

# Relazione Progetto Java PR 2

Venturi Ludovico  
Docente: [Francesca Levi](#)

UNIFI, Novembre 2019

## Indice

<b>1</b>	<b>Scelte progettuali</b>	<b>1</b>
1.1	Data . . . . .	1
1.1.1	Ipotesi . . . . .	1
1.1.2	MyData . . . . .	1
1.2	Board «E extends Data» . . . . .	2
1.2.1	Ipotesi . . . . .	2
1.2.2	Implementazione 1 . . . . .	2
1.3	Implementazione 2: Board «E extends Data» . . . . .	2
<b>2</b>	<b>Eseguire il codice</b>	<b>3</b>
2.1	Test ed esempi . . . . .	3



# 1 Scelte progettuali

Nella relazione verranno spiegate le scelte di implementazione e progettuali 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*.

(`setCategory()` viene chiamato soltanto da )

(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

### 1.2.2 Implementazione 1

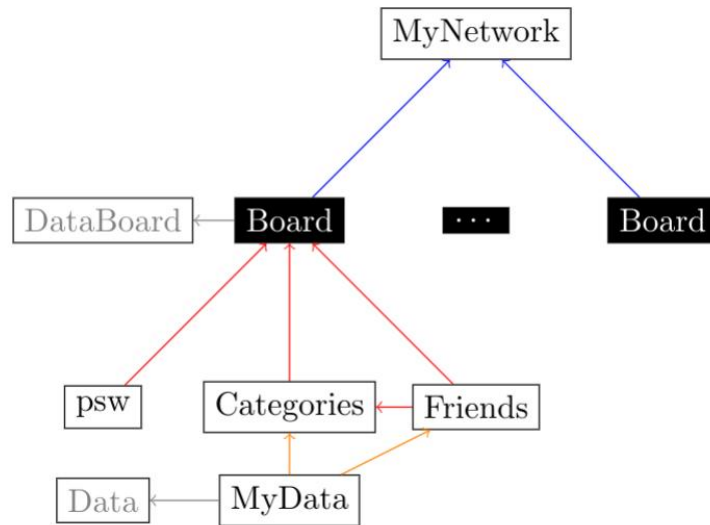


Figura 1: Struttura generale del progetto con la prima implementazione di Board

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.  
Lista di test effettuati nel *main*:

- password della bacheca  $\ll$  8 caratteri
- get di una bacheca non presente