

RELAZIONE SUL PROGETTO DI PR2

(1° parte: Java)

Nome: Lorenzo	Cognome: Ammatuna	Matricola : 567400	Corso : A
----------------------	-----------------------------	---------------------------	------------------

INTRODUZIONE

L'interfaccia proposta chiamata DataBoard rappresenta la bacheca di un utente. Quest'ultimo può creare categorie, aggiungere dati in bacheca e definire i dati con la quale i suoi amici può interagire.

La *struttura dati* utilizzata è la seguente tripla:

`<passw, {Insieme delle categorie per dati}, {Insieme delle categorie per amici}>`

Nello specifico:

- `passw` è una stringa rappresentate la password necessaria ai controlli di identità dell'utente
- `{Insieme delle categorie per dati}` è un insieme di coppie dove ognuna di esse è formata da:
 - o Un nome di Categoria
 - o Un insieme di Dati
- `{Insieme delle categorie per amici}` è un insieme di coppie dove ognuna di esse è formata da:
 - o Un nome di Categoria
 - o Un insieme di Amici

Sono state create due *implementazioni* dell'interfaccia DataBoard:

- Una con l'utilizzo di Vector ovvero MyDataBoard
- Una con l'utilizzo di HashMap ovvero MyDataBoardV2

Per quanto riguarda i *Dati* si è scelto di:

- Associare ad ogni dato un contatore per i like e un vettore contente gli amici che hanno messo like al dato
- Estendere Data con il tipo Photo, costituito da una serie di parametri:
 - o Descrizione della foto
 - o Dimensione della foto
 - o Luogo in cui è stata scattata la foto
 - o Data (Giorno/Mese/Anno) in cui è stata scattata la foto

Per quanto riguarda gli *Amici* si è scelto di:

- Permettere ad ogni amico di essere associato a più categorie

- Ogni amico è rappresentato attraverso una stringa

Per quanto riguarda le **Categorie** si è scelto di:

- Gestirle attraverso:
 - o 2 vettori chiamati dataCategory e friendCategory formati da oggetti rispettivamente delle due classi chiamate CategoryData e CategoryFriends per quanto riguarda la prima implementazione dove:
 - CategoryData è formata da :
 - Una stringa NameCat che definisce il nome della Categoria
 - Un vettore CatData che definisce un vettore di dati
 - CategoryFriends è formata da:
 - Una stringa NameCat che definisce il nome della Categoria
 - Un vettore CatFriends che definisce un vettore di stringhe, dove ogni stringa rappresenta un amico.
 - o 2 HashMap chiamate hashDataCategory e hashFriendsCategory per quanto riguarda la seconda implementazione dove:
 - Ogni campo di hashDataCategory è costituito da
 - una stringa che definisce il nome della Categoria (definisce anche una chiave)
 - Un hashSet di dati
 - Ogni campo di hashFriendCategory è costituito da
 - una stringa che definisce il nome della Categoria (definisce anche una chiave)
 - Un hashSet di stringhe, dove ogni stringa rappresenta un amico

Infine, abbiamo introdotto dei **vincoli** progettuali:

- Se in dataCategory è presente una determinata categoria x in posizione y allora in friendCategory in posizione y si troverà la categoria x e viceversa
- Un amico può inserire un like ad un dato y solo una volta. La gestione di tutto ciò viene effettuata attraverso il vettore degli amici che hanno messo like presente in ogni oggetto Data
- Se un amico viene rimosso da una o più categorie i like che esso ha inserito non vengono cancellati
- Se in hashDataCategory è presente una categoria x allora quella categoria sarà presente in hashFriendsCategory e viceversa.
- Considerare unico ogni dato presente in bacheca (non esistono 2 dati uguali bacheca)
- Considerare unica ogni Categorie (non esistono 2 categorie uguali)

METODI AGGIUNTIVI

DataBoard

Metodi	Descrizione del metodo
private boolean verifyPass (String password)	Verifica la validità della password immessa
private int verifyCategory (String Categoria)	Restituisce la posizione della categoria "Categoria". La posizione è valida sia per friendCategory sia per dataCategory. Se la categoria non esiste viene restituito -1
private int verifyFriends (int position, String Amico)	Restituisce la posizione dell'amico "Amico" nella categoria in posizione "position". Se l'amico non è presente in quella categoria viene restituito -1
private int verifyData (E dato)	Restituisce la posizione di "dato". Se il dato non è presente in bacheca restituisce null
private E findData (E dato)	Restituisce una copia di "dato". Se il dato non è presente in bacheca restituisce null
private int findCategory (int pos, E dato)	Restituisce la posizione della categoria dove è presente "dato" in posizione "pos". Se "dato" non è presente in posizione "pos" in nessuna categoria restituisce -1.

DataBoardV2

Metodi	Descrizione del metodo
private boolean verifyPass(String password)	Verifica la validità della password immessa
private E verifyDato (E dato)	Restituisce una copia di "dato". Se dato non è presente in bacheca restituisce null
private String verifyCategory (E dato)	Restituisce la stringa corrispondente alla categoria associata a "dato". Se "dato" non è presente in bacheca restituisce null

TEST CASE

Per la verifica del funzionamento del progetto nell'uso dei vari metodi e implementazioni è stato creato un main usato appunto da test case del progetto.

Il main è già predisposto per la verifica del programma attraverso l'uso della 1° o della 2° implementazione dell'interfaccia DataBoard.

Di default è possibile proseguire alla verifica dei test del main attraverso un oggetto che implementa l'interfaccia DataBoard attraverso l'uso di Vector (1° implementazione: MyDataBoard).

È possibile procedere alla verifica dei test presenti nel main con un oggetto che implementa l'interfaccia DataBoard attraverso l'uso di HashMap (2° implementazione : MyDataBoardV2) commentando l'oggetto di tipo MyDataBoard e decommentando l'oggetto di tipo MydataBoardV2.