Université de Carthage

Ecole Supérieure de la Statistique et de l'Analyse de l'Information

Enseignant.e.s : Mme Aïcha El Golli

Durée de l'épreuve : 1h30

Année Universitaire 2024/2025

Examen Session Principale
Module : POO II Java niveau 2

Niveau d'études : 3ème année

Les Documents, calculatrices, téléphone portable sont interdits. Veuillez rendre une copie propre et claire. La qualité de l'écriture et de la présentation sera prise en compte dans la note finale. Si la syntaxe d'une instruction est fausse alors la note est 0.

Questions de réflexion : (4 points)

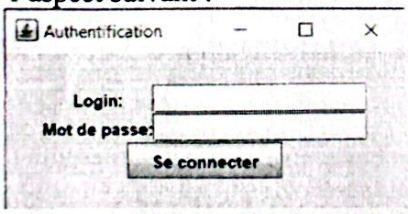
1- Nous souhaitons avoir un conteneur intermédiaire qui construit une interface divisée en 2 parties. Quelle est la classe du conteneur qui permet de le faire ?
2- A quel package appartient la classe DriverManager ?
3- L'utilisation de la méthode getConnection() du DriverManager est susceptible de lancer une exception, mais de quel type ? cette exception est-elle obligatoire à attraper ou pas (checked ou unchecked) ?
4- Dans l'API JDBC, que permet de faire la méthode insertRow() de la classe ResultSet ?

Exercice 1 (12 Points)

Nous disposons d'une base de données nommée **application** avec une table **users:**
users(**login**, mdp, nom, prenom, age), constituée de 5 propriétés.

Colonne	Type
login (primary key)	varchar(20)
mdp	varchar(20)
prenom	varchar(20)
nom	varchar(20)
age	int

Nous allons réaliser la classe **AuthentificationApp** qui est une fenêtre JFrame permettant aux utilisateurs de s'authentifier. Cette fenêtre doit avoir l'aspect suivant :



Le bouton Se connecter permet de se connecter à une application si le login et le mot de passe sont bons.
La zone centrale est composée de 2 labels, 2 zones de JTextField et un bouton. L'organisation des différents composants dans la fenêtre est réalisée avec un gestionnaire de layout **GridBagLayout au niveau d'un panel.**

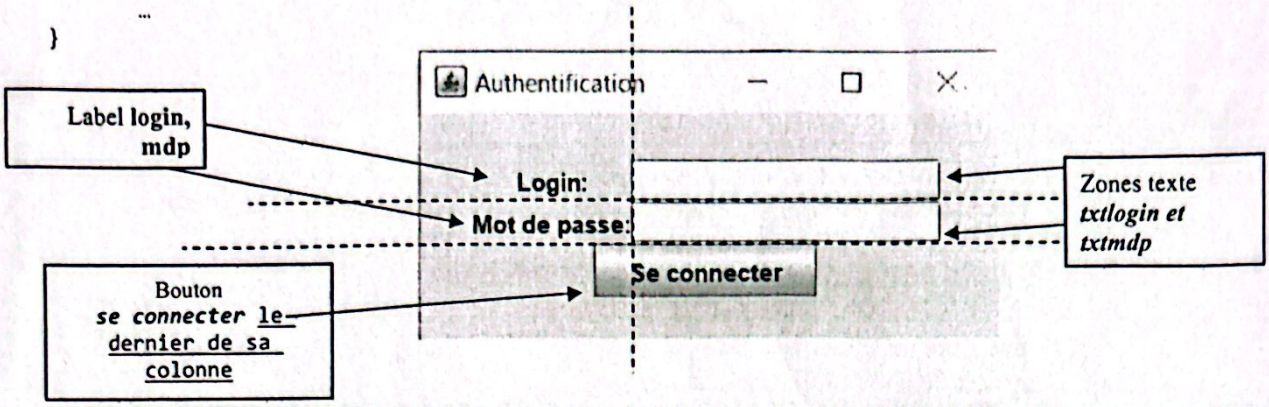
La classe **AuthentificationApp** est décrite comme suit :

```
public class AuthentificationApp extends JFrame{
              protected JButton seconnecter;
              protected JLabel login, mdp;
              protected JTextField txtlogin, txtmdp;
              protected JPanel panel;

static Connection maConnection() throws SQLException {
              final String url = "jdbc:mysql://localhost/application";
              Connection conn = DriverManager.getConnection(url, "root", "");
                 return conn;       }
public static void main(String[] args){
              JFrame f=new AuthentificationApp ();
               f.setVisible(true);    }

public AuthentificationApp() {
setTitle("Authentification");
setSize(300, 150);
setDefaultCloseOperation(EXIT_ON_CLOSE);
```

```
setLocationRelativeTo(null);
initComponents();
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
    ...
}
```



1. Complétez la méthode initiComponents() suivante pour placer les différents composants **(5,5 points):**

```
void initComponents() {
 //Initialisez les différents composants;
//les zones textes txtlogin, txtmdp sont initialisées avec une chaine vide et de taille 15
//Remplacez simplement le gestionnaire par défaut du panel par un gestionnaire de type
//GridBagLayout et ajoutez tous les à la fenêtre

...
                    } }
```

2. Nous devons traiter les événements de type Action déclenchés par le bouton ***se connecter.***

Nous commençons par récupérer le contenu des champs textes ***txtlogin et txtmdp*** afin de vérifier qu'ils ne sont pas vides et après vérifier si l'user existe dans la table users.

