



UNIVERSITÀ DI PISA

Dipartimento di Informatica
Corso di Laurea in Informatica

Relazione progetto Java

Programmazione II

Prof. Gianluigi Ferrari
Prof.ssa Francesca Levi

Carmine Vitiello
578070- Corso A

Anno Accademico 2020/2021

Relazione

Dettagli sul metodo di sviluppo

Il progetto è stato sviluppato con il supporto dell'IDE *VisualStudio* e una sua estensione per il linguaggio java, senza tuttavia l'utilizzo di strumenti esterni. Infatti, grazie all'IDE sono riuscito a compilare e creare una batteria di Test, che viene eseguita attraverso la compilazione e l'esecuzione del Main chiamato App.java.

Dettagli implementativi

Il progetto richiede la progettazione, implementazione e documentazione di un **Blog**, che contiene tutte le relazioni tra gli **Utenti** e **following**, cioè persone che l'Utente specifico X segue. Il Blog gestisce tutti gli Utenti con una board di comando, che permette al creatore del blog di controllare in modo efficiente cosa è presente nella piattaforma e soprattutto cosa fanno gli Utenti. Infatti, gli **Utenti** "registrati" al blog possono creare dei contenuti chiamati **Post**, che possono guardare tutti gli utenti del blog. Inoltre attraverso una diversa estensione del blog si può introdurre la possibilità di segnalare un **Post**, che contiene un contenuto non idoneo alla piattaforma, così si ha un metodo per controllare meglio gli **Utenti** in caso che non rispettino le regole del blog. Infine tutti i file sono stati commentati seguendo le linee guida di Javadoc, ed ogni classe è corredata da una breve overview e dalle sue AF e RI.

1 Post

Il tipo di dato **Post** rappresenta in maniera astratta l'oggetto che l'utente andrà a creare ed a condividere. Ogni post ha alcuni campi che lo identificano, quali Id cioè un codice identificativo che si incrementa in modo sequenziale, *author* che coincide sempre con il **Username** del proprietario del Post, *Testo* che invece rappresenta il contenuto del post, *TimeStamp* che rappresenta le informazioni relative a data e ora e infine i *menzionati* che rappresentano gli Utenti che vengono citati nel testo del Post, grazie al simbolo speciale @ seguiti dall'Username dell'Utente citato.

2 Utente

Il tipo **Utente** mi rappresenta la persona che si è iscritta al blog, visto che per ovviare a molte problematiche progettuali, che rendevano il codice inconsistente su più fronti soprattutto sul lato controlli ed eccezioni. L'Utente contiene varie informazioni utili al blog per creare e gestire le connessioni tra persone. Infatti, l'oggetto ha un **id** per essere identificato dal gestore del blog, un **Username** identificativo e univoco, un **Nome** della persona, una **Lista di Post** per avere una raccolta dei Post scritti in passato e un **counter** per rappresentare il numero dei seguaci dell'Utente. Inoltre attraverso i suoi metodi l'Utente può creare un Post e controllare il numero dei follower, cioè il numero delle persone che lo seguono (seguaci). Infine, ho pensato di aggiungere una estensione dell'Utente per gestire un blog che avesse delle specifiche ulteriori necessarie per controllare i contenuti scritti dagli Utenti. Infatti, attraverso la classe Utente2 sono stati implementati dei nuovi parametri e dei nuovi metodi per permettere in prima battuta, all'Utente di segnalare dei contenuti non idonei alla piattaforma e in un secondo momento dare la possibilità alla

piattaforma di annotare i messaggi di report mandati dagli Utenti, in modo tale da poter gestire gli Utenti segnalati come meglio si crede.

3 Blog

Questa è la parte più rilevante del progetto, visto che rappresenta una zona dove vengono definiti sia i metodi di sistema sia una parte dedicata alla persona che avrà il compito di amministrare il blog. La classe è formata da una **lista di Username**, una **lista di oggetti Utenti** e una **mappa di following**, che rappresenta tutte le relazioni tra gli Utenti iscritti al blog. Infatti, attraverso la map la classe Blog annota le persone iscritte e che seguono. Inoltre, attraverso i vari metodi della classe si possono ricavare molte informazioni che erano richieste nelle specifiche del Progetto. Infatti, non descriverò tutte le possibili azioni perchè sono definite in modo maniacale nel codice attraverso i commenti e la notazione **Javadoc**. L'unica cosa a cui vorrei dedicare l'attenzione è l'estensione del blog nella classe Blog2 che serve ad implementare una piattaforma con dei controlli maggiori soprattutto sui contenuti dei Post, che potrebbero essere offensivi per qualche Utente. Inoltre, la soluzione trovata prende in considerazione l'isolamento di questi Utenti scorretti, ma ogni gestore potrà decidere come meglio crede, visto che i dati inerenti ad un Utente isolato non vengono cancellati totalmente, quindi in caso che si voglia aggiungere nuovamente un utente è possibile farlo.

Test

Il **file di test** è stato scritto come un'esecuzione totale del Progetto, nel quale si cerca di evidenziare tutte le possibili azioni che sono rese disponibili dal programma. Cercando di evidenziare i possibili errori che verrebbero raccolti al tempo di esecuzione attraverso le eccezioni. Infatti, il risultato del test si può verificare attraverso l'output video, che andrà a mostrare la chiamata del metodo, il risultato del lancio dell'eccezione, in caso che sia presente, attraverso un messaggio [FATAL ERROR] e alcune volte dei test con la dicitura [TEST] a video.