

Compte Rendu TP1 java

Année Universitaire : 2024/2025

Réalisé par :

El Khelladi Khaoula

Encadré par :

Kchbal Ilham

0.1 Introduction

Ce premier travail pratique en Java porte sur la gestion des employés en utilisant les concepts de DAO (Data Access Object) et MVC (Modèle-Vue-Contrôleur). L'objectif est de séparer les différentes couches de l'application afin d'améliorer la maintenabilité et la réutilisabilité du code. Le modèle gère les données des employés, la vue permet d'afficher ces données à l'utilisateur, et le contrôleur assure l'interaction entre les deux. En utilisant DAO, nous avons séparé la logique de gestion des données du reste de l'application, permettant ainsi une meilleure organisation et une plus grande flexibilité dans l'accès à la base de données.

0.2 Les étapes de projet

Voici une section avec du contenu exemple.

0.2.1 La class Employee

Tout d'abord, je crée la classe principale Employe dans le model, Voici le script.

```
package Model;
    public class Employe {
private int id;
public int getId() {
return id;
    }

    public void setId(int id) {
        this.id = id;
    }
public Employe(int id, String nom, String prenom, String email, String telephone,
Poste poste) {
    super();
    this.id = id;
    this.nom = nom;
    this.prenom = prenom;
    this.email = email;
    this.telephone = telephone;
    this.salaire = salaire;
    this.role = role;
    this.poste = poste;
}
private String nom;
private String prenom;
private String email;
private String telephone;
private double salaire;
private Role role;
private Poste poste;
```

```

public Employe(String nom,String prenom,String email,
String telephone,double salaire,Role role,Poste poste) {
    this.nom=nom;
    this.prenom=prenom;
    this.email=email;
    this.telephone=telephone;
    this.salaire=salaire;
    this.role=role;
    this.poste=poste;
}

    public String getNom() {
        return nom;
    }
public void setNom(String nom) {
    this.nom=nom;
}

public String getPrenom() {
    return prenom;
}

public String getEmail() {
    return email;
}

public String getTelephone() {
    return telephone;
}

public double getSalaire() {
    return salaire;
}

public Role getRole() {
    return role;
}

public Poste getPoste() {
    return poste;
}

public void setPrenom(String prenom) {
    this.prenom = prenom;
}

public void setEmail(String email) {
    this.email = email;
}

```

```

}

public void setTelephone(String telephone) {
    this.telephone = telephone;
}

public void setSalaire(double salaire) {
    this.salaire = salaire;
}

public void setRole(Role role) {
    this.role = role;
}

public void setPoste(Poste poste) {
    this.poste = poste;
}

public enum Role {
    ADMIN,
    EMPLOYE
}

public enum Poste {
    INGENIEURE,
    TEAM_LEADER,
    PILOTE
}
}

```

0.2.2 la classe Connexion

puis je crée la classe Connexion pour la connexion avec la base de données dans le DAO. Voici le script.

```

package DAO;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class connexion {
    public static final String url = "jdbc:mysql://localhost:3306/salma";
    public static final String user = "root";
    public static final String password = "";
    private static Connection conn = null;

    public static Connection getConnexion() {
        if (conn == null) {

```

```

        try {
            conn = DriverManager.getConnection(url, user, password);
            System.out.println("Connexion_etablie_avec_succes_!");
        } catch (SQLException e) {
            System.out.println("Erreur_de_connexion_!!!!");
        }
    }
    return conn;
}
public static void closeConnexion() {
    if (conn != null) {
        try {
            conn.close();
        } catch (SQLException e) {
            System.out.println("Erreur_lors_de_la_fermeture_de_la_connexion_!!!!");
        }
    }
}
}

```

0.2.3 la classe EmployeI

puis je crée la classe EmployeI c'est l'interface qui contient les methodes dans le DAO, Voici le script.

```

package DAO;

import java.util.List;

import Model.Employe;
import Model.Employe.Poste;
import Model.Employe.Role;

public interface EmployeI {

    Employe findById(int employeeId);
    List<Employe> findAll();
    void add(Employe E);
    void update(Employe E, int id);
    void delete(int id);
    List<Role>findAllRoles();
    List<Poste>findAllPostes();

}

