Relazione Progetto Java PR 2

Venturi Ludovico
Docente: Francesca Levi

UNIPI, Novembre 2019

Indice

1 Scelte progettuali

Nella relazione verranno spiegate le scelte progettuali e implementative che sono state prese.

1.1 Data

OVERVIEW: *Data rappresenta un dato sottoforma di un insieme di 3 attributi e alcune operazioni. È una struttura astratta immutable, di dimensione finita e fissa.*

Data viene implementata come classe astratta. Tale scelta deriva dalla volontà di attribuire a tutte le classi che discendono da *Data* delle caratteristiche comuni, ovvero dei metodi già implementati e una struttura implementativa di base:

private String ***dataName****;*
private String ***content****;*
private String ***category****;*

Come da specifica la classe *Data* riporta anche il metodo astratto display che verrà implementato dalle sottoclassi:

public abstract void display();

Data ridefinisce anche *equals()* per permettere la deep equality e mette a disposizione dei getter per accedere ai dati privati in lettura.
Rappresenta un *contratto* cui tutte le sottoclassi (= i vari dati) dovranno sottostare, ovvero condivideranno con *Data* la struttura di base, i vari metodi non astratti e dovranno ridefinire il metodo *display()*.

1.1.1 Ipotesi

- Non ci sono setter poichè si è ipotizzato che *Data* fosse una struttura *immutable*.

(Nel testo viene riportato ≪*i dati possono essere visualizzati dagli amici ma modificati solamente dal proprietario della bacheca*≫: ciò è stato interpretato come: "la modifica consiste nell'aggiunta o la rimozione dei dati, non nella modifica effettiva del contenuto dei dati").

1.1.2 MyData

MyData è una sottoclasse di *Data*. Implementa il metodo display senza aggiungere altro alla struttura della sopraclasse.

1.2 Board «E extends Data»

1.2.1 Ipotesi

- Non sono ammessi elementi *null*
- Non sono ammessi duplicati di alcun genere
- Il numero di likes non dipende solamente dal dato ma anche dalla bacheca in cui si trova ⇒ *Data* non possiede il contatore dei like, ma questo si trova nella bacheca, relativamente ad ogni dato

1.2.2 Implementazione 1

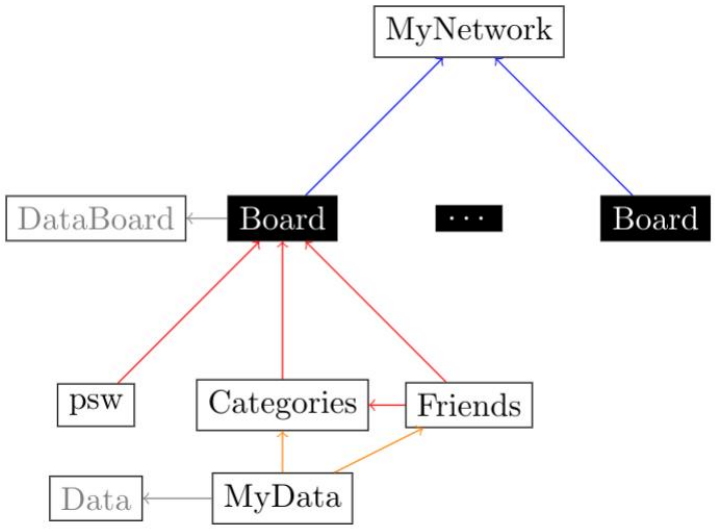

Figura 1: Struttura generale del progetto con la prima implementazione di Board

Non ho riportato la specifica di ogni metodo nel codice di *Board«E extends Data»* poichè risultava troppo confusionario; l'implementazione ha comunque seguito di pari passo la specifica riportata nell'interfaccia *DataBoard«E extends Data»*.

1.3 Implementazione 2: Board «E extends Data»

2 Eseguire il codice

2.1 Test ed esempi

Non saranno verificate *tutte* i casi in cui parametri sono null per ovvie ragioni, così come tutte le eccezioni ripetute, quali i controlli che la categoria esista o che l'amico sia presente nella lista amici; per queste ultime, pur se generabili in differenti metodi, verranno esplicitamente testate solamente una volta ciascuna. In generale più istruzioni su un singolo blocco *try* indicano che l'ultima sarà quella che genera l'eccezione mentre le precedenti sono eseguite con successo. Lista di test effettuati nel *main*:

- password della bacheca $\ll 8$ caratteri
- get di una bacheca non presente
- password errata
- categoria già presente
- rimozione di una categoria non presente
- condivisione di una stessa categoria con uno stesso amico
- condivisione di una categoria non presente nella bacheca
- rimozione di un amico non presente nella lista amici
- rimozione di un amico da una categoria cui non ha accesso, anche se presente nella lista amici
- inserimento di un dato già presente
- inserimento di un dato la cui categoria non è presente in bacheca
- get di un dato la cui categoria non è presente
- get di un dato non presente
- get di un dato precedentemente inserito in modo corretto ma la cui categoria è stato poi rimossa
- rimozione di un dato non presente
- getDataCategory e modifica della lista ritornata
- amico vuole inserire like ad un dato di una categoria non condivisa con lui
- amico vuole inserire like ad un dato cui lo ha già messo
- amico vuole inserire like ad un dato non presente
- ITERATORI, prove varie
- elimino una categoria e itero sui dati di tale categoria tramite una amico con cui essa era condivisa