

École Supérieure de Technologie - Safi

Rapport de TP2

Application de Gestion des Congés avec Généricité, MVC et DAO

Filière : Génie Informatique Réalisé par : Omar Akalay

Année Universitaire: 2024/2025

Contents

Introduction		2
1	Couche Modèle 1.1 Implémentation du Modèle	3 3 3
2	Couche DAO 2.1 Interface GenericDAOI	4 4 4
3	Interface Graphique 3.1 Conception de l'Interface	5
4	Contrôleur 4.1 Gestion des événements	6
Co	Conclusion	

Introduction

L'objectif principal de ce TP est de développer une application de gestion des congés en Java en utilisant les concepts de généricité, DAO et MVC. Ce rapport présente les étapes de la conception à l'implémentation.

1. Couche Modèle

1.1. Implémentation du Modèle

Dans cette étape, une nouvelle classe nommée Holiday a été introduite avec les attributs à gérer. Les getters, setters, et une énumération HolidayType permettent une gestion propre des données.

1.1.1. Exemple de Code

```
public class Holiday {
    private int id;
    private String name;
    private HolidayType type;

    // Constructeur
    public Holiday(int id, String name, HolidayType type) {
        this.id = id;
        this.name = name;
        this.type = type;
    }

    // Getters et setters
}
```

2. Couche DAO

2.1. Interface GenericDAOI

```
public interface GenericDAOI<T> {
    void add(T t);
    void update(T t);
    void delete(int id);
    T findById(int id);
    List<T> findAll();
}
```

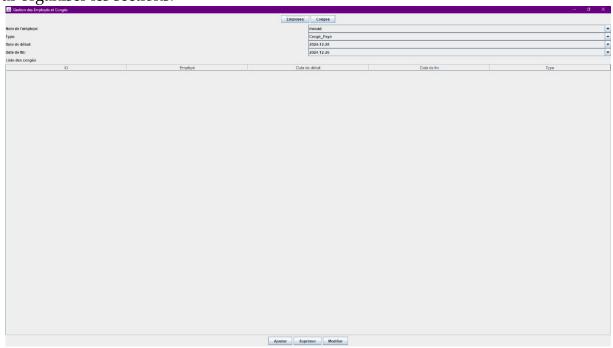
2.2. Implémentation de HolidayDAOImp

```
public class HolidayDAOImp implements GenericDAOI<Holiday> {
      // Implémentation des méthodes CRUD
}
```

3. Interface Graphique

3.1. Conception de l'Interface

L'application utilise Swing pour concevoir l'interface graphique, notamment avec JTabbedPane pour organiser les sections.



4. Contrôleur

4.1. Gestion des événements

```
public class HolidayController {
    private HolidayModel holidayModel;
    private HolidayView view;

public HolidayController(HolidayModel holidayModel, HolidayView view) {
        this.holidayModel = holidayModel;
        this.view = view;
        // Gestion des événements
    }
}
```

Conclusion

Ce TP a permis d'intégrer plusieurs concepts avancés de POO pour concevoir une application complète en Java. Cette approche assure une bonne maintenabilité et facilité d'extension.