

# Application de gestion de bibliothèque

Réaliser par: Nouredine Yassine

Voici le code de chaque fichier utilisé :

## 1. InterfaceAffiche.java

```
package examen_finale;  
public interface InterfaceAffiche {  
    void affichageDetails();  
}
```

## 2. LivrePapier.java

```
package examen_finale;  
public class LivrePapier implements InterfaceAffiche {  
    private String titre;  
    private String auteur;  
    private String datePublication;  
    public LivrePapier(String titre, String auteur, String datePublication) {  
        this.titre = titre;  
        this.auteur = auteur;  
        this.datePublication = datePublication;  
    }  
    public String getTitre() {  
        return titre;  
    }  
    public String getAuteur() {  
        return auteur;  
    }  
    public String getDatePublication() {  
        return datePublication;  
    }  
  
    public void affichageDetails() {  
        System.out.println("Le titre du livre papier : " + titre + ", d'auteur : " +  
auteur + ", et la date de publication : " + datePublication);  
    }  
}
```

```
}  
}
```

### 3. LivreNumerique.java

```
package examen_finale;  
public class LivreNumerique implements InterfaceAffiche {  
    private String titre;  
    private String auteur;  
    private String datePublication;  
    private double tailleFichier;  
    public LivreNumerique(String titre, String auteur, String  
datePublication, double tailleFichier) {  
        this.titre = titre;  
        this.auteur = auteur;  
        this.datePublication = datePublication;  
        this.tailleFichier = tailleFichier;  
    }  
    public String getTitre() {  
        return titre;  
    }  
    public String getAuteur() {  
        return auteur;  
    }  
    public String getDatePublication() {  
        return datePublication;  
    }  
    public double getTailleFichier() {  
        return tailleFichier;  
    }  
    public void affichageDetails() {  
        System.out.println("Le titre du livre numérique est : " + titre + ", de  
l'auteur : " + auteur + ", date de publication : " + datePublication + ", taille  
: " + tailleFichier + " Mo");  
    }  
}
```

#### 4. Stockage.java

```
package examen_finale;
import java.io.*;
import java.util.*;
public class Stockage {
    private List<InterfaceAffiche> livres;
    public Stockage() {
        livres = new ArrayList<>();
    }
    public void ajouterLivre(InterfaceAffiche livre) {
        livres.add(livre);
    }
    public List<InterfaceAffiche> getLivres() {
        return livres;
    }
    public void sauvegarderLivres(String nomFichier) {
        try {
            PrintWriter sortie = new PrintWriter(new FileWriter(nomFichier));
            for (InterfaceAffiche livre : livres) {
                if (livre instanceof LivrePapier) {
                    LivrePapier lp = (LivrePapier) livre;
                    sortie.println("Papier;" + lp.getTitre() + ";" + lp.getAuteur() +
";" + lp.getDatePublication());
                } else if (livre instanceof LivreNumerique) {
                    LivreNumerique ln = (LivreNumerique) livre;
                    sortie.println("Numerique;" + ln.getTitre() + ";" +
ln.getAuteur() + ";" + ln.getDatePublication() + ";" + ln.getTailleFichier());
                }
            }
            sortie.close();
            System.out.println("Les livres ont été sauvegardés dans le fichier
" + nomFichier);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```
}
```

