

Leonardo Vona
Matricola 545042
Corso A
Programmazione II – Primo Progetto Intermedio
03/12/2019

Scelte di implementazione.

In un'ottica di programmazione difensiva, si è previsto che nei metodi copy, put, get dell'interfaccia venga effettuato un copy in / copy out degli elementi in modo da privare un accesso alla struttura interna dell'implementazione che non sia attraverso i metodi messi a disposizione.

Il copy in / copy out viene effettuato attraverso una deep copy. Essa viene implementata attraverso il metodo clone() di Data, il quale serializza l'oggetto in byte e deserializza i byte in un nuovo oggetto indipendente dal primo.

Un limite di questa realizzazione della deep copy è che richiede elementi che implementano Serializable.

La batteria di test è effettuata attraverso la classe TestDB. Il costruttore richiede un intero che rappresenta l'implementazione che si vuole testare. I test sono effettuati su tutti i metodi pubblici dell'interfaccia DataBoard.

CheckRep() è un metodo che verifica se l'invariante di rappresentazione è rispettata.

Il metodo equals() di Data è stato riscritto in modo che due Data per essere uguali è sufficiente che abbiano lo stesso dato effettivo, indipendentemente dal numero di like.

Per ordinare i dati in base al numero di like si è creata una classe SortByLikes che implementa Comparator<Data>. In essa viene implementato il metodo compare in modo che un dato con più like sia prima di un dato con meno like. Tale classe è utilizzata nel metodo sort(.) per ordinare la lista di elementi che estendono Data.

Implementazione 1.

La prima implementazione prevede l'utilizzo di un User utente e di una HashMap<String, Category<E>> per contenere le categorie della bacheca. Le chiavi della HashMap sono i nomi delle categorie.

Ogni categoria è composta da una lista di elementi che estendono Data che appartengono a tale categoria e una lista di amici che hanno accesso ai dati nella categoria. Un elemento che estende Data ha al suo interno il dato vero e proprio e il numero di like associati.

Implementazione 2.

La seconda implementazione utilizza un User per l'utente e una List<Category<E>> per contenere le categorie associate alla bacheca.