Progetto-sessione autunnale

Studente: **Stefano Pea** Matricola: **561020**
Corso A

- ***Implementazione DataBoard<E extends Data>***

DataBoard<E extends Data> rappresenta un contenitore di oggetti generici che estendono il tipo di dato *Data*.
Entrambe le implementazioni proposte rispettano l'interfaccia seppur presentando alcune differenze per quanto riguarda la rappresentazione delle categorie presenti nella board.
I campi contenuti all'interno di quest'ultima sono:

1. ***MasterPassw***: rappresenta la password impostata dal creatore della board nel momento della sua creazione ed è fondamentale per permettere la modifica e l'accesso ai metodi della board.
2. ***Owner***: rappresenta il proprietario della board ed è assegnato al momento della sua creazione.
3. ***numCategories***: intero che rappresenta il numero di categorie associate alla board, al momento della creazione è impostato a 0.
4. ***categories***: questo campo può essere modificato in base alle preferenze riguardo al tipo di contenitore da utilizzare per oggetti di tipo *Category<E extends Data>*

All'interno della classe Board1 *categories* viene rappresentato con una *Map* che associa una stringa (rappresentante il nome della categoria) ad un oggetto di tipo *Category<E extends Data>*, semplificando cosi' la ricerca di una particolare categoria, essendo sufficiente ricercarla tramite il nome ad essa associato.
Nella Board2 invece sono state utilizzate due *ArrayList* (una contenente elementi di tipo String e l'altra di tipo *Category<E extends Data>*), dove la lista denominata *names* contiene tutti i nomi delle categorie presenti nella board, mentre quella chiamata *categories* contiene le categorie vere e proprie.
In questo caso la ricerca viene effettuata in maniera leggermente diversa:
ogni volta che si richiama il metodo *createCategory()* viene aggiunta in coda alla lista *names* la stringa contenente il nome associato alla categoria e in

categories viene aggiunto l'oggetto di tipo *Category* (lo stesso vale anche per removeCategory()).
Per questo motivo per cercare una categoria *x* occorre scorrere *names* per trovare l'indice a cui è salvata la stringa *x* e di conseguenza quello sarà anche l'indice a cui è salvata la categoria corrispondente in *categories*.
Per quanto riguarda i controlli, *Databoard* non accetta parametri di tipo *null*, ne' per il campo *Owner* ne' per il campo *MasterPassw*, ma accetta stringhe vuote.

- ***Implementazione Data***

Data rappresenta invece il dato che viene inserito nelle varie categorie presenti nella board.
Data contiene al suo interno vari campi:

1. ***Owner***: rappresenta il proprietario del dato ed è assegnato al momento della sua creazione.
2. ***Category***: rappresenta la categoria di appartenenza del dato, al momento della creazione è impostata a null.
3. ***Post***: Stringa di testo che rappresenta un oggetto simile ad un post.
4. ***NumLike***: intero che rappresenta il numero di like che quel dato ha ricevuto.
5. ***likes***: ArrayList di stringhe che contiene i nomi di tutti gli amici che hanno messo like al dato.

Anche *Data* (come *DataBoard*) non accetta parametri null ma accetta stringhe vuote.

- ***Test correttezza implementazioni***

Per quanto riguarda i test, sono presenti due file specifici chiamati *test_board_1* e *test_board_2* che eseguono gli stessi test ma su due tipi differenti di board, *test_board_1* su oggetti di tipo Board1 e *test_board_2* su Board2.
Nel terminale è possibile controllare la correttezza dei test che si presentano nella forma riportata qui sotto:

"- Creo una board con una password null, ottengo NullPointerException
OK: java.lang.NullPointerException"

in cui *OK* rappresenta un catch corretto dell'eccezione, in caso contrario si ha un catch di un'eccezione non corretta e viene mostrato *ERROR*.