

```

// Fondamenti di Informatica/Informatica Grafica
// Soluzione del compito del 21/03/2003
// per semplicità si è deciso di riportare in un unico file il testo
// delle soluzioni degli esercizi 1, 2 e 3.
// ogni classe poteva appartenere ad un file distinto: Libro.java,
// Utente.java, Prestito.java, Biblioteca.java e Applicazione.java

import fiji.io.SimpleReader;

public class Applicazione {

    public static void main(String[] args) {

        // inizializzazione della biblioteca con utenti, libri e prestiti
        // non richiesta
        Biblioteca.inizializza();

        // inizio del compito
        SimpleReader in = new SimpleReader();

        String nome, cognome, autore, titolo;
        int codice;
        Libro libro;
        Utente utente;

        System.out.println("Inserisci il Codice utente: ");
        codice = in.readInt();

        System.out.println("Inserisci il Nome utente: ");
        nome = in.readString();

        System.out.println("Inserisci il Cognome utente: ");
        cognome = in.readString();

        utente = new Utente(codice, nome, cognome);

        System.out.println("Inserisci Autore: ");
        autore = in.readString();

        System.out.println("Inserisci Titolo: ");
        titolo = in.readString();

        libro = Biblioteca.cercaLibro(autore, titolo);
        if (libro == null) {
            System.out.println("Libro non censito in biblioteca!");
        } else {
            System.out.println("Libro: codice " + libro.getISBN() + " editore " +
libro.getEditore());
            if (Biblioteca.inPrestito(libro)) {
                System.out.println("Libro non disponibile");
            } else {
                Biblioteca.inserisciPrestito(utente, libro);
                System.out.println("Libro preso a Prestito!");
            }
        }
    }
}

public class Libro {

```

```

private String ISBN;
private String autore;
private String titolo;
private String editore;

/**
 * Costruttore con parametri
 */
public Libro(String ISBN, String autore, String titolo, String editore) {
    this.ISBN = ISBN;
    this.autore = autore;
    this.titolo = titolo;
    this.editore = editore;
}

/**
 * equals
 */
public boolean equals(Libro l) {
    return (l.getISBN().equals(ISBN) &&
            l.getEditore().equals(editore) &&
            l.getTitolo().equals(titolo) &&
            l.getAutore().equals(autore));
}

public String toString() {
    return(autore + " - " + titolo);
}

/**
 * Gets the ISBN
 */
public String getISBN() {
    return ISBN;
}

/**
 * Gets the autore
 */
public String getAutore() {
    return autore;
}

/**
 * Gets the titolo
 */
public String getTitolo() {
    return titolo;
}

/**
 * Gets the editore
 */
public String getEditore() {
    return editore;
}
}

public class Utente {

```

```

private int codice;
private String nome;
private String cognome;

/**
 * Costruttore con parametri
 */
public Utente(int codice, String nome, String cognome) {
    this.codice = codice;
    this.nome = nome;
    this.cognome = cognome;
}

/**
 * equals
 */
public boolean equals(Utente u) {
    return (u.getCodice() == codice &&
            u.getNome().equals(nome) &&
            u.getCognome().equals(cognome));
}

/**
 * Gets the codice
 */
public int getCodice() {
    return codice;
}

/**
 * Gets the nome
 */
public String getNome() {
    return nome;
}

/**
 * Gets the cognome
 */
public String getCognome() {
    return cognome;
}
}

public class Prestito {

    private static int codice = 0;
    private Utente utente;
    private Libro libro;

    public Prestito(Utente utente, Libro libro) {
        this.utente = utente;
        this.libro = libro;

        // generazione del codice progressivo
        codice = codice + 1;
    }
}

```

```

    /**
     * equals
     */
    public boolean equals(Prestito p) {
        return (p.getCodice() == codice &&
                p.getLibro().equals(libro) &&
                p.getUtente().equals(utente));
    }

    /**
     * Gets the codice
     */
    public static int getCodice() {
        return codice;
    }

    /**
     * Gets the utente
     */
    public Utente getUtente() {
        return utente;
    }

    /**
     * Gets the libro
     */
    public Libro getLibro() {
        return libro;
    }

}

public class Biblioteca {

    public static final int MAX_LIBRI = 10000;
    public static final int MAX_UTENTI = 1000;
    public static final int MAX_PRESTITI = 200;

    public static String nome;
    private static Libro[] libri = new Libro[MAX_LIBRI];
    private static Utente[] utenti = new Utente[MAX_UTENTI];
    private static Prestito[] prestiti = new Prestito[MAX_PRESTITI];

    private static int numLibri = 0;
    private static int numUtenti = 0;
    private static int numPrestiti = 0;

    /**
     * Gets the nome
     */
    public static String getNome() {
        return nome;
    }

    /**
     * inserisce un prestito nell'Array se vi e' posto
     */
    public static void inserisciPrestito(Utente u, Libro l) {
        if (numPrestiti < MAX_PRESTITI) {
            prestiti[numPrestiti] = new Prestito(u,l);
            numPrestiti++;
        }
    }
}

```

```

        }
    }

/***
 * Elimina un prestito rimuovendolo dall'Array
 */
public static void eliminaPrestito(Prestito p) {
    boolean found = false;
    int i=0;
    while(i < numPrestiti && !found) {
        if (prestiti[i].equals(p))
            found = true;
        i++;
    }
    if (found){
        while(i<numPrestiti)
            prestiti[i-1]=prestiti[i];
        prestiti[i-1]=null;
        numPrestiti--;
    } else {
        System.out.println("il prestito "+ p + " non e' stato trovato!");
    }
}

/***
 * conteggio dei prestiti per utente
 */
public static int contaPrestitiPerUtente(Utente u) {
    int ret = 0;
    for (int i=0; i < numPrestiti;i++) {
        if (prestiti[i].getUtente().equals(u)) {
            ret++;
        }
    }
    return ret;
}

/***
 * controlla lo stato di un libro
 */
public static boolean inPrestito(Libro l) {
    boolean found = false;
    for (int i=0;i < numPrestiti && !found;i++) {
        if (prestiti[i].getLibro().equals(l)) {
            found = true;
        }
    }
    return found;
}

/***
 * ricerca e restituisce un libro dati autore e titolo
 */
public static Libro cercaLibro(String autore, String titolo) {
    boolean found = false;
    Libro ret = null;
    for (int i=0; i < numLibri && ret == null;i++) {
        if (libri[i].getAutore().equals(autore) &&
            libri[i].getTitolo().equals(titolo)) {
            ret = libri[i];
        }
    }
}

```

```
    return ret;
}

// inizializzazione della biblioteca con utenti, libri e prestiti
// non richiesta
public static void inizializza() {
    numLibri = 3;
    libri[0] = new Libro("ISBN1", "autore1", "titolo1", "editore1");
    libri[1] = new Libro("ISBN2", "autore2", "titolo2", "editore2");
    libri[2] = new Libro("ISBN3", "autore3", "titolo3", "editore3");

    numUtenti = 2;
    utenti[0] = new Utente(123, "Fabio", "Rossi");
    utenti[1] = new Utente(456, "Giulia", "Bianchi");

    numPrestiti = 2;
    prestiti[0] = new Prestito(utenti[0], libri[0]);
    prestiti[1] = new Prestito(utenti[1], libri[2]);
}

}
```