

Rapport de TP2 JAVA

GHANDOUR Nouhaila

December 26, 2024

Lien GitHub

Vous pouvez accéder au projet complet sur GitHub via le lien suivant: [NouhailaGhand/TP-java-](https://github.com/NouhailaGhand/TP-java-)

Interface G'én'érique

Dans cette section, nous avons expliqué le code d'une interface g'én'érique en Java. L'interface suivante est utilisée pour effectuer des opérations de base de données (ajouter, supprimer, mettre à jour, et afficher des éléments g'én'ériques).

Code de l'Interface

Voici l'implémentation du code de l'interface g'én'érique :

```
1 package DAO;  
2  
3 import java.util.List;  
4  
5 public interface GenericDAOI <T> {  
6     public void add(T e);  
7     public void delete(int id);  
8     public void update(T e);  
9     public List<T> display();  
10 }
```

Explication du Code

- Cette interface spécifie les méthodes fondamentales pour manipuler des objets de type générique T. Voici une présentation des différentes méthodes proposées :
- `add(T e)` : Cette méthode permet d'ajouter un élément de type T dans une base de données.
- `delete(int id)` : Cette méthode permet de supprimer un élément de type T de la base de données, en utilisant un identifiant unique.
- `update(T e)` : Cette méthode permet de mettre à jour un élément de type T dans la base de données.

- `display()` : Cette méthode retourne une liste d'éléments de type T, permettant d'afficher les données de la base.

Implémentation de HolidayDAOimpl

La classe `HolidayDAOimpl` implémente l'interface `GenericDAOI<Holiday>`. Elle est responsable de la gestion des congés des employés dans la base de données. Ci-dessous, vous trouverez la structure de cette classe ainsi qu'une description des principales méthodes qu'elle contient.

Code de la Classe HolidayDAOimpl

```

1 package DAO;
2
3 public class HolidayDAOimpl implements GenericDAOI<Holiday>{
4
5     @Override
6     public void add(Holiday e) {...}
7
8     @Override
9     public void delete(int id) {...}
10
11     @Override
12     public void update(Holiday e) {...}
13
14     @Override
15     public List<Holiday> display() {...}
16 }
```

Explication de la Classe HolidayDAOimpl

La classe `HolidayDAOimpl` implémente l'interface générique `GenericDAOI<Holiday>` et fournit les méthodes nécessaires pour gérer les congés des employés dans la base de données.

- `add(Holiday e)` : Cette méthode permet d'ajouter un nouveau congé dans la base de données.
- `delete(int id)` : Cette méthode permet de supprimer un congé existant en utilisant son identifiant.
- `update(Holiday e)` : Cette méthode met à jour un congé existant dans la base de données.
- `display()` : Cette méthode retourne une liste de tous les congés enregistrés dans la base de données.

je cache le code de chaque fonction par ce que comme le file `EmployeeDAOimpl`.

Classe Holiday

La classe Holiday représente un congé d'un employé dans le système. Elle contient des informations essentielles telles que l'identifiant du congé, l'identifiant de l'employé, la date de début et de fin du congé, ainsi que le type de congé. Cette classe sert de modèle pour gérer les données liées aux congés dans la base de données.

Code de la Classe Holiday

Voici l'implémentation de la classe Holiday :

```
1 package Model;
2
3 import java.sql.Date;
4
5 public class Holiday{
6     private int id_holiday;
7     private int id_employe;
8     private Date startDate;
9     private Date endDate;
10    private Type_holiday type;
11
12    public Holiday(int id_holiday, int id_employe, Date startDate, Date endDate , Type_holiday type){
13        this.id_holiday = id_holiday;
14        this.id_employe = id_employe;
15        this.startDate = startDate;
16        this.endDate = endDate; 17 this.type = type;
18    }
19
20    public int getId_holiday() { 21    return
id_holiday;
22    }
23
24    public Date getStartDate() { 25    return
startDate;
26    }
27
28    public Date getEndDate() {
29        return endDate;
30    }
31
32    public Type_holiday getType() {
33        return type;
34    }
35
36    public int getId_employe() { 37    return
id_employe;
38    }
39    }
```

Classe HolidayModel

La classe HolidayModel gère la logique métier des congés. Elle interagit avec la classe HolidayDAOimpl pour effectuer des opérations sur les congés et réaliser des vérifications logiques avant d'ajouter, de supprimer ou de mettre à jour des congés dans la base de données. La méthode principale de cette classe est addHoliday, qui comprend plusieurs vérifications avant l'ajout d'un congé.