```

0.2.4 la classe EmployeImpl

puis je crée la classe EmployeImpl pour l'implémentation des méthodes de l'interface dans le DAO, Voici le script.

```
package DAO;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import Model.Employe.Role;

import Model.Employe;
import Model.Employe.Poste;

public class EmployeImpl implements EmployeI {
    private Connection conn;

    public EmployeImpl() {
        this.conn = connexion.getConnexion();
    }

    @Override
    public void add(Employe E) {
        String Query = "INSERT INTO Employee(nom, _prenom, _email, _te

try (PreparedStatement stmt = conn.prepareStatement(Query)) {
        stmt.setString(1, E.getNom());
        stmt.setString(2, E.getPrenom());
        stmt.setString(3, E.getEmail());
        stmt.setString(4, E.getTelephone());
        stmt.setDouble(5, E.getSalaire());
        stmt.setString(6, E.getRole().name());
        stmt.setString(7, E.getPoste().name());
        stmt.executeUpdate();
        System.out.println("Employe _ajoute _avec _succes _!");
    } catch (SQLException e) {
        System.out.println("Erreur _lors _de _l 'ajout _de _l 'employe _!");
        //e.printStackTrace();

    }
}
```

```

        @Override
        public Employee findById(int employeeId) {
            String query = "SELECT_*_FROM_Employee_WHERE_id_=_" + employeeId + "?";
            try (PreparedStatement stmt = conn.prepareStatement(query)) {
                stmt.setInt(1, employeeId);
                ResultSet rs = stmt.executeQuery();
                if (rs.next()) {
                    return new Employee(
                        rs.getString("nom"),
                        rs.getString("prenom"),
                        rs.getString("email"),
                        rs.getString("telephone"),
                        rs.getDouble("salaire"),
                        Employee.Role.valueOf(rs.getString("role")),
                        Poste.valueOf(rs.getString("poste"))
                    );
                }
            } catch (SQLException e) {
                System.out.println("Erreur lors de la recherche de l'employé par ID : " + e.getMessage());
                //e.printStackTrace();
            }
            return null;
        }

```

```

        @Override
        public List<Employee> findAll() {
            List<Employee> employees = new ArrayList<>();
            String query = "SELECT_*_FROM_Employee";
            try (Statement stmt = conn.createStatement();
                ResultSet rs = stmt.executeQuery(query)) {
                while (rs.next()) {
                    employees.add(new Employee(
                        rs.getInt("id"),
                        rs.getString("nom"),
                        rs.getString("prenom"),
                        rs.getString("email"),
                        rs.getString("telephone"),
                        rs.getDouble("salaire"),
                        Employee.Role.valueOf(rs.getString("role")),
                        Poste.valueOf(rs.getString("poste"))
                    ));
                }
            } catch (SQLException e) {
                System.out.println("Erreur lors de la récupération de tous les employés : " + e.getMessage());
                //e.printStackTrace();
            }
            return employees;
        }

```

```
}
```

```
@Override
```

```
public void update(Employee E, int id) {  
    String query = "UPDATE_Employee_SET_nom=_?,_prenom=_?,_email=_?"  
    try (PreparedStatement stmt = conn.prepareStatement(query)) {  
        stmt.setString(1, E.getNom());  
        stmt.setString(2, E.getPrenom());  
        stmt.setString(3, E.getEmail());  
        stmt.setString(4, E.getTelephone());  
        stmt.setDouble(5, E.getSalaire());  
        stmt.setString(6, E.getRole().name());  
        stmt.setString(7, E.getPoste().name());  
        stmt.setInt(8, id);  
        stmt.executeUpdate();  
        System.out.println("Employe_modifier_avec_succes_!");  
    } catch (SQLException e) {  
        System.out.println("Erreur_lors_de_la_modification_de_l'employe_!!!!");  
        //e.printStackTrace();  
    }  
}
```

```
@Override
```

```
public void delete(int id) {  
    String query = "DELETE_FROM_Employee_WHERE_id=_?";  
    try (PreparedStatement stmt = conn.prepareStatement(query)) {  
        stmt.setInt(1, id);  
        stmt.executeUpdate();  
        System.out.println("Employe_supprime_avec_succes_!");  
    } catch (SQLException e) {  
        System.out.println("Erreur_lors_de_la_suppression_de_l'emp  
        //e.printStackTrace();  
    }  
}
```

```
@Override
```

```
public List<Employee.Role> findAllRoles() {  
    return Arrays.asList(Employee.Role.values());  
}
```

```
@Override
```

```
public List<Poste> findAllPostes() {  
    return Arrays.asList(Poste.values());  
}
```

```
}
```


