**Relazione del progetto in Java del corso di Programmazione 2**

**A.A. 2019/20**

**SIMONE IOZIA N.549108**

**Obiettivo**

Realizzare, progettare e documentare una collezione di oggetti generici che estendono il tipo *Data*, divisi in categorie, mediante l’uso di 2 strutture di implementazione differenti.

**Strutture di implementazione**

Le strutture di implementazione usate nel progetto sono: ArrayList e HashMap, descritte nelle classi *BoardV1* e *BoardV2*, che implementano l’interfaccia **Databoard**, all’interno della quale sono presenti tutti i metodi necessari per lavorare con i dati con le relative specifiche. Entrambe le strutture hanno come parametro una stringa che rappresenta la password (in teoria conosciuta solo dal proprietario della bacheca) utile a creare, rimuovere o richiedere intere categorie o singoli dati.

* In ***BoardV1*** la bacheca è realizzata come un ArrayList in cui gli elementi che lo compongono sono le varie categorie di dati, descritte dalla classe *Categories* spiegata nel paragrafo successivo.
* In ***BoardV2*** la bacheca è realizzata mediante un HashMap che “mappa” i nomi delle varie categorie con la classe *Categories* corrispondente.

**Classe *Categories***

La classe *Categories* rappresenta invece una singola categoria. Ogni categoria contiene una stringa che identifica il nome della categoria (nel caso della HashMap, potrebbe risultare superfluo), una lista di dati corrispondenti alla categoria, e una lista di amici che possono visualizzare i dati contenuti in essa.

**Classe MyIterator**

La classe *MyIterator*, che implementa l’interfaccia *Iterator*, disponibile sulla libreria di Java, mi permette di creare un iteratore sprovvisto di metodo *remove* (infatti se chiamato lancia un’eccezione). Tale classe richiede come parametro un iteratore ed è usata nei metodi *getIterator* (nella quale viene passato un iteratore che agisce su una lista che contiene tutti i dati in bacheca ordinata in base ai like) e *getFriendIterator* (nella quale viene passato un iteratore che agisce su una lista di tutti i dati visibili da un amico), presenti in *BoardV1* e *BoardV2*.

**Classe *DecrOrder***

Usata nel metodo *getIterator*, la classe *DecrOrder* permette di ordinare la lista contenente di tutti i dati in bacheca in base al numero dei like in modo decrescente. Implementa l’interfaccia *Comparator*, disponibile sulla libreria di Java.

**Classe *Data***

La classe *Data* è utile a definire cosa contiene il singolo dato presente in bacheca: un contenuto, specificato da una stringa; una categoria, alla quale il dato viene associato solo dopo che ne faccia parte; un numero di likes; la lista di amici che hanno messo like. La classe implementa l’interfaccia ***DataType*** che contiene tutti i metodi necessari per lavorare con i singoli dati. I dati contenuti in bacheca possono anche estendere tale classe.

**Test**

Nel main è presente una serie di test sulle 2 strutture di implementazioni (per cambiare struttura, basta modificare *BoardV1* in *BoardV2*, o viceversa). Il codice commentato rappresenta tutte le possibili eccezioni che i singoli metodi testati possono lanciare. Di seguito sono descritte le eccezioni checked:

* *WrongPasswordException*: lanciata quando la password richiesta da un metodo non risulta uguale a quella scelta.
* *DuplicateException*: lanciata quando si tenta di creare una categoria avente nome uguale ad un’altra oppure quando si tenta di inserire un amico omonimo ad un altro all’interno della stessa categoria.
* *ElementNotFoundException*: lanciata quando si passa come parametro una categoria non presente in bacheca, oppure un dato o un amico non presente in una determinata categoria.