

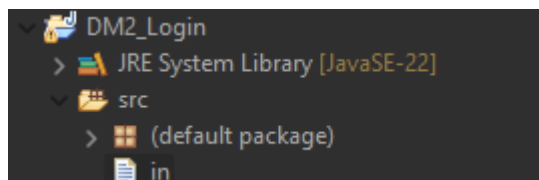
DEMIR  
Umut  
2BTS SL

## DM2 : Les entrées et sorties & interface graphique en JAVA

### Exercice1 :

Tout d'abord, je vais créer un répertoire à la racine du dossier `/src/` nommé `in.txt` et y introduire les lignes suivantes :

```
User1 passUser1
User2 passUser2
User3 passUser3
```



Ensuite, je vais créer une app pour copier le fichier, voici le code que j'y ai introduit, le code suivant :

```
import java.io.*;
public class Exo1 {
    public static void copyFile(String source, String destination) throws
IOException {
        try (BufferedReader reader = new BufferedReader(new
FileReader(source));
            BufferedWriter writer = new BufferedWriter(new
FileWriter(destination))) {
            String line;
            while ((line = reader.readLine()) != null) {
                writer.write(line);
                writer.newLine();
            }
        }
    }
}
```

### Exercice2 :

Ensuite, je vais écrire un programme permettant de lire le fichier 'in.txt', qui va également saisir deux chaînes de caractères "l'identifiant" et le "mot de passe" et qui va devoir vérifier l'existence de ces deux chaînes de caractères, voici le programme :

```
import java.io.*;
public class Exo2 {
    public static boolean authenticate(String login, String password, String
filePath) throws IOException {
        try (BufferedReader reader = new BufferedReader(new
FileReader(filePath))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] credentials = line.split(" ");
                if (credentials.length == 2 && credentials[0].equals(login) &&
credentials[1].equals(password)) {
                    return true;
                }
            }
        }
        return false;
    }
}
```

### Exercice3 :

Ensuite, comme dans l'exercice 2 je vais écrire un programme permettant de lire le fichier 'in.txt' et qui va compter le nombre des lignes, des chaînes et des caractères du fichier lu:

```
import java.io.*;
import java.util.StringTokenizer;
public class Exo3 {
    public static void countFileContent(String filePath) throws IOException
{
        int lineCount = 0;
        int wordCount = 0;
```

```

    int charCount = 0;
    try (BufferedReader reader = new BufferedReader(new
FileReader(filePath))) {
        String line;
        while ((line = reader.readLine()) != null) {
            lineCount++;
            charCount += line.length();
            StringTokenizer tokenizer = new StringTokenizer(line);
            wordCount += tokenizer.countTokens();
        }
    }
    System.out.println("Nombre de lignes : " + lineCount);
    System.out.println("Nombre de mots : " + wordCount);
    System.out.println("Nombre de caractères : " + charCount);
}
}

```

#### **Exercice4 :**

Pour finir, je vais reprendre l'exercice 2 et ajouter une interface de connexion:

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
public class Exo4 {
    public static void createLoginGUI(String filePath) {
        JFrame frame = new JFrame("Authentification");
        frame.setSize(300, 150);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(3, 2));
        JLabel userLabel = new JLabel("Login:");
        JTextField userText = new JTextField();
        JLabel passwordLabel = new JLabel("Mot de passe:");
        JPasswordField passwordText = new JPasswordField();
    }
}

```

```

JButton loginButton = new JButton("Se connecter");
JLabel resultLabel = new JLabel("");
panel.add(userLabel);
panel.add(userText);
panel.add(passwordLabel);
panel.add(passwordText);
panel.add(loginButton);
panel.add(resultLabel);
frame.add(panel);
frame.setVisible(true);
loginButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String login = userText.getText();
        String password = new String(passwordText.getPassword());
        try {
            if (Exo2.authenticate(login, password, filePath)) {
                resultLabel.setText("Authentification réussie");
            } else {
                resultLabel.setText("PB Authentification");
            }
        } catch (IOException ex) {
            resultLabel.setText("Erreur de lecture du fichier");
        }
    }
});
}
}

```

Pour exécuter ces programmes et vérifier qu'ils fonctionnent je vais créer une app 'main'

```
import java.io.IOException;
public class Main {
    public static void main(String[] args) {
        String filePath =
"C:\\Users\\udemil\\eclipse-workspace\\DM2_Login\\src\\in";
        String copyPath = "copy_in.txt";
        try {
            Exo1.copyFile(filePath, copyPath);
            System.out.println("Fichier copié avec succès.");
            boolean auth = Exo2.authenticate("User1", "passUser1", filePath);
            System.out.println(auth ? "Authentification réussie" : "PB
Authentification");
            Exo3.countFileContent(filePath);
            Exo4.createLoginGUI(filePath);
        } catch (IOException e) {
            System.err.println("Erreur : " + e.getMessage());
        }
    }
}
```

**Voici le résultat obtenu :**

