

Esercizio POO Biblioteca

Roberto Borelli

10 Dicembre 2023

Testo dell'Esercizio

Creare un sistema per una piccola biblioteca che tiene traccia dei libri disponibili e dei libri presi in prestito da un unico utente.

Approccio

- **Classi suggerite:** Libro, Utente, Biblioteca.
- **Responsabilità:**
 - **Libro:** Memorizza informazioni di base sul libro e tiene traccia dello stato (disponibile o in prestito).
 - **Utente:** Rappresenta un singolo utente che può prendere in prestito un solo libro alla volta.
 - **Biblioteca:** Gestisce l'inventario dei libri e permette all'utente di prendere in prestito o restituire un libro.

Esercizio POO Biblioteca

Roberto Borelli

10 Dicembre 2023

Suggerimento 1: Design delle Classi

Responsabilità di Utente e Libro

- **Classe Utente:**
 - Si occupa di registrare i libri presi in prestito e restituirli.
 - Metodi come `prendiInPrestito` e `restituisci` devono essere implementati.
 - Gestisce l'aggiunta e la rimozione dei libri presi in prestito nell'array `libriInPrestito`.
- **Classe Libro:**
 - Gestisce le informazioni specifiche di un libro.
 - Mantiene il titolo del libro e il suo stato di disponibilità.
 - Potrebbe includere metodi come `getTitolo` e `setDisponibile`.

Esercizio POO Biblioteca

Roberto Borelli

10 Dicembre 2023

Suggerimento 2: Responsabilità di Biblioteca

Gestione dell'Inventario

- **Classe Biblioteca:**
 - Gestisce l'inventario dei libri.
 - Implementa un metodo per aggiungere un nuovo libro all'inventario (`aggiungiLibro`), controllando la capacità massima.
 - Potrebbe avere un metodo per cercare un libro nell'inventario per titolo (`cercaLibro`).
 - Gestisce la rimozione di un libro dall'inventario, se necessario (`rimuoviLibro`).
- **Utilizzo dei metodi:** I metodi devono garantire la corretta manipolazione dell'inventario e la gestione dei libri presi in prestito dagli utenti. Ad esempio, `Utente.prendiInPrestito` deve aggiornare lo stato del libro come non disponibile e aggiungere il libro all'array `libriInPrestito` dell'utente, mentre `Biblioteca.aggiungiLibro` deve posizionare il libro nell'inventario in una posizione disponibile.

Esercizio POO Biblioteca

Roberto Borelli

10 Dicembre 2023

Suggerimento 3: Implementazione inventario

Aggiunta e Rimozione dei Libri

- **Implementazione con array:**

- **Gestione dell'Inventario:** Utilizza array tradizionali per l'inventario dei libri sia nella classe `Biblioteca` che nell'array `libriInPrestito` della classe `Utente`.
- **Aggiunta e Rimozione dei Libri:** Assicurati che l'aggiunta di un libro all'inventario gestisca correttamente la posizione nell'array e controlli che non superi la capacità massima della biblioteca (`MAX_LIBRI`).
- **Rimozione di un Libro:** Implementa la rimozione di un libro dall'inventario, aggiornando la disponibilità del libro e gestendo gli spazi vuoti nell'array in modo efficiente.
- **Utilizzo dei Metodi:** I metodi devono gestire correttamente l'aggiunta e la rimozione dei libri, evitando errori di indice fuori dalla dimensione dell'array e controllando la capacità massima della biblioteca.

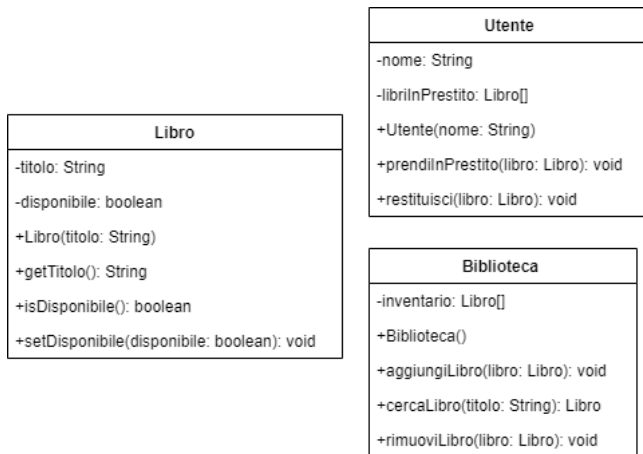
Esercizio POO Biblioteca

Roberto Borelli

10 Dicembre 2023

Sketch della Soluzione dell'Esercizio

Uml



Implementazione in Java

Ecco il codice sorgente Java per l'implementazione delle classi:

```
public class Libro {
    private String titolo;
    private boolean disponibile;

    public Libro(String titolo) {
        this.titolo = titolo;
        this.disponibile = true;
    }

    public String getTitolo() {
        return titolo;
    }

    public boolean isDisponibile() {
        return disponibile;
    }

    public void setDisponibile(boolean disponibile) {
        this.disponibile = disponibile;
    }
}

public class Utente {
    private String nome;
    private Libro[] libriInPrestito;
    private int numLibriInPrestito;
}
```

```

private final int MAX_LIBRI = 5; // Massimo 5 libri in prestito

public Utente(String nome) {
    this.nome = nome;
    this.libriInPrestito = new Libro[MAX_LIBRI];
    this.numLibriInPrestito = 0;
}

// Altri metodi e getter/setter
// ...

}

public class Biblioteca {
    private Libro[] inventario;
    private final int MAX_LIBRI = 100; // Capacità massima della biblioteca

    public Biblioteca() {
        this.inventario = new Libro[MAX_LIBRI];
    }

    // Metodi per gestire l'inventario e le operazioni di prestito e restituzione
    // ...

}

```