

1^ implementazione:

Bacheca1 è un contenitore di dati generici che estendono il tipo `Data1`.

Essa estende l'interfaccia `Dashboard1` e ne implementa tutti i metodi.

Si compone di tre elementi principali:

- una password settata dal proprietario della bacheca tramite il costruttore;
- una lista di categorie creata dal proprietario;
- una lista di dati generici aggiunti dal proprietario.

Ogni **categoria** creata dal proprietario della bacheca è caratterizzata da un *nome* e da una *lista di amici* con i quali egli sceglie di condividere i dati relativi a quella determinata categoria.

Ogni **dato** creato dal proprietario della bacheca contiene un *nome* che lo identifica, una *categoria* a cui è associato e un numero di *likes*, che gli amici, ai quali è permesso visualizzare i dati della relativa categoria, possono lasciare.

La **creazione delle categorie** avviene per mezzo del metodo specifico "*createCategory*" all'interno della bacheca, il quale utilizza a sua volta il costruttore della classe `Category`, e si accerta che il nome desiderato per la nuova categoria non sia nullo, vuoto o già stato utilizzato.

La **creazione della bacheca e dei dati** avvengono invece nel programma, per mezzo dei costruttori delle due classi.

In questa visione del progetto, chiunque può creare nuovi dati (nuove classi che estendono `Data1`), semplicemente passando al costruttore della classe `Data1` il nome del dato desiderato (non nullo e non vuoto).

Successivamente l'utente può inserire il dato creato nella propria bacheca tramite il metodo "*put*", identificandosi con la propria password (quella che ha utilizzato per creare la bacheca la prima volta) e scegliendo la categoria alla quale quel determinato dato è associato. Il metodo eseguirà poi tutti i controlli che garantiranno la non presenza di dati con lo stesso nome, o di dati associati a categorie che non sono state mai create.

Il proprietario può inoltre rimuovere categorie (a patto che non siano associate a dei dati ancora esistenti), aggiungere e rimuovere amici da una categoria.

L'**aggiunta e la rimozione degli amici** dalle categorie avvengono rispettivamente con i metodi "*addFriend*" e "*removeFriend*". Il primo, garantisce che l'utente non inserisca nomi per gli amici che siano nulli, vuoti o che siano già stati utilizzati.

Altre operazioni possibili sono: ottenere una copia di un dato in bacheca tramite il metodo "*get*", rimuovere dati dalla bacheca per mezzo del metodo "*remove*" e registrare un like ad un dato da parte di un amico (che ovviamente dovrà essere stato registrato nella lista degli amici della relativa categoria) con il metodo "*insertLike*". In questa visione del progetto, ogni amico può lasciare tutti i likes che vuole, basta che sia presente nella lista degli amici della categoria relativa al dato che gli piace.

Infine la bacheca offre la possibilità di:

- ottenere una *lista di tutti i dati* presenti in essa di una determinata categoria, passata come parametro, con il metodo "*getDataCategory*";
- ottenere un *iteratore*, su di una **lista non modificabile** (impedendo quindi l'operazione di *remove*), che genera tutti i dati che sono presenti in bacheca ordinati in base al numero di likes (in ordine decrescente), tramite il metodo "*getIterator*";
- ottenere un *iteratore*, anch'esso su una lista non modificabile, che genera tutti i dati presenti in bacheca che sono stati condivisi con l'amico passato come parametro, per mezzo del metodo "*getFriendIterator*".

2^ implementazione:

Bacheca2 è un contenitore di dati generici che estendono il tipo `Data2`.

Essa estende l'interfaccia `Dashboard2` e ne implementa tutti i metodi.

Si compone di tre elementi principali:

- una password settata dal proprietario della bacheca tramite il costruttore;

- una mappa di categorie e dati generici associati a ciascuna di esse;
- una una mappa di categorie e amici aggiunti ad ognuna.

Quest'altra visione del progetto non necessita della classe Category. Le due implementazioni infatti differiscono soltanto per questo particolare.