- soit l'user existe dans ce cas une boîte de dialogue informative affichera « Authentification réussie ! » et vous remettez par la suite à vide le contenu des champs texte ***txtlogin, txtmdp***.
- soit l'user n'existe pas et dans ce cas une boîte de dialogue informative l'invitera à re-saisir son login et mot de passe avec ce message « login ou mot de passe incorrect. Veuillez ressaisir vos données »

Sachant que nous disposons dans la classe **AuthentificationApp** d'une méthode static maConnection() décrite ci-dessus:

Terminez la méthode actionPerformed du bouton **se connecter** suivante **(3,5 points)** :

```
seconnecter.addActionListener(new ActionListener(){
@Override
public void actionPerformed(ActionEvent e) {
Connection bd=null;
try {….       }

catch(SQLException ex) {
JOptionPane.showMessageDialog(null, "une erreur SQL\n!" + ex.getMessage(), "Erreur",
JOptionPane.ERROR_MESSAGE);
}
finally {
      try {  bd.close();
      } catch (SQLException e1) {e1.printStackTrace();  }
                                                  }
}});
```

3. Dans la table users nous avons les 4 users suivants :

login	mdp	prenom	nom	age
hassen	hass1999	Hassen	Mejri	20
sony	sonialam	Sonia	Lamari	22
samsoum	ommi2020	Sami	Ben dali	21
zohra	1234	Zohra	Trimech	25

26

Qu'affiche le programme suivant **(3 points)** :

```
public class MaJdbc {
    static Connection maConnection() throws SQLException {
    final String url = "jdbc:mysql://localhost/application";
          Connection c = DriverManager.getConnection(url, "root", "");
          return c;       }

 public static void main(String[] args) {
            Connection bd=null;
            try {
                  bd = maConnection();
 Statement st = bd.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);

 String sql = "SELECT * FROM users";
 ResultSet res = st.executeQuery(sql);
 System.out.println("row= "+ res.getRow());
 while(res.next())
 {System.out.println(res.getString(3)+" - " +res.getString(5) ) ; }
 System.out.println("row= "+ res.getRow());
 res.absolute(0);
 while(res.next())
 {    if(res.getString(2).length()<6)
      {res.updateInt(5, res.getInt(5)+1);
       res.updateRow();}
 }
res.moveToInsertRow();
res.updateString(1, "essailogin");
res.updateString(2, "essai12");
res.updateString(3, "etudiant");
res.updateString(4, "Salha");
res.updateString(5, "20");
res.insertRow();
res.beforeFirst();
while(res.next())
{System.out.println(res.getString(1)+" - " +res.getString(5) ) ; }

} catch (SQLException e) {e.printStackTrace();
                  if(bd==null)System.out.println("la connection a echoué");
                  else System.out.println("une SQLexception est levée");
            }
            finally
            {     try {if (bd != null ) {bd.close();  System.out.println("connection
fermée");}}
                  catch (SQLException e)
                  {     System.out.println("Impossible de fermer la connection.") ;      }
            } } }
```

Exercice 2 (4 points)

On a demandé à ChatGPT d'écrire une interface graphique avec swing, voici le code généré :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class InterfaceGraphique extends JFrame {

    public InterfaceGraphique() {
        setTitle("Interface avec Boutons");
        setSize(300, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        JPanel panel = new JPanel();
        panel.setLayout(new FlowLayout());
        JButton okButton = new JButton("OK");
```

```
        okButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
         JOptionPane.showMessageDialog(InterfaceGraphique.this, "Vous avez cliqué sur OK",
"Information", JOptionPane.INFORMATION_MESSAGE);
            }
        });

        JButton cancelButton = new JButton("Annuler");
        cancelButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
JOptionPane.showMessageDialog(InterfaceGraphique.this, "Vous avez cliqué sur Annuler",
"Information", JOptionPane.INFORMATION_MESSAGE);
            }
        });

        panel.add(okButton);
        panel.add(cancelButton);
        add(panel, BorderLayout.CENTER);
        setVisible(true);
    }

    public static void main(String[] args) {
InterfaceGraphique ig=  new InterfaceGraphique();
            }
      }
```

1. Dessinez l'interface obtenue
2. Qu'est ce qui peut être amélioré dans ce code ?
3. ChatGPT a implémenté les écouteurs en classes anonymes, réécrire la classe InterfaceGraphique pour qu'elle soit son propre écouteur (implémentant les écouteurs au niveau de la classe principale InterfaceGraphique) tout en améliorant et simplifiant le code

Bon travail