Relazione del primo progetto intermedio in Java del corso di Programmazione II

Anno Accademico 2019/2020

Simone Schiavone Corso A Mat. 582418

Scelte di implementazione:

Le classi MyDataBoard<E extends Data> e MyDataBoard2<E extends Data> implementano l'interfaccia DataBoard<E extends Data>. Ho rappresentato Data come una classe astratta ed ho definito le classi Immagine, Post, NotaAudio e Video che estendono Data.

La specifica della classe MyDataBoard prevede la presenza di un attributo stringa "password", settato al momento della creazione di un'istanza di MyDataBoard, che funge da meccanismo di sicurezza necessario per la corretta esecuzione di alcuni metodi della classe. Il comportamento di una MyDataBoard è simile a quello di una bacheca in cui vengono affissi dei Dati (post, immagini, ecc..) che sono catalogati in una o più categorie. Nelle implementazioni si suppone che nella DataBoard non ci siano amici che abbiano lo stesso nome e non ci siano categorie con stesso attributo "nome". Entrambe le implementazioni permettono la pubblicazione di un dato su una sola DataBoard ma all'interno di essa è possibile la pubblicazione in più categorie, purché in ognuna di esse non ci siano duplicati. Tale scelta si è resa necessaria in quanto se un dato fosse pubblicato su più databoards, i like (attributo del singolo dato) potrebbero provenire da entrambe le databoards; la invocazione del metodo getiterator (restituisce un iteratore che genera tutti i dati in bacheca ordinati rispetto al numero di like) determinerebbe un ordinamento dei dati rispetto ai like complessivi, che però non sarebbero filtrati dai like ottenuti in altre databoards. Inoltre se due amici diversi con lo stesso nome, presenti in due databoard distinte, avessero accesso a due categorie che contengono uno stesso dato (situazione possibile in quanto non sono ammessi amici con lo stesso nome in una istanza di DataBoard ma non in tutte le databoard), qualora entrambi volessero mettere un like al dato otterrei che il secondo risulterebbe già nella lista di utenti che hanno messo like perciò sarei in una situazione anomala. Per tale motivazione ho ritenuto necessario aggiungere alla databoard un attributo intero statico Id in modo che ogni oggetto della classe databoard abbia un Id diverso, al fine di poter effettuare i controlli sulla

pubblicazione multipla di un singolo dato. Per eliminare le ambiguità rispetto alla presenza o meno di amici con lo stesso nome sarebbe utile sviluppare una classe amico con associato una stringa nome ed un id, in modo di effettuare i controlli sulla presenza di duplicati considerando l'id. In merito al metodo InsertLike della DataBoard al momento dell'eliminazione dell'amico dalla lista degli amici con accesso alla categoria,qualora egli avesse messo un like ad un dato di quella categoria questo rimane inserito. In entrambe le implementazioni, nei metodi che restituiscono oggetti del tipo Data (o suoi sottotipi), è stato scelto di effettuare una deep-copy dei dati in modo da evitare degli accessi dall'esterno agli elementi della databoard.

La sostanziale differenza tra le due implementazioni è rappresentata dalle strutture dati utilizzate. La prima implementazione prevede che la DataBoard abbia un attributo di tipo ArrayList<Categoria<E>> che contiene tutte le categorie. Introduco quindi la classe Categoria<E extends Data> generica che al suo interno contiene un ArrayList<E> di dati che sono pubblicati in bacheca ed un ArrayList<String> di nomi di amici autorizzati alla visione dei dati della categoria. La seconda implementazione non necessita della classe Categoria<E> in quanto prevede l'utilizzo di due HashMap; una del tipo <String,ArrayList<E>> che rappresenta l'associazione tra categoria ed i dati della categoria, ed una <String, ArrayList<String>> che rappresenta l'associazione tra la categoria e le stringhe degli amici che hanno visibilità ai dati della categoria. La classe astratta Data ha come attributi una stringa nome, un ArrayList di stringhe che contiene i nomi degli utenti che hanno già messo like e un intero db che rappresenta l'id della DataBoard su cui un dato è pubblicato, con la convenzione che l'intero 0 indica che il dato non è stato pubblicato in nessuna databoard. Poiché la DataBoard ha un comportamento simile ad un social network, nell'implementazione del metodo InsertLike se un amico mette like due volte ad un dato, tale azione comporta una rimozione del like.

Per quanto riguarda il testbench, ho effettuato delle prove su entrambe le implementazioni testando l'effettivo lancio delle eccezioni per alcuni casi limite dei metodi della DataBoard. Ho inoltre mostrato in alcune circostanze l'effettiva creazione di una copia del del dato, per esempio nel metodo Get.