## 5. Interface.java

```
package examen_finale;
import javax.swing.*;
import java.awt.*;
public class Interface extends JFrame {
    private Stockage stockage; // Stockage avec une majuscule
    private JTextField titreField, auteurField, dateField, tailleField;
    private JTextArea affichageArea;
    private DefaultListModel<String> livreListModel;
    private JList<String> livreList;
    public Interface() {
        stockage = new Stockage(); // Instanciation correcte de Stockage
        setTitle("Gestion de Bibliothèque");
        setSize(700, 400);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        JPanel inputPanel = new JPanel(new GridLayout(5, 2));
        inputPanel.add(new JLabel("Titre:"));
        titreField = new JTextField();
        inputPanel.add(titreField);
        inputPanel.add(new JLabel("Auteur:"));
        auteurField = new JTextField();
        inputPanel.add(auteurField);
        inputPanel.add(new JLabel("Date:"));
        dateField = new JTextField();
        inputPanel.add(dateField);
        inputPanel.add(new JLabel("Taille fichier (Mo):"));
        tailleField = new JTextField();
        inputPanel.add(tailleField);
        livreListModel = new DefaultListModel<>();
        livreList = new JList<>(livreListModel);
        JScrollPane scrollPane = new JScrollPane(livreList);
        JPanel buttonPanel = new JPanel();
        JButton ajouterButton = new JButton("Ajouter");
        ajouterButton.addActionListener(e -> ajouterLivre());
    }
}
```

```

buttonPanel.add(ajouterButton);
JButton afficherButton = new JButton("Afficher");
afficherButton.addActionListener(e -> afficherLivreDetails());
buttonPanel.add(afficherButton);
JButton sauvegarderButton = new JButton("Sauvegarder");
sauvegarderButton.addActionListener(e -> {
    stockage.sauvegarderLivres("test.txt");
    affichageArea.append("Livres sauvegardés dans test.txt.\n");
});
buttonPanel.add(sauvegarderButton);
affichageArea = new JTextArea();
affichageArea.setEditable(false);
add(inputPanel, BorderLayout.NORTH);
add(buttonPanel, BorderLayout.CENTER);
add(scrollPane, BorderLayout.WEST);
add(new JScrollPane(affichageArea), BorderLayout.SOUTH);
setVisible(true);
}

private void ajouterLivre() {
    String titre = titreField.getText();
    String auteur = auteurField.getText();
    String date = dateField.getText();
    String taille = tailleField.getText();
    if (!taille.isEmpty()) {
        try {
            double tailleFichier = Double.parseDouble(taille);
            stockage.ajouterLivre(new LivreNumerique(titre, auteur, date,
tailleFichier));
            livreListModel.addElement(titre + " (Numérique)");
        } catch (NumberFormatException ex) {
            affichageArea.append("Erreur: Taille du fichier invalide.\n");
        }
    } else {
        stockage.ajouterLivre(new LivrePapier(titre, auteur, date));
        livreListModel.addElement(titre + " (Papier)");
    }
    titreField.setText("");
}

```

```

    auteurField.setText("");
    dateField.setText("");
    tailleField.setText("");
}
private void afficherLivresDetails() {
    String selectedLivre = livreList.getSelectedValue();
    if (selectedLivre != null) {
        for (InterfaceAffiche livre : stockage.getLivres()) {
            if (livre instanceof LivrePapier &&
selectedLivre.contains("Papier")) {
                LivrePapier lp = (LivrePapier) livre;
                afficherDetailsFenetre(lp.getTitre(), lp.getAuteur(),
lp.getDatePublication(), "Papier");
            } else if (livre instanceof LivreNumerique &&
selectedLivre.contains("Numérique")) {
                LivreNumerique ln = (LivreNumerique) livre;
                afficherDetailsFenetre(ln.getTitre(), ln.getAuteur(),
ln.getDatePublication(), "Numérique", ln.getTailleFichier());
            }
        }
    }
}
private void afficherDetailsFenetre(String titre, String auteur, String
date, String type) {
    JFrame detailsFrame = new JFrame("Détails du Livre");
    detailsFrame.setSize(400, 300);

    detailsFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE
);
    JTextArea detailsArea = new JTextArea();
    detailsArea.setText("Titre: " + titre + "\nAuteur: " + auteur + "\nDate
de Publication: " + date + "\nType: " + type);
    detailsArea.setEditable(false);
    detailsFrame.add(new JScrollPane(detailsArea));
    detailsFrame.setVisible(true);
}

```

```

    private void afficherDetailsFenetre(String titre, String auteur, String
date, String type, double taille) {
        JFrame detailsFrame = new JFrame("Détails du Livre");
        detailsFrame.setSize(400, 300);

detailsFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE
);
        JTextArea detailsArea = new JTextArea();
        detailsArea.setText("Titre: " + titre + "\nAuteur: " + auteur + "\nDate
de Publication: " + date + "\nType: " + type + "\nTaille: " + taille + " Mo");
        detailsArea.setEditable(false);
        detailsFrame.add(new JScrollPane(detailsArea));
        detailsFrame.setVisible(true);
    }
    public static void main(String[] args) {
        new Interface();
    }
}

```

## 6. Fichier de sauvegarde ( **test.txt**):

dans lequel les données sont enregistrées.

voici la fenêtre:

Gestion de Bibliothèque

Titre:

Auteur:

Date:

Taille fichier (Mo):

Art (Papier)  
EHTP (Numérique)

