

Esercitazione: Gestione Biblioteca con Ereditarietà ed Eccezioni

Classe 4A INF - Lab. Informatica

March 13, 2025

1 Traccia

Si richiede di implementare un sistema di gestione di una biblioteca con le seguenti classi:

- **Utente**: identificato da nome, cognome e un codice univoco assegnato alla registrazione.
- **Libro**: rappresenta un libro con titolo, autore, codice ISBN e un contatore per il numero di copie disponibili (condiviso tra libri con lo stesso ISBN).
- **Biblioteca**: gestisce un elenco di libri e utenti, permettendo di prestare e restituire libri.
- **Eccezioni**:
 - **LibroNonDisponibileException**: lanciata se un libro non è disponibile per il prestito.
 - **BibliotecaPienaException**: lanciata se si supera il numero massimo di libri in biblioteca.

2 Funzionalità da Implementare

La classe **Biblioteca** deve avere le seguenti funzionalità:

- Un array fisso di libri con capacità massima di 30 libri.
- Un array fisso di utenti con capacità massima di 50 utenti.
- Il metodo `aggiungiLibro(Libro libro)` per aggiungere un libro, incrementando il contatore delle copie disponibili se già presente; altrimenti, aggiunge un nuovo libro se c'è spazio, altrimenti lancia **BibliotecaPienaException**.
- Il metodo `rimuoviLibro(String isbn)` per rimuovere un libro (se esiste).
- Il metodo `modificaLibro(String isbn, String nuovoTitolo, String nuovoAutore)` per modificare i dati di un libro esistente.
- Il metodo `visualizzaLibri()` per stampare l'elenco dei libri presenti in biblioteca.
- Il metodo `prestaLibro(String isbn)` per prestare un libro disponibile, decrementando il contatore delle copie.
- Il metodo `restituisceLibro(String isbn)` per restituire un libro alla biblioteca.

- Il metodo `registraUtente(Utente utente)` per aggiungere un nuovo utente.
- Il metodo `rimuoviUtente(int codiceUtente)` per rimuovere un utente dal sistema.
- Il metodo `modificaUtente(int codiceUtente, String nuovoNome, String nuovoCognome)` per modificare i dati di un utente.
- Il metodo `visualizzaUtenti()` per stampare l'elenco degli utenti registrati in biblioteca.

3 Test e Simulazione

Si richiede di creare una classe di test che esegua le seguenti operazioni:

- Registrazione, modifica ed eliminazione di un utente.
- Inserimento, modifica ed eliminazione di libri (gestendo i casi di duplicati con ISBN uguali o diversi).
- Simulazione di prestiti e restituzioni di libri.
- Verifica della gestione delle eccezioni nei casi limite.
- Visualizzazione degli utenti e dei libri presenti in biblioteca.

4 Considerazioni

Si discuta sui punti deboli del sistema implementato e suggerire possibili miglioramenti da implementare in una seconda release. Alcuni aspetti da considerare:

- Il sistema attuale non tiene conto del fatto che ci potrebbero essere utenti duplicati con lo stesso nome e cognome. Una possibile soluzione potrebbe essere l'introduzione del codice fiscale (CF) per identificare univocamente gli utenti.
- Non esiste un controllo per evitare la rimozione di un libro o di un utente se questo è attualmente in uso (ad esempio, un libro in prestito non dovrebbe poter essere eliminato).
- La gestione dei libri si basa su array statici, che limitano la scalabilità del sistema.
- Si potrebbero implementare meccanismi di ricerca più avanzati per migliorare l'usabilità della biblioteca.