

```

public interface Sellable {

    public double sellingPrice();
    public void display();
}

public abstract class Book implements Sellable{

    protected String title;
    protected int nbPages;

    Book(String title,int nbPgs)
    {
        this.title = title;
        nbPages = nbPgs;
    }

    Book(Book b)
    {
        title = b.title;
        nbPages = b.nbPages;
    }

    public void display() {
        System.out.print(title + nbPages);
    }

    public String getTitle() {
        return title;
    }

    public int getNbPages() {
        return nbPages;
    }
}

public class TextBook extends Book {

    private int year;

    TextBook(String title, int nbPgs, int year)
    {
        super(title,nbPgs);
        this.year = year;
    }

    TextBook(TextBook t) {

```

```

        super(t);
    }

    public double sellingPrice()
    {
        return nbPages * 2 + (year%200)*30;
    }

    public int getYear()
    {
        return year;
    }
}

public class LiteratureBook extends Book {

    private String type;

    LiteratureBook(String title, int nbPgs, String type)
    {
        super(title,nbPgs);
        this.type = type;
    }

    LiteratureBook(LiteratureBook l)
    {
        super(l);
        type = l.type;
    }

    public double sellingPrice()
    {
        return nbPages * 2 + 2000 * 30;
    }

    public String getType() {
        return type;
    }
}

```

```

public class Store {

    private String name;
    private Book[] arBooks;
    private int nb;

    Store(String name, int size)
    {
        this.name = name;
        arBooks = new Book[size];
        nb = 0;
    }

    public void addBook(Book b)
    {
        if(nb >= arBooks.length)
            return;

        if(b instanceof TextBook)
            arBooks[nb] = new TextBook( (TextBook) b );
        else
            arBooks[nb] = new LiteratureBook( (LiteratureBook) b );
        nb++;
    }

    public int addBooks(Book[] b)
    {
        int count = 0;
        for(int i=0; i< b.length; i++)
        {
            if(nb < arBooks.length)
            {
                addBook(b[i]);
                count++;
            }
            else
                return count;
        }

        return count;
    }

    public void diplayAll()
    {
        for(int i=0; i<nb ; i++)
        {
            arBooks[i].display();
        }
    }

    public int countTextBooks()

```

```

{
    int count = 0;
    for(int i =0; i<nb; i++)
        if(arBooks[i] instanceof TextBook)
            count++;
    return count;
}

public double averagePriceOfTextBooks(double s)
{
    int count = 0;
    double sum = 0;

    for(int i=0; i<nb; i++)
        if(arBooks[i] instanceof TextBook)
            if(arBooks[i].sellingPrice() > s)
            {
                count++;
                sum+= arBooks[i].sellingPrice();
            }

    if( count ==0) return 0;

    return sum/count;
}

public LiteratureBook getLiteratureBook(String t)
{
    for(int i=0; i<nb; i++)
    {
        if(arBooks[i] instanceof LiteratureBook)
            if(arBooks[i].getTitle().equals(t))
                return (LiteratureBook) arBooks[i];
    }

    throw new IllegalArgumentException("No Literature Book with title "+t+"
found");
}

public TextBook[] getTextBooks(int y)
{
    TextBook[] t = new TextBook[nb];
    int j=0;

    for(int i=0; i<nb; i++)
    {
        if(arBooks[i] instanceof TextBook)
        {
            TextBook x = (TextBook) arBooks[i];
            if(x.getYear() > y)
                t[j++] = x;
        }
    }
}

```

```
        return t;
    }

    public void splitBooks(TextBook[] tb, LiteratureBook[] lb)
    {
        int j = 0;
        int k = 0;

        for(int i=0; i<nb; i++)
        {
            if(arBooks[i] instanceof TextBook)
                tb[j++] = (TextBook) arBooks[i];
            else
                lb[k++] = (LiteratureBook) arBooks[i];
        }
    }
}
```