

RELAZIONE PRIMO PROGETTO PROGRAMMAZIONE II

Scopo di questo documento è fornire la documentazione relativa alla progettazione e realizzazione del primo progetto della sessione autunnale proposto dai docenti del Corso di Programmazione II, che ha lo scopo di applicare i concetti e le tecniche di programmazione Object-Oriented esaminate durante le lezioni.

ISTRUZIONI PER ESEGUIRE IL PROGRAMMA

Per eseguire il programma è necessario posizionarsi da terminale nella cartella:

...ProgettoPR2ZIP/ProgettoPR2 ed eseguire i comandi:

- compilazione: `javac Main.java`
- esecuzione: `java Main`

Verrà eseguita la batteria di test che comprende due metodi statici di test:

- test eccezioni (in ordine, prima e seconda implementazione);
- test app (in ordine, prima e seconda implementazione).

Nella prima parte vengono testate tutte le eccezioni lanciate dai metodi implementati in `MyDataboard1` e `MyDataBoard2` grazie al costrutto `try{...} catch() {...}`, mentre nella seconda parte viene effettuata un test pulito (senza la possibilità che alcuna eccezione venga lanciata) che verifica il corretto funzionamento dei metodi delle suddette classi.

SCELTE PROGETTUALI

SPECIFICA:

Data: tipo di dato astratto modificabile che intende rappresentare un dato appartenente ad una bacheca, composto da un campo autore e da un campo testo. Ho scelto di mantenere questo TDA il più generico possibile, senza inserire nella struttura il numero di like, l'insieme di friend che hanno messo like e la categoria a cui il dato appartiene richiesti dal testo, progettando un tipo di dato che sia slegato da queste componenti.

DataBoard: tipo di dato astratto modificabile che intende rappresentare una bacheca di dati generici raggruppati per categoria. Non ci possono essere due categorie con lo stesso nome. Per ogni dato, esso è associato ad una ed una sola categoria. La bacheca garantisce la privacy dei dati fornendo un proprio meccanismo di gestione della condivisione dei dati. Il proprietario della bacheca può definire le proprie categorie e stilare una lista di contatti (amici) a cui saranno visibili i dati per ogni tipologia di categoria. I dati condivisi sono visibili solamente in lettura: in particolare i dati possono essere visualizzati dagli amici ma modificati solamente dal proprietario della bacheca. Gli amici possono associare un "like" al contenuto condiviso. Ai metodi proposti nel testo del progetto sono stati aggiunti alcuni metodi di supporto (`getCategoryName()`, `getFriendsCategory()` e `getLikesCategory()`). Da sottolineare è che la `removeFriend()` elimina l'amico friend dall'insieme di amici abilitati alla lettura dei dati in una categoria ma non elimina i like già messi da friend ai dati presenti.

Like: Tipo di dato astratto modificabile che intende rappresentare il numero di likes associati ad un dato generico e l'insieme degli amici che hanno messo like al medesimo dato generico.

Category (utilizzata solo per la prima implementazione): Tipo di dato astratto modificabile

che intende rappresentare una categoria appartenente ad una bacheca di dati generici. La categoria possiede un nome che la identifica e tre insiemi che contengono oggetti di tipo E (generici), Like e String e che rappresentano rispettivamente insiemi di dati generici, di Like e di amici. Ogni oggetto di tipo Like è associato ad uno ed uno solo oggetto di tipo generico. L'insieme di String rappresenta gli amici che sono abilitati alla lettura dei dati generici presenti all'interno dell'insieme di dati generici e all'inserzione di like per un determinato oggetto generico.

DataLike: Tipo di dato astratto che intende rappresentare un dato di tipo generico con associato il numero di likes che esso ha ricevuto. Questo TDA viene utilizzato solamente per implementare il metodo `getIterator()` grazie a cui viene restituito un iteratore sui dati che vengono ordinati in modo crescente.

IMPLEMENTAZIONI:

La **prima implementazione**, che possiede tre variabili di istanza (admin, password e un ArrayList di MyCategory), sfrutta una classe di appoggio (MyCategory) che descrive una singola categoria della bacheca. Questa classe, al suo interno, contiene una stringa name che la identifica, una lista di generici, una lista di MyLike (che contiene il numero di likes di un dato e l'insieme degli amici che hanno messo like) e una lista di amici (quelli abilitati alla lettura dei dati). Gli ArrayList di dati generici e di MyLike hanno la stessa dimensione, in quanto ad ogni dato generico corrisponde, a pari indice, un oggetto di tipo MyLike.

La **seconda implementazione** invece non utilizza la classe d'appoggio MyCategory in quanto si basa su una struttura interamente contenuta nella classe MyDataBoard2 che è composta dalle solite stringhe admin e password e da tre HashMap: la prima mappa una stringa che identifica una categoria ad un HashSet di dati generici, la seconda mappa sempre dalla chiave categoria ad un HashSet di stringhe che rappresentano gli amici abilitati alla lettura dei dati presenti nella categoria e la terza mappa un dato generico appartenente alla bacheca ad un oggetto di tipo MyLike, istanziato all'aggiunta del dato.

ECCEZIONI:

Nel programma vengono lanciate le seguenti eccezioni (nelle implementazioni di alcuni metodi, alcune eccezioni, per esempio la `NullPointerException`, sono lanciate nelle classi più interne):

- **NullPointerException**, disponibile in Java, unchecked, se è presente qualche parametro uguale a null;
- **WrongPasswordException**, non disponibile in Java, checked, se la password inserita alla chiamata del metodo non corrisponde alla password del proprietario della bacheca;
- **InexistentElementException**, non disponibile in Java, checked, se un oggetto cercato all'interno della bacheca non viene trovato;
- **DuplicateException**, non disponibile in Java, checked, se un oggetto (o un like nel caso della `insertLike()`) è già presente nella bacheca e si sta cercando di inserirlo nuovamente;
- **PermissionDeniedException**, non disponibile in Java, checked, se viene chiamata la `insertLike()` passando per parametro un amico che non dispone dei privilegi necessari per mettere like ad un dato (friend non è presente nell'insieme degli amici abilitati alla lettura dei dati nella specifica categoria).