



# MICROBLOG

Progetto Java | Anno Accademico 2020/2021

Leonardo Pantani | Corso A

## ISTRUZIONI PER LA COMPILAZIONE

*Con Eclipse:*

Il progetto è stato sviluppato con *IntelliJ IDEA*, tuttavia è stato esportato per supportare anche la compilazione ed esecuzione con *Eclipse*. Basta importare il progetto tramite l'apposita funzione "Import", in seguito selezionare dal menu a tendina "General" l'opzione "Projects from Folder or Archive" e scegliere la cartella principale del progetto. Tutti i file .java sono sotto la cartella "src".

*Da riga di comando su Linux:*

1. `git clone`  
[https://github.com/LeonardoPantani/Progetto\\_Java\\_PR2-UNIFI](https://github.com/LeonardoPantani/Progetto_Java_PR2-UNIFI)
2. `cd Progetto_Java_PR2-UNIFI/src`
3. `javac *.java app/*.java exception/*.java`
4. `java Main`

## SCELTE IMPLEMENTATIVE

### Utente

Si è preferito non aggiungere un sistema di autenticazione utenti per semplicità di sviluppo. Ogni utente è identificabile come richiesto, dall'username, però ogni eccetto azione eccetto la creazione di un post (come la segnalazione) è svolta anonimamente. Ogni username deve avere una dimensione in caratteri tra i 3 e i 24 (i valori sono salvati dentro delle costanti).

### Post

È una classe "immutable". Inizialmente l'obiettivo era rendere il testo di un post modificabile, ma ciò avrebbe reso difficile creare un sistema di segnalazioni adatto perché un utente avrebbe potuto scrivere un post segnalabile e poi modificare il testo per evitare la punizione. Così il sistema è più sicuro.

### Report

Il sistema di report permette di segnalare un post per alcuni motivi già preimpostati, oppure un motivo personalizzato che può essere scritto al momento della segnalazione. Ogni report è composto dall'id del post segnalato, l'insieme dei motivi per cui il post è segnalato e un eventuale campo custom che viene riempito solo nel caso di una segnalazione personalizzata. Per esempio, un post può essere segnalato sia per "violent\_content" che per "promotes\_terrorism" ma anche

per un motivo personalizzato. Il suddetto post sarà stato segnalato per 3 volte (1 solo report, ma 3 segnalazioni diverse).

### Social Network

La classe implementa tutti i metodi richiesti dalla consegna. Oltre a questi, sono stati creati altri metodi privati/protetti per rendere più semplici alcune operazioni sulla map e sulla lista dei post. Gli utenti vengono inizializzati da un metodo privato specifico per la batteria di test "initializeUser" che riceve l'username dell'utente da creare e un set di stringhe che conterrà i nomi degli utenti seguiti da username. Implementati anche i metodi printAllPosts & printAllUsers per motivi di debug/test. In un post è possibile menzionare un utente nel testo aggiungendo una chiocciola "@" prima del nome utente. Si presuppone che la lunghezza degli username sia compresa tra i 3 e i 24 caratteri. Alcuni esempi di menzione validi: @leonardo, @marco, @andrea123. Perché una menzione sia considerata tale, non ci devono essere spazi tra la chiocciola e il nome utente. La funzione getMentionedUsers supporta più menzioni in ogni post.

### Safe Social Network

Questa classe estende Social Network implementando i metodi che gestiscono la segnalazione di un post. La classe principalmente lavora sulla lista di oggetti di tipo Report rp. Il metodo principale reportPost è overloaded, il primo lavora sulle segnalazioni semplici (come "violent\_content"), l'altro invece richiede come reason "custom" e text il motivo speciale per cui segnalare il post con id post\_id. Si è scelto di implementare il metodo getMostReportedPosts che restituisce una map<Post, Integer> che contiene i post più segnalati in ordine decrescente. Si può specificare un numero minimo di report per visualizzarlo nella lista. Infine, il metodo removeReportedPosts consente di rimuovere i post con tot segnalazioni tutti insieme senza specificare manualmente gli id dei post.

### Report Reason (enum)

Questa classe contiene un tipo personalizzato vincolato di nome ReportReason che può assumere solo valori specificati. Questi sono i motivi di una segnalazione e si è obbligati a scegliere solo tra quelli. Nel caso di una segnalazione custom, si deve scegliere l'opzione "custom" e, tramite la funzione setCustomReport, specificare anche il motivo personalizzato della segnalazione.

### Regex username (controllo dell'username)

```
\B@(?!(?:[a-z0-9.]*_){2})(?!(?:[a-z0-9_\.]{2})[._a-z0-9]{3,24})\b
```

Prima della "@" accetta caratteri maiuscoli e minuscoli, numeri, punti e trattini bassi.

Dopo la "@" accetta caratteri maiuscoli e minuscoli, numeri, punti e trattini bassi.

La menzione viene "interrotta" da tutti i caratteri eccetto lettere, numeri, punti e trattini bassi.