# DOCUMENTAZIONE DATABOARD

1. **DataBoard**

**L’interfaccia DataBoard** è la specifica di un contenitore di oggetti generici che devono estendere il tipo Data. Il contenitore deve essere organizzato in modo che ogni dato faccia parte di almeno una categoria e che ad ognuna di queste corrisponda un elenco di amici (anche vuoto). Un amico può accedere solo ai dati contenuti nelle categorie di cui fa parte. L’accesso consiste nella visualizzazione e nella possibilità di aggiungere un like al dato in questione. L’amministratore invece, che per usufruire dei suoi privilegi dovrà autenticarsi tramite una password alfanumerica, avrà accesso alle funzionalità di aggiunta (rimozione) di una categoria e di un dato/amico a (da) una categoria. Se la password inserita per eseguire uno dei metodi amministratore è sbagliata viene lanciata una InvalidPasswordException. I metodi amministratore sono i seguenti:

* **createCategory**, aggiunge una categoria a DataBoard a patto che al suo interno non sia presente una categoria con lo stesso nome (nel caso questo si verificasse viene lanciata una ExistingCategoryException)
* **removeCategory**, rimuove una categoria da DataBoard. I post e gli amici presenti all’interno della categoria persistono in DataBoard solo se presenti anche in altre categorie. Se la categoria specificata non esiste lancia un InvalidCategoryException
* **addFriend**, permette di aggiungere un amico a una categoria. Se la categoria specificata non esiste lancia un InvalidCategoryException
* **removeFriend**, rimuove un amico da una categoria. Se la categoria specificata non esiste lancia un InvalidCategoryException, se l’amico non è presente all’interno della categoria lancia un InvalidFriendException
* **put**, permette di aggiungere un dato a una categoria. Se la categoria specificata non esiste lancia un InvalidCategoryException, se nella categoria specificata il dato è già presente lancia una DuplicateDataException
* **get**, permette di ottenere una copia del dato in bacheca. Essendo l’oggetto restituito una deep copy dell’originale, esso può essere modificato e reinserito nella stessa categoria dalla quale è stato copiato. Se il dato da copiare non è presente in nessuna categoria di DataBoard lancia una InvalidDataException
* **remove**, rimuove un dato dato da DataBoard (e quindi da tutte le categorie in cui esso è presente) e lo restituisce. Se il dato non è presente in nessuna categoria di DataBoard lancia una InvalidDataException
* **getDataCategory**, restituisce una lista di tutti i post presenti nella categoria specificata. Se la categoria specificata non esiste lancia un InvalidCategoryException
* **getIterator**, restituisce un iteratore per poter scorrere tutti i dati presenti in DataBoard. I dati presenti in più categorie vengono visualizzati una volta sola.

I metodi hai quali hanno accesso gli amici sono i seguenti:

* **insertLike**, permette ad un amico di inserire un like ad un post. Il numero massimo di like che un amico può inserire a un singolo dato è 1. Se si prova a superare quel limite lancia una DuplicateLikeException. Se l’amico prova a inserire un like ad un dato che non è presente in nessuna delle categorie alle quali l’amico ha accesso viene lanciata una InvalidFriendException
* **getFriendIterator**, restituisce un iteratore per permettere ad un amico di scorrere tutti i dati ai quali ha accesso. Se l’amico non è presente in nessuna delle categorie di DataBoard viene lanciata una InvalidFriendException

I metodi accessibili a qualsiasi utente sono:

* **numCategories**, restituisce il numero di categorie presenti in DataBoard
* **numFriend**, restituisce il numero di amici presenti in una data categoria
* **numData**, restituisce il numero di dati presenti all’interno della categoria

1. **Data**

L’interfaccia **Data** è la specifica di un contenitore per un parametro di tipo generico (stringa, intero, ecc…) affiancato da un autore, una descrizione e una lista di amici che hanno messo like al dato in questione. L’interfaccia contiene i metodi per poter modificare il dato, restituire il dato, aggiungere un like (max 1 per amico), ottenere il numero di like totali, ottenere una copia del dato e ottenere una stampa a video del dato.

1. **Prima implementazione**

La prima implementazione di DataBoard utilizza una HashMap (fornita da java.util.HashMap) che associa il nome della categoria rappresentato come stringa a un oggetto di tipo Category, il quale contiene la lista degli amici e dei post che ne fanno parte. La classe category contiene i metodi per ottenere un dato o un amico, ottenere tutti i dati o gli amici presenti, aggiungere un dato o un amico oppure verificare se questi sono presenti.

1. **Seconda implementazione**

La seconda implementazione di DataBoard utilizza due HashMap friends e dataSet che associano rispettivamente il nome di una categoria, rappresentato come stringa, all’insieme degli amici e all’insieme dei post ad essa appartenenti. Gli amici e i post di una categoria sono rappresentati come una List (fornita da java.util.List).

1. **Differenze tra le implementazioni**

La prima implementazione risulta più intuitiva e permette di ottenere sia gli amici sia i post con una sola ricerca all’interno di una tabella hash. La seconda implementazione ha però il vantaggio di non utilizzare alcuna classe ausiliaria. La differenza di prestazioni tra le due si manifesta solo nel caso in cui di una categoria siano richiesti sia gli amici sia i dati.