

PROGRAMMAZIONE II - A. A. 2019 – 20

Primo Progetto-sessione autunnale

Il progetto ha l'obiettivo di applicare i concetti e le tecniche di programmazione Object-Oriented esaminate durante il corso.

Si richiede di progettare, realizzare e documentare la collezione `DataBoard<E extends Data>`. La collezione `DataBoard<E extends Data>` è un contenitore di oggetti generici che estendono il tipo di dato `Data`. Intuitivamente la collezione si comporta come uno spazio per la memorizzazione e visualizzazioni di dati che possono essere di vario tipo ma che implementano obbligatoriamente il metodo `display()`. La bacheca deve garantire la privacy dei dati fornendo un proprio meccanismo di gestione della condivisione dei dati. Ogni dato presente nella bacheca ha associato la categoria del dato. Il proprietario della bacheca può definire le proprie categorie e stilare una lista di contatti (amici) a cui saranno visibili i dati per ogni tipologia di categoria. I dati condivisi sono visibili solamente in lettura: in particolare i dati possono essere visualizzati dagli amici ma modificati solamente dal proprietario della bacheca. Gli amici possono associare un "like" al contenuto condiviso.

Alcune dell'operazione della bacheca sono definite descritte di seguito.

```
public interface DataBoard<E extends Data>{
    // Crea una categoria di dati
    // se vengono rispettati i controlli di identità
    public void createCategory(String category, String passw);
    // Rimuove una categoria di dati
    // se vengono rispettati i controlli di identità
    public void removeCategory(String category, String passw);
    // Aggiunge un amico ad una categoria di dati
    // se vengono rispettati i controlli di identità
    public void addFriend(String category, String passw, String friend);
    // rimuove un amico da una categoria di dati
    // se vengono rispettati i controlli di identità
    public void removeFriend(String category, String passw, String friend);
    //Inserisce un dato in bacheca
    // se vengono rispettati i controlli di identità
    public boolean
        put(String passw, <E extends Data> dato, String categoria);
    //Restituisce una copia del dato in bacheca
    // se vengono rispettati i controlli di identità
    public E get(String passw, <E extends Data> dato);
    // Rimuove il dato dalla bacheca
    // se vengono rispettati i controlli di identità
```

```

public E remove(String passw, <E extends Data> dato);
//Crea la lista dei dati in bacheca di una determinata categoria
// se vengono rispettati i controlli di identità
// Aggiunge un like a un dato
// se vengono rispettati i controlli di identità
void insertLike(String friend, <E extends Data> data);
public List<E extends Data>
    getDataCategory(String passw, String category);
// restituisce un iteratore (senza remove) che genera tutti i dati in
// bacheca ordinati rispetto al numero di like
// se vengono rispettati i controlli di identità
public Iterator<E> getIterator(String passw);
// restituisce un iteratore (senza remove) che genera tutti i dati in
// bacheca condivisi
public Iterator<E> getFriendIterator(String friend);

// ... altre operazione da definire a scelta
}

```

1. Si definisca la specifica completa e l'implementazione del tipo di dato Data, fornendo le motivazioni delle scelte effettuate.
2. Si definisca la specifica completa del tipo di dato Board<E extends Data>, fornendo le motivazioni delle scelte effettuate.
3. Si definisca l'implementazione del tipo di dato Board<E extends Data>, fornendo almeno due implementazioni che utilizzano differenti strutture di supporto. In entrambi i casi si devono comunque descrivere sia la funzione di astrazione sia l'invariante di rappresentazione. Si discutano le caratteristiche delle due implementazioni.

Per valutare il comportamento dell'implementazioni proposte si realizzi una batteria di test in grado di operare, senza modifiche specifiche, su entrambe le implementazioni proposte.

Modalità di consegna

- Il progetto deve essere svolto e discusso col docente individualmente. Il confronto con colleghi mirante a valutare soluzioni alternative durante la fase di progetto è incoraggiato.
- Il progetto deve essere costituito da
 - i file sorgente contenenti il codice sviluppato e la batteria di test, ove tutto il codice deve essere adeguatamente commentato;
 - una relazione di massimo due pagine, che descrive le principali scelte progettuali ed eventuali istruzioni per eseguire il codice.

- La consegna va fatta inviando per email al Prof. Ferrari o alla Prof. ssa Levi con oggetto “[PR2A] Consegna Progetto Intermedio 1” e “[PR2B] Consegna Progetto Intermedio 1”, rispettivamente .
- Il progetto deve essere consegnato entro il 6 Dicembre 2019.

Altre informazioni

- Per quanto riguarda il progetto, i docenti risponderanno solo a eventuali domande riguardanti l’interpretazione del testo, e non commenteranno soluzioni parziali prima della consegna.