0.2.5 la classe EmployeModel

puis je crée la classe EmployeModel pour faire la logique métier dans le Model, Voici le script.

```
package Model;

import DAO.EmployeImpl;
import Model.Employe.Poste;

public class EmployeModel {
    private EmployeImpl dao;

    public EmployeModel(EmployeImpl dao) {
        this.dao=dao;
    }

    //logique Metier
    public boolean addEmploye(String nom,String pronom,String email,String telephone) {
        if (salaire <=0) {
            System.out.println("Le_salaire_doit_etre_superieur_0");
            return false;
        }

        if (email == null || !email.contains("@")) {
            System.out.println("L'email_n'est_pas_valide!");
            return false;
        }

        Employe NvEmploye = new Employe(nom,pronom,email,telephone);
        dao.add(NvEmploye);
        return true;
    }
}
```

0.2.6 la classe EmployeController

puis je crée la classe EmployeController pour faire une liaison entre le view et model dans le Controller, Voici le script.

```
package controller;

import java.util.List;

import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

import DAO.EmployeImpl;
import Model.Employe;
import Model.Employe.Poste;
import Model.Employe.Role;
```

```

import Model.EmployeModel;
import View.EmployeView;

public class EmployeController {
    private EmployeModel model;
    private EmployeView view;

    public EmployeController(EmployeModel model, EmployeView view) {
        this.model=model;
        this.view=view;
        this.view.btnAjouter.addActionListener(e->addEmploye());
        this.view.btnModifier.addActionListener(e->updateEmploye());
        this.view.btnAfficher.addActionListener(e -> afficherEmploye());
        this.view.btnSupprimer.addActionListener(e -> supprimerEmploye());

    }
    private void addEmploye() {
        String nom=view.getNom();
        String prenom=view.getPrenom();
        String email=view.getEmail();
        String telephone=view.getTelephone();
        double salaire =view.getSalaire();
        Poste poste=view.getPoste();
        Role role=view.getRole();

        boolean addEmploye=model.addEmploye(nom, prenom, email, telephone, salaire ,
        if(addEmploye) System.out.println("Employe_ajoute_avec_Succes");
        else System.out.println("Echec_d'ajout_d'employee_!!!!");
    }

    private void updateEmploye() {
        int selectedRow = view.table.getSelectedRow();
        int id = (int) view.table.getValueAt(selectedRow, 0);

        String nom=view.getNom();
        String prenom=view.getPrenom();
        String email=view.getEmail();
        String telephone=view.getTelephone();
        double salaire =view.getSalaire();
        Poste poste=view.getPoste();
        Role role=view.getRole();

```

```
Employee employe = new Employee(nom, prenom, email, telephone, salaire, role,
EmployeeImpl employeImpl = new EmployeeImpl());
```

```
employeImpl.update(employe, id);
```

```

    }
    public void afficherEmploye() {
        EmployeeImpl employeImpl = new EmployeeImpl();
        List<Employee> employes = employeImpl.findAll();

        DefaultTableModel model = (DefaultTableModel) view.table.getModel();
        model.setRowCount(0);

        for (Employee employe : employes) {
            model.addRow(new Object[]{
employe.getId(),
employe.getNom(),
employe.getPrenom(),
employe.getEmail(),
employe.getTelephone(),
employe.getSalaire(),
employe.getRole(),
employe.getPoste()
            });
        }
    }
    public void supprimerEmploye() {
        int selectedRow = view.table.getSelectedRow();
        if (selectedRow == -1) {
            JOptionPane.showMessageDialog(null, "Veuillez_selectionner_un_employe");
            return;
        }
        int id =view.getId(view.table);

        EmployeeImpl employeImpl = new EmployeeImpl();

        int confirmation = JOptionPane.showConfirmDialog(null, "Voulez-vous_vraiment_supprimer_cet_employe");
        if (confirmation == JOptionPane.YES_OPTION) {
            employeImpl.delete(id);
            JOptionPane.showMessageDialog(null, "Employe_supprime_avec_succes");
        }
    }
}

