Progetto Java 2019/2020

Marco Colavita

May 2, 2020

Introduzione

Il progetto consiste nel realizzare un contenitore di oggetti generici che estendono il tipo data chiamato $DataBoard$. La bacheca garantisce la privacy dei dati fornendo un proprio meccanismo della condivisione dei dati. Ogni dato presente nella bacheca ha associato la categoria del dato. Le categorie sono create esclusivamente dal proprietario del bacheca che puo' inoltre stilare una lista di contatti(amici) a cui saranno visibili i dati i dati per ogni tipologia di categoria. I dati possono essere modificati solo dal proprietario di DataBoard; gli amici invece possono associare un like al dato in bacheca.

Studio del software

Come richiesto, DataBoard e' stato implementato mediante l'utilizzo di due strutture dati differenti. La prima e' stata realizzata con la struttura dati $ArrayList < Categoy >$ (Category verra' descritta in segiuto), mentre la seconda implementazione e' stata realizzata mediante la struttura dati $HashMap < String, Category >$ dove $String$ e' la chiave e $Category$ e' il valore associato alla chiave.

Interfacce

Le interfacce utilizzate sono $DataBoard < EextendsData >$, contenente soltanto la descrizione informale dei metodi e $Data$ implementata dalla classe MyData.

Da qui in aventi verranno descritte tutte le classi utilizzate per la realizzazione della bacheca.

Category

La classe Category contiene come variabili di istanza una stringa per il nome, un $ArrayList < E >$ per i dati e un $ArrayList < String >$ per gli amici appartenenti a quella categoria.
La classe e' stata implementata con i metodi $getName()$ (restituisce il nome della categoria), $friendList()$ (restituisce l'array di amici della categoria), $dataList()$ (restituisce i dati della categoria), $getSizeFriends$ (restituisce la dimensione dell'array di amici), $getSizeCollection()$ (restituisce la dimensio dell'array di dati), $getFriend(inti)$ (restituisce l'oggetto con indice di dell'array di amici), $addFriend(String\ friend)$ (aggiunge friend all'array di amici), $removeFriend(int\ i)$ (rimuove l'oggetto con inidice i nell'array di amici) ed infine il metodo $equals$ che permette le operazione di confronto tra utenti.

MyData

La classe MyData implementa tutti i metodi eraditati dall'interfaccia $Data$. La classe prende come parametri di istanza un elemento di tipo E per indicare il dato, un intero per il numero di like associato al dato e un $ArrayList < String >$ per gli amici che mettono like. Tale classe implementa i metodi $getData$ (restituisce il dato), $getLikes$ (resituisce il numero di like dell'oggetto), $Display()$ (stampa l'elemento e i suoi relativi likes), $insertLikes(String\ friend)$ che incrementa il numero di likes di quell'oggetto e aggiunge friend all'array di amici. L'ultimo metodo e' $cloneData()$ che restituisce una copia dell'elemento.

MyDataBoard1

Classe che implementa l'interfaccia $DataBoard$. Ha il compito di realizzare tutti i metodi ereditati dall'interfaccia. Tale classe, come scritto in precedenza e' stata realizzata con la struttura dati $ArrayList < Category >$. Sono stati aggiunti a questa classe i metodi $findCategory(String\ name)$ (restituisce true se la categoria e' presente nella bacheca) e $findFriend(String\ Category\ String\ friend)$ (che restituisce true se l'amico comapre all'interno della categoria). All'intenrno di questa classe e' stata definita la classe $Sort$ che implementa $Comparator$ per oridnare i dati per numero di likes.

Board2

Classe che implementa $DataBoard$. Come MyDataBoard1 anch'essa ha il compito di realizzare tutti i metodi ereditati dall'interfaccia. Questa classe e' stata realizzate con la struttura dati $HashMap < String, Category >$. In questa classe a differenza di MyDataBoard non sono stati aggiunti metodi extra, ma solo la classe $Sort$ che implementa $Comparator$ per ordinare i dati per nuemro di likes.

Eccezioni

Le eccezioni che vengono lanciate sono: $NullPointerExeption$, lanciata se uno dei valori passati come parametro risulta uguale a NULL; $FriendAlreadyPresent$, lanciata se un amico compare gia' in una categoria; $LikesAlreadyPresent$, lanciata se compare gia' il like di un amico ad un dato nella categoria, ed infine $ImpossibleToPerform$, lanciata se si hanno problemi nelle operazioni di creazione, inserimento, rimozione etc.

Test

Classe che ha i compito di verificare il corretto funzionamento di DataBoard, quindi vengono verificati i risultati di tutti i metodi dell'interfaccia. Verra' verificato il corretto funzionamento di entrambe le strutture dati utilizzate per l'implementazione della bacheca. Sono stati testati tutti i metodi dell'interfaccia comprese le eccezioni che lanciano questi ultimi.