

```
/*  
Classe Book
```

```
*/
```

```
package com.company;
```

```
public class Book {  
    private int numPages;  
    private String title; //dichiarazione degli attributi  
    private String genre;  
  
    public Book(int nPages, String bookTitle, String bookGenre){  
        this.genre = bookGenre;  
        this.numPages = nPages;  
        this.title = bookTitle;  
    }  
  
    public int getNumPages() {  
        return numPages;  
    }  
  
    public String getTitle() {  
        return title;  
    }  
  
    public String getGenre() {  
        return genre;  
    }  
  
    public String showInfo(){  
        return "Book title: " + this.title + "\nBook pages: " + this.numPages + "\nBook genre: " + this.genre;  
    }  
  
    public void setNumPages(int num){  
        this.numPages = num;  
    }  
  
    public void setGenre(String genre) {  
        this.genre = genre;  
    }  
  
    public void setTitle(String title) {  
        this.title = title;  
    }  
}
```

/*

CLASSE TEST,

Questo programma consente di andare ad operare su uno scaffale con una selezione multipla aggiungendo libri e vedendone le informazioni principali come:

- il numero delle pagine

- il titolo

- il genere del libro

Consente inoltre di vedere le informazioni di tutti i libri o di un singolo libro e di ordinarli per numero di pagine attraverso un algoritmo di sorting

*/

```
package com.company;
```

```
import java.util.Scanner;
```

```
public class Test {
```

```
    static Scanner scan = new Scanner(System.in);
```

```
    public static void main(String[] args) {
```

```
        Shelf shelf = new Shelf(100, 3); //Istanza della classe scaffale
```

```
        int r;
```

```
        do{
```

```
            System.out.println("1. Add books to the shelf\n" +  
                "2. Show how many books are in the shelf\n" +  
                "3. Show a specified books info\n" +  
                "4. Sort the shelf\n" +  
                "5. Show every books info\n" +  
                "6. Exit");
```

```
            r = scan.nextInt();
```

```
            switch (r){
```

```
                case 1:
```

```
                    if(!shelf.isFull()){
```

```
                        System.out.print("Book title: ");
```

```
                        String bookTitle = scan.next();
```

```
                        System.out.print("Book pages: ");
```

```
                        int bookPages = scan.nextInt();
```

```
                        System.out.print("Book genre: ");
```

```
                        String bookGenre = scan.next();
```

```
                        shelf.addBook(new Book(bookPages, bookTitle, bookGenre));
```

```
                        System.out.println("Book successfully added");
```

```
                    }
```

```
                else{
```

```
                    System.out.println("Shelf is full");
```

```
                }
```

```
                break;
```

```
                case 2:
```

```
                    System.out.println("There are " + shelf.getNumBooks() + " books in the shelf");
```

```
                    break;
```

```
                case 3:
```

```
                    System.out.print("Enter the title of book which you want to see the info: ");
```

```
                    String title = scan.next();
```

```
                    shelf.search(title);
```

```
                    break;
```

```
                case 4:
```

```
                    shelf.sortBooks();
```

```
                    System.out.println("Shelf successfully sorted!");
```

```
                    break;
```

```
        case 5:  
            shelf.showInfo();  
            break;  
        }  
    }while ( r != 6);  
}
```

```
/*
Classe Shelf.
In questa classe è contenuto l'array di oggetti books di tipo Book
L'array di oggetti viene inizializzato nel costruttore con dimensione max data come parametro
E' possibile aggiungere un libro finchè c'è spazio, ordinare i libri in base alle pagine
*/
```

```
package com.company;
```

```
public class Shelf {
    private int numBooks = 0;
    private int id;
    private boolean isFull = false;
    private Book[] books;    //Dichiarazione array di oggetti di tipo Book

    public Shelf(int identifier, int nMax){
        this.id = identifier;
        books = new Book[nMax];    //inizializzazione dell'array con dimensione nMax
    }

    public void addBook(Book book){
        if(numBooks < books.length){
            books[numBooks] = book;    //controlla se c'è abbastanza spazio per inserire un nuovo oggetto libro e controlla
se è pieno
            numBooks++;
        }
        else{
            isFull = true;
        }
    }

    public boolean isFull(){
        return isFull;
    }

    public int getNumBooks(){
        return numBooks;
    }

    public int getId() {
        return id;
    }

    public Book[] getBooks() {
        return books;
    }

    public void sortBooks(){    //Algoritmo di sort Selection Sort
        for(int i = 0; i < numBooks; i++){
            for(int j = 0; j < numBooks; j++){
                if(books[i].getNumPages() > books[j].getNumPages()){
                    int pages1, pages2;
                    String title1, genres1, title2, genres2;    //VARIABILI DI APPOGGIO PER FARE IL SELECTION SORT

                    pages1 = books[i].getNumPages();
                    pages2 = books[j].getNumPages();
                    title1 = books[i].getTitle();
                    title2 = books[j].getTitle();
                    genres1 = books[i].getGenre();
```

```

        genres2 = books[j].getGenre();

        books[i].setNumPages(pages2);
        books[j].setNumPages(pages1);
        books[i].setTitle(title2);
        books[j].setTitle(title1);
        books[i].setGenre(genres2);
        books[j].setGenre(genres1);
    }
}
}

public void showInfo(){
    if(numBooks != 0){
        for(int i = 0; i < numBooks; i++){
            System.out.println(books[i].showInfo());
        }
    }
    else{
        System.out.println("Shelf's empty");
    }
}

public void search(String title){
    int i = 0;
    boolean hasBeenFound = false;
    while(!hasBeenFound || i <= numBooks){
        if(books[i].getTitle().equals(title)){ //Algoritmo per cercare un elemento in base al titolo dato come
                                                //parametro della funzione
            hasBeenFound = true;
        }
        else{
            i++;
        }
    }
    System.out.println(books[i].showInfo());
}
}

```