Ajouter

Afficher

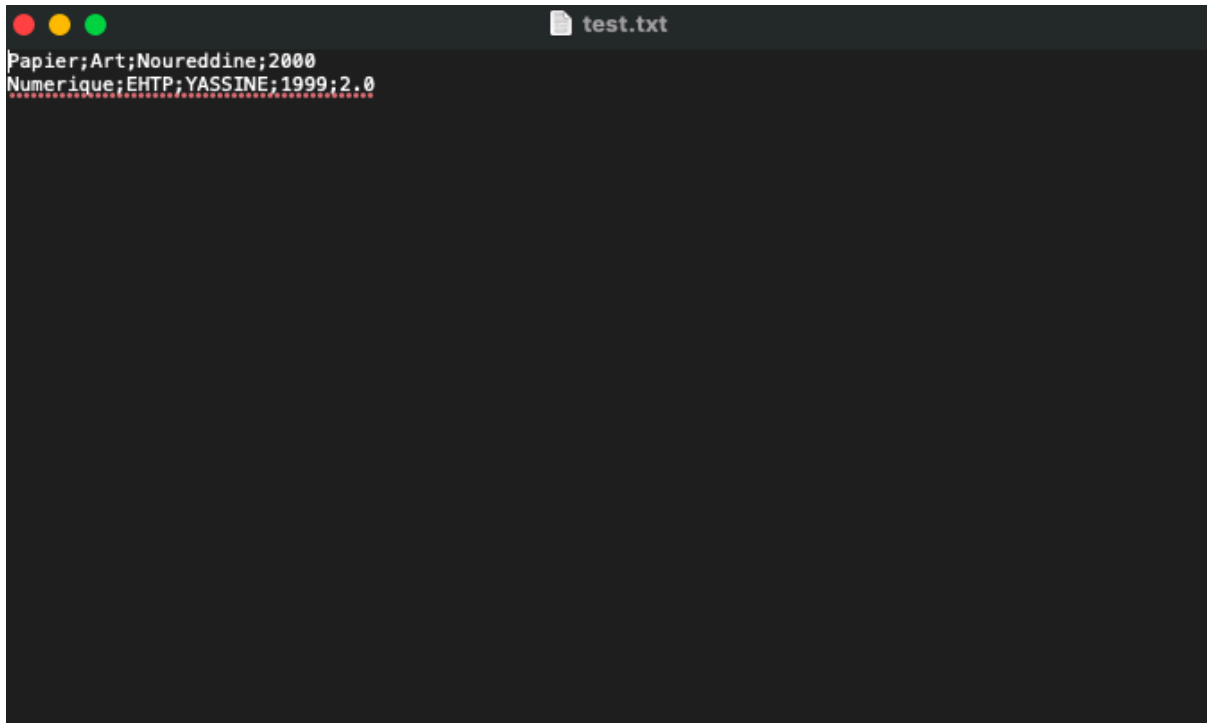
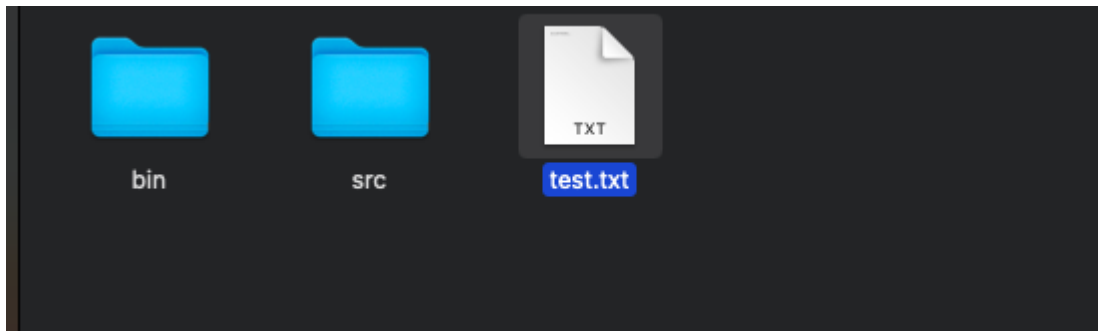
Sauvegarder

Détails du Livre

Titre: EHTP  
Auteur: YASSINE  
Date de Publication: 1999  
Type: Numérique  
Taille: 2.0 Mo



Voici le fichier

A screenshot of a terminal window titled 'test.txt'. It displays two lines of text: 'Papier;Art;Noureddine;2000' and 'Numerique;EHTP;YASSINE;1999;2.0'. The second line is underlined with a red dotted line.

```
Papier;Art;Noureddine;2000
Numerique;EHTP;YASSINE;1999;2.0
```