```

0.2.7 la classe EmployeView

puis je crée la classe EmployeView pour faire un interface graphique dans le View, Voici le script.

```
package View;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import Model.Employe.Role;
import Model.Employe.Poste;

import java.awt.*;
import java.util.ArrayList;

public class EmployeView extends JFrame {

    private JPanel mainPanel, topPanel, centerPanel, bottomPanel;
private JLabel lblNom, lblPrenom, lblEmail, lblTelephone, lblSalaire, lblPoste;
private JTextField txtNom, txtPrenom, txtEmail, txtTelephone, txtSalaire;
private JComboBox<Poste> cbPoste;
private JComboBox<Role> cbRole;
public JTable table;
public JButton btnAjouter;
public JButton btnModifier;
public JButton btnSupprimer;
public JButton btnAfficher;

    public EmployeView() {
        setTitle("Gestion_des_Employes");
        setSize(600, 400);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        mainPanel = new JPanel(new BorderLayout());
        topPanel = new JPanel(new GridLayout(7, 2, 10, 10));
        centerPanel = new JPanel(new BorderLayout());
        bottomPanel = new JPanel(new GridLayout(1, 4, 10, 10));

        lblNom = new JLabel("Nom:");
        lblPrenom = new JLabel("Prenom:");
        lblEmail = new JLabel("Email:");
        lblTelephone = new JLabel("Telephone:");
        lblSalaire = new JLabel("Salaire:");
        lblPoste = new JLabel("Poste:");
        lblRole = new JLabel("Role:");

        txtNom = new JTextField();
```

```

txtPrenom = new JTextField();
txtEmail = new JTextField();
txtTelephone = new JTextField();
txtSalaire = new JTextField();

cbRole = new JComboBox<>(Role.values());
cbPoste = new JComboBox<>(Poste.values());

table = new JTable(new DefaultTableModel(new Object[] { "ID", "Nom", "Prenom", "Email", "Telephone", "Salaire", "Role", "Poste" }));
JScrollPane scrollPane = new JScrollPane(table);

btnAjouter = new JButton("Ajouter");
btnModifier = new JButton("Modifier");
btnSupprimer = new JButton("Supprimer");
btnAfficher = new JButton("Afficher");

topPanel.add(lblNom);
topPanel.add(txtNom);
topPanel.add(lblPrenom);
topPanel.add(txtPrenom);
topPanel.add(lblEmail);
topPanel.add(txtEmail);
topPanel.add(lblTelephone);
topPanel.add(txtTelephone);
topPanel.add(lblSalaire);
topPanel.add(txtSalaire);
topPanel.add(lblRole);
topPanel.add(cbRole);
topPanel.add(lblPoste);
topPanel.add(cbPoste);

centerPanel.add(scrollPane, BorderLayout.CENTER);

bottomPanel.add(btnAjouter);
bottomPanel.add(btnModifier);
bottomPanel.add(btnSupprimer);
bottomPanel.add(btnAfficher);

mainPanel.add(topPanel, BorderLayout.NORTH);
mainPanel.add(centerPanel, BorderLayout.CENTER);
mainPanel.add(bottomPanel, BorderLayout.SOUTH);

add(mainPanel);
setVisible(true);
}

```

```

    public int getId(JTable table) {
        int selectedRow = table.getSelectedRow();

        if (selectedRow == -1) {
            JOptionPane.showMessageDialog(null, "Veuillez_selectionner");
            return -1;
        }
        return (int) table.getValueAt(selectedRow, 0);
    }

    public String getNom() {
        return txtNom.getText();
    }
    public String getPrenom() {
        return txtPrenom.getText();
    }
    public String getEmail() {
        return txtEmail.getText();
    }
    public String getTelephone() {
        return txtTelephone.getText();
    }
    public double getSalaire() {
        return Double.parseDouble(txtSalaire.getText());
    }
    public Role getRole() {
        Role r=(Role) cbRole.getSelectedItem();
        return r;
    }
    public Poste getPoste() {
        Poste p=(Poste) cbPoste.getSelectedItem();
        return p;
    }

    public static void main(String[] args) {
        new EmployeView();
    }
}

```

0.2.8 la classe main

puis je crée la classe main pour faire le teste dans le Main, Voici le script.

```

package main;

import DAO.EmployeImpl;

```

```

import Model.EmployeModel;
import View.EmployeView;
import controller.EmployeController;

public class main {

    public static void main(String[] args) {
        EmployeView View = new EmployeView();
        EmployeImpl DAO = new EmployeImpl();
        EmployeModel Model = new EmployeModel(DAO);
        new EmployeController(Model, View);

        View.setVisible(true);
    }
}

```

0.3 Captures d'écran

ID	Nom	Prénom	Email	Téléphone	Salaire	Poste	Rôle

FIGURE 1 – L'interface de l'application