Ogni **categoria** creata dal proprietario della bacheca è aggiunta sia alla mappa degli amici sia alla mappa dei dati automaticamente, in modo da far sì che le due mappe abbiano sempre le stesse chiavi (ovviamente la condizione è rispettata anche dal metodo *removeCategory* che rimuove la categoria da entrambe le mappe, a patto che ad essa non siano associati dati ancora esistenti).

Ogni **dato** creato dal proprietario della bacheca contiene un *nome* che lo identifica, una *categoria* a cui è associato e un numero di *likes*, che gli amici, ai quali è permesso visualizzare i dati della relativa categoria, possono lasciare.

La **creazione delle categorie** avviene per mezzo del metodo “*createCategory*” all'interno della bacheca, il quale si accerta che il nome desiderato per la nuova categoria non sia nullo, vuoto o già stato utilizzato e poi provvede a creare un nuovo mapping in entrambe le mappe della bacheca.

La **creazione della bacheca e dei dati** avvengono invece nel programma, per mezzo dei costruttori delle due classi.

Anche in questa visione del progetto, chiunque può creare nuovi dati (nuove classi che estendono Data1), semplicemente passando al costruttore della classe Data1 il nome del dato desiderato (non nullo e non vuoto).

Successivamente l'utente può inserire il dato creato nella propria bacheca tramite il metodo “*put*”, identificandosi con la propria password (quella che ha utilizzato per creare la bacheca la prima volta) e scegliendo la categoria alla quale quel determinato dato è associato. Il metodo eseguirà poi tutti i controlli che garantiranno la non presenza di dati con lo stesso nome, o di dati associati a categorie che non sono state mai create.

Il resto dei metodi è praticamente identico alla prima implementazione.

CARATTERISTICHE COMUNI ALLE DUE IMPLEMENTAZIONI

La classe “**Order (1/2)**”, implementa l'interfaccia “Comparator” e contiene l'override del metodo “*compare*” che definisce la regola di ordinamento dei dati rispetto al numero dei likes, utilizzata dal metodo “*getIterator*” per restituire l'iteratore sulla lista ordinata in ordine decrescente.

La **classe main** contiene una batteria di test per testare il comportamento del programma (in entrambe le implementazioni). Tale test deve essere eseguito confrontando attentamente il codice delle Bacheche con quello della classe main, notando, per ogni prova, il lancio dell'eccezione corretta e l'esecuzione delle corrette operazioni da parte dei metodi.

In aggiunta ai metodi previsti ne è stato aggiunto uno in particolare di nome “*status*”, col solo scopo di verificare l'effettivo stato corrente della *bacheca* (1/2), delle sue *categorie*, degli *amici* aggiunti e dei *dati* presenti. Tale metodo sarà ovviamente disponibile nel progetto vero e proprio, soltanto per il proprietario della bacheca (che si autentica ogni volta con la propria password).

Gli altri metodi aggiuntivi sono privati o inutilizzabili, per via dei vincoli di implementazione, o inutili all'utente che intende gestire una bacheca (servono soltanto per la corretta costruzione della bacheca).

Il package del progetto contiene *tre samples di dati* (tre per ogni implementazione ovviamente) che sono utilizzati a scopo illustrativo nella batteria di test.

Per utilizzare i metodi che richiedono il **passaggio di un dato come parametro**, l'utente deve procedere alla creazione di un nuovo dato con lo stesso nome di quello che si trova all'interno della propria bacheca e che intende utilizzare. Ogni metodo procederà poi autonomamente ai vari controlli del caso.

Le interfacce **DataBoard1** e **DataBoard2** contengono tutte le specifiche dei metodi, con tanto di clausole, che ne identificano, per ognuno, i requisiti, le eccezioni lanciate, le strutture modificate e gli effetti prodotti. Contengono inoltre una overview delle implementazioni, caratterizzata da un elemento tipico, da una funzione d'astrazione e da un'invariante di rappresentazione.