

```
public class Book
{
    private int id;
    private String title;
    private String author;
    private double price;

    public Book(int pID, String pTitle, String pAuthor, double pPrice)
    {
        this.id = pID;
        this.title = pTitle;
        this.author = pAuthor;
        this.price = pPrice;
    }

    public String getTitle()
    {
        return title;
    }

    public String getAuthor()
    {
        return author;
    }

    public double getPrice()
    {
        return price;
    }

    public int getID()
    {
        return id;
    }

    public String toString()
    {
        return "[" + id + "] -> " + author + " - " + title + " (" + price + "€)";
    }
}
```

```
import java.util.ArrayList;
public class Library
{
    private ArrayList<Book> alBooks = new ArrayList<>();

    public void addBook(Book pBook)
    {
        alBooks.add(pBook);
    }

    public int countBooks()
    {
        return alBooks.size();
    }

    public Object[] toArray()
    {
        return alBooks.toArray();
    }

    public Double searchCheapestPrice()
    {
        Book b;

        if (alBooks.size() == 0)
            return null;

        double min = alBooks.get(0).getPrice();
        for (int i = 1; i < alBooks.size(); i++)
        {
            b = alBooks.get(i);
            if (b.getPrice() < min)
                min = b.getPrice();
        }

        return min;
    }

    public String searchAuthorOfBiggestID()
    {
        Book b;

        if (alBooks.size() == 0)
            return null;

        Book max = alBooks.get(0);
        for (int i = 1; i < alBooks.size(); i++)
        {
            b = alBooks.get(i);
            if (b.getID() > max.getID())
                max = b;
        }

        return max.getAuthor();
    }

    public Book searchFirstBookOfAuthor(String pAuthor)
    {
        int i = 0;
        boolean found = false;
        while (!found && (i < alBooks.size()))
        {
            if (alBooks.get(i).getAuthor().equals(pAuthor))
                found = true;
            else
                i++;
        }

        if (found)
            return alBooks.get(i);
        else
            return null;
    }

    public int countBooksOfAuthor(String pAuthor)
    {
        int count = 0;
        for (int i = 0; i < alBooks.size(); i++)
        {
            if (alBooks.get(i).getAuthor().equals(pAuthor))
                count++;
        }
        return count;
    }

    // ... suite page suivante ...
}
```

```
public void sortByTitle()
{
    for (int i = 0; i < alBooks.size() - 1; i++)
    {
        String max = alBooks.get(i).getTitle();
        int posMax = i;
        for (int j = i + 1; j < alBooks.size(); j++)
        {
            if (alBooks.get(j).getTitle().compareTo(max) > 0)
            {
                max = alBooks.get(j).getTitle();
                posMax = j;
            }
        }
        if (i != posMax)
        {
            Book temp = alBooks.get(i);
            alBooks.set(i, alBooks.get(posMax));
            alBooks.set(posMax, temp);
        }
    }
}

public void deleteBooks(double price)
{
    for (int i=alBooks.size()-1; i>=0; i--)
    {
        if (alBooks.get(i).getPrice() < price)
            alBooks.remove(i);
    }
}
}
```