

# Sistema di Gestione Biblioteca

L'obiettivo di questo esercizio è consolidare i concetti di **classi**, **ereditarietà**, **ArrayList** e **MySQL**, senza usare **generics**, **classi astratte** e **interfacce**.

---

## Obiettivo

Dovrete sviluppare un **programma Java** che consenta di gestire i libri di una biblioteca, permettendo di:

1. **Aggiungere un libro.**
  2. **Visualizzare tutti i libri disponibili.**
  3. **Ricerca un libro per titolo o ID.**
  4. **Prestare un libro a un lettore.**
  5. **Restituire un libro.**
- 

## Struttura delle Classi

1. **Classe Libro**
    - Contiene le informazioni di un libro:
      - id
      - titolo
      - autore
      - annoPubblicazione
      - disponibile (boolean, indica se il libro è disponibile o in prestito)
  2. **Classe Lettore**
    - Contiene le informazioni di un lettore:
      - id
      - nome
      - cognome
      - email
  3. **Classe GestoreBiblioteca**
    - Usa un ArrayList per gestire i libri.
    - Contiene metodi per:
      - aggiungiLibro(Libro libro)
      - mostraLibri()
      - cercaLibro(String titolo)
      - prestitoLibro(int idLibro, Lettore lettore)
      - restituisciLibro(int idLibro)
  4. **Classe DatabaseManager**
    - Gestisce la connessione a **MySQL**.
    - Contiene metodi per salvare e recuperare i libri dal database.
  5. **Classe Main**
    - Contiene il menu principale per testare il programma.
-

## Compiti da Svolgere

### Creare il Database MySQL

- Creare un database biblioteca.
- Creare una tabella libri con i seguenti campi:

```
CREATE TABLE libri (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  titolo VARCHAR(100),  
  autore VARCHAR(100),  
  anno_publicazione INT,  
  disponibile BOOLEAN  
);
```

- Creare una tabella lettori per memorizzare i dati dei lettori:

```
CREATE TABLE lettori (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nome VARCHAR(50),  
  cognome VARCHAR(50),  
  email VARCHAR(100)  
);
```

---

## Scrivere le Classi in Java

### Classe Libro

```
class Libro {  
  int id;  
  String titolo;  
  String autore;  
  int annoPubblicazione;  
  boolean disponibile;  
  
  public Libro(int id, String titolo, String autore, int annoPubblicazione, boolean disponibile) {  
    this.id = id;  
    this.titolo = titolo;  
    this.autore = autore;  
    this.annoPubblicazione = annoPubblicazione;  
    this.disponibile = disponibile;  
  }  
  
  public void mostraDettagli() {  
    System.out.println("ID: " + id + " | Titolo: " + titolo + " | Autore: " + autore + " | Anno: " +  
annoPubblicazione + " | Disponibile: " + (disponibile ? "Sì" : "No"));  
  }  
}
```

---

### Classe Lettore

```
class Lettore {  
  int id;  
  String nome;  
  String cognome;  
  String email;  
  
  public Lettore(int id, String nome, String cognome, String email) {  
    this.id = id;  
    this.nome = nome;  
    this.cognome = cognome;  
    this.email = email;  
  }  
}
```

---

## Gestire i Libri con ArrayList

### Classe GestoreBiblioteca

```
import java.util.ArrayList;

class GestoreBiblioteca {
    private ArrayList<Libro> catalogo = new ArrayList<>();

    public void aggiungiLibro(Libro libro) {
        catalogo.add(libro);
        System.out.println("Libro aggiunto con successo.");
    }

    public void mostraLibri() {
        if (catalogo.isEmpty()) {
            System.out.println("Nessun libro disponibile.");
        } else {
            for (Libro libro : catalogo) {
                libro.mostraDettagli();
            }
        }
    }

    public void cercaLibro(String titolo) {
        for (Libro libro : catalogo) {
            if (libro.titolo.equalsIgnoreCase(titolo)) {
                libro.mostraDettagli();
                return;
            }
        }
        System.out.println("Libro non trovato.");
    }

    public void prestitoLibro(int idLibro, Lettore lettore) {
        for (Libro libro : catalogo) {
            if (libro.id == idLibro && libro.disponibile) {
                libro.disponibile = false;
                System.out.println("Libro prestato a " + lettore.nome + " " + lettore.cognome);
                return;
            }
        }
        System.out.println("Libro non disponibile.");
    }

    public void restituisciLibro(int idLibro) {
        for (Libro libro : catalogo) {
            if (libro.id == idLibro && !libro.disponibile) {
                libro.disponibile = true;
                System.out.println("Libro restituito.");
                return;
            }
        }
        System.out.println("ID non valido o libro già disponibile.");
    }
}
```

---

## Salvare e Recuperare i Libri con MySQL

### Classe DatabaseManager

```
import java.sql.*;

class DatabaseManager {
    private static final String URL = "jdbc:mysql://localhost:3306/biblioteca";
    private static final String USER = "root";
    private static final String PASSWORD = "";

    public static Connection connettiDB() throws SQLException {
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }

    public void salvaLibro(Libro libro) {
        try (Connection conn = connettiDB()) {
            String query = "INSERT INTO libri (titolo, autore, anno_pubblicazione, disponibile) VALUES (?, ?, ?, ?)";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setString(1, libro.titolo);
            stmt.setString(2, libro.autore);
            stmt.setInt(3, libro.annoPubblicazione);
            stmt.setBoolean(4, libro.disponibile);

            stmt.executeUpdate();
            System.out.println("Libro salvato nel database.");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

---

## Implementare il Menu di Test

### Classe Main

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        GestoreBiblioteca gestore = new GestoreBiblioteca();
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\n--- Sistema di Gestione Biblioteca ---");
            System.out.println("1. Aggiungi libro");
            System.out.println("2. Mostra tutti i libri");
            System.out.println("3. Cerca un libro");
            System.out.println("4. Prestito libro");
            System.out.println("5. Restituisci libro");
            System.out.println("6. Esci");
            System.out.print("Scelta: ");

            int scelta = scanner.nextInt();
            scanner.nextLine();

            switch (scelta) {
                case 1:
                    System.out.println("Inserisci titolo: ");
                    String titolo = scanner.nextLine();
                    System.out.println("Inserisci autore: ");
                    String autore = scanner.nextLine();
                    System.out.println("Inserisci anno pubblicazione: ");
```

```
        int anno = scanner.nextInt();
        Libro libro = new Libro(gestore.catalogo.size() + 1, titolo, autore, anno, true);
        gestore.aggiungiLibro(libro);
        break;
    case 2:
        gestore.mostraLibri();
        break;
    case 3:
        System.out.println("Inserisci titolo libro:");
        gestore.cercaLibro(scanner.nextLine());
        break;
    case 4:
        System.out.println("Inserisci ID libro:");
        gestore.prestitoLibro(scanner.nextInt(), new Lettore(1, "Mario", "Rossi", "mario@email.com"));
        break;
    case 5:
        System.out.println("Inserisci ID libro:");
        gestore.restituisceLibro(scanner.nextInt());
        break;
    case 6:
        System.exit(0);
    }
}
}
```