# Relazione Java 2019/2020

**1 Prima implementazione**

Come prima implementazione ho pensato di utilizzare due HashMap che contengano, come chiavi, la categoria e uno, come valori, un ArrayList<E> contenente tutti i dati associati a quella categoria e l’altro un ArrayList<String> contenete la lista di amici associati a quella determinata categoria. Ho pensato a questa soluzione proprio perchè ogni categoria è unica, e, a ogni categoria, possono appartenere uno o più dati di tipo generico E (o anche nessuno) e, data la struttura dell’HashMap secondo la quale ogni chiave è unica, la gestione di ciò sarebbe stata più veloce. Ho inoltre ipotizzato che i vari nominativi e password degli utenti debbano essere diversi da null e da stringa vuota, e che i dati siano diversi da null e che inoltre le password debbano avere un numero di caratteri maggiori di 8 e debbano necessariamente contenere caratteri alfanumerici, per rendere più sicura la protezione dei dati. Per effettuare tutti gli opportuni controlli su username, password, categoria e testo ho implementato una classe check utilizzata per verificare il corretto funzionamento dei dati nella Board.   
L’utente di una bacheca può aggiungere post che ho pensato come testi, e gli amici dell’utente possono aggiungere like, e vedere i post solo in lettura.  
Ho implementato il metodo display() che effettua semplicemente una stampa dei dati con tutti i suoi campi.   
La ricerca con questa implementazione viene effettuata in tempo lineare, che è un grande vantaggio.

**2 Seconda implementazione**

Come seconda implementazione ho pensato di implementare una classe Category costituita dai campi nome, un ArrayList<String> contenete gli amici e un ArrayList<E> contenente i dati. Inoltre in questa classe ho implementato alcuni metodi ausiliari utilizzati poi in Board\_2.  
Board\_2 è costituita da una ArrayList<Category> contenente le categorie, e i campi password e autore.  
Per il controllo del corretto inserimento dei dati i vari metodi usano la classe Check, che verifica la correttezza semantica della password. L’accesso a una bacheca è regolato da una password quindi solo l’utente può accedervi e effettuare modifiche su di essa, invece tutti i dati sono in sola lettura per gli amici.   
Attraverso questa implementazione la ricerca non è ottimizzata.

DATA

Ho strutturato la classe Data come l’insieme di categorie, la lista di amici, il creatore di un post, il numero di like e il post come stringa di testo.  
Entrambe le implementazioni estendono la classe Data.

TEST

La classe test è la classe contenente il main, quest’ultimo contente due metodi: batteryTestHashMap(), batteryTestArrayList() nei quali ho testato manualmente tutte le funzioni implementate, inserendo tutti i casi possibili. Tramite la struttura try-catch ho catturato tutte le possibili eccezioni che il codice poteva generare, senza interrompere l’esecuzione del programma, gestendole in maniera opportuna.