Code de la Classe HolidayModel

Voici l'implémentation de la classe HolidayModel :

```
1 package Model;
2
3 public class HolidayModel {
4     private HolidayDAOimpl dao;
5
6     public HolidayModel(HolidayDAOimpl dao) {
7         this.dao = dao;
8     }
9
10    public boolean addHoliday(int id, int id_employe, Date startdate, Date enddate, Type_holiday type ,
11        Employe targetEmploye) {
12
13        if(startdate.after(enddate)) return false;
14        if(startdate.equals(enddate)) return false;
15        if(startdate.before(new Date(System.currentTimeMillis()))) return false;
16        if(enddate.before(new Date(System.currentTimeMillis()))) return false;
17        for (Holiday holiday : new HolidayModel(new HolidayDAOimpl()).
18            displayHoliday()) {
19            if (holiday.getId_employe() == id_employe) {
20                if(startdate.after(holiday.getStartDate()) &&
21                    startdate.before(holiday.getEndDate()) &&
22                    enddate.before(holiday.getEndDate()) &&
23                    enddate.after(holiday.getStartDate())){
24                    System.out.println("le congé est superposé à
25                        un autre congé");
26                    return false;
27                }
28
29                if(startdate.after(holiday.getStartDate()) &&
30                    startdate.before(holiday.getEndDate()) &&
31                    enddate.after(holiday.getEndDate())){
32                    startdate = holiday.getEndDate();
33                    continue;
34                }
35                if(startdate.before(holiday.getStartDate()) &&
36                    enddate.after(holiday.getStartDate()) && enddate
37                        .before(holiday.getEndDate())){
38                    enddate = holiday.getStartDate();
39                    continue;
40                }
41            }
42        }
43    }
44 }
```

```

30         }
31         break;
32     }
33 }
34
35 long daysBetween = (enddate.toLocalDate().toEpochDay() - startdate.toLocalDate().toEpochDay());
36 if(daysBetween > targetEmploye.getSolde()) return false;
37 EmployeController.updateSolde(targetEmploye.getId(), targetEmploye.getSolde() - (int)
    daysBetween);
38 Holiday e = new Holiday(id, id_employe, startdate, enddate,
    type);
39 dao.add(e);
40 return true;
41 }
42 }

```

Explication de la Méthode addHoliday

La méthode addHoliday gère l'ajout d'un congé pour un employé. Elle effectue plusieurs vérifications avant d'ajouter le congé dans la base de données. Voici les étapes de cette méthode :

- **Vérification des dates:**
 - Si la date de début est après la date de fin, la méthode retourne false.
 - Si la date de début est égale à la date de fin, la méthode retourne false.
 - Si la date de début ou la date de fin est antérieure à la date actuelle, la méthode retourne false.
- **Vérification des chevauchements de congés:** La méthode vérifie si le nouveau congé chevauche un congé existant de l'employé. Si un chevauchement est trouvé, la méthode retourne false. Si le congé est partiellement chevauché, la méthode ajuste les dates de début et de fin.
- **Vérification du solde de congés:** La méthode compare le nombre de jours du congé demandé avec le solde de congés de l'employé. Si le solde est insuffisant, la méthode retourne false.
- **Mise à jour du solde de congés:** Si toutes les vérifications sont réussies, le solde de congés de l'employé est mis à jour en soustrayant le nombre de jours demandés.
- **Ajout du congé:** Enfin, un objet Holiday est créé et ajouté à la base de données via le HolidayDAOimpl.

Classe HolidayController

La classe HolidayController est responsable de la gestion des interactions entre l'interface utilisateur, le modèle de données et la logique métier des congés. Elle gère les actions de l'utilisateur telles que l'ajout, la suppression, la mise à jour et l'affichage des congés dans l'interface.

Explication du Code

Cette classe interagit avec les vues et le modèle pour effectuer les opérations sur les congés des employés. Voici les différentes méthodes et leurs rôles :

- **HolidayController** : Constructeur qui initialise les composants de la vue et lie les actions de l'utilisateur (ajout, mise à jour, suppression, affichage) aux méthodes correspondantes de la classe.

```
1      public HolidayController(Employe_HolidayView view, HolidayModel model) {
2          this.View = view;
3          this.model_holiday = model;
4
5          View.getdeleteButton_holiday().addActionListener(e -> deleteHoliday());
6          View.getupdateButton_holiday().addActionListener(e -> updateHoliday());
7          View.getaddButton_holiday().addActionListener(e -> addHoliday());
8      }
```

- **addHoliday** : Cette méthode permet d'ajouter un nouveau congé. Elle récupère les informations saisies par l'utilisateur, vérifie si l'employé existe, puis ajoute le congé via la méthode addHoliday du modèle HolidayModel. Si l'ajout réussit, un message de succès est affiché.

```
1      private void addHoliday(){
2          int id_employe = View.getId_employe();
3
4          Date startDate = Date.valueOf(View.getStartDate());
5
6          Date endDate = Date.valueOf(View.getEndDate());
7
8          Type_holiday type = View.getType_holiday();
9
10         View.viderChamps_ho();
11
12         boolean addreussi = model_holiday.addHoliday(0, id_employe, startDate, endDate, type,
13             targetEmploye);
14         if(addreussi) {
15
16             View.afficherMessageSucces("Holiday a bien ete ajoutee.
17
18             ");
19             displayHoliday();
20
21         } else {
22
23             View.afficherMessageErreur("Holiday n'a pas ete ajoutee
24
25             .");
26         }
27     }
```

```

14         }
15     }

```

- **displayHoliday** : Cette méthode affiche tous les congés dans la vue sous forme de tableau. Elle récupère la liste des congés à partir du modèle, puis les affiche dans la table de la vue. Si aucun congé n'est trouvé, un message d'erreur est affiché.

```

1         private void displayHoliday() {
2             List<Holiday> Holidays = model_holiday.displayHoliday();
3             DefaultTableModel tableModel1 = (DefaultTableModel)
4             Employe_HolidayView.Tableau1.getModel();
5             tableModel1.setRowCount(0);
6             for (Holiday e : Holidays) {
7                 tableModel1.addRow(new Object[]{e.getId_holiday(), e.getStartData(),
8                 e.getEndDate(), e.getType()});
9             }
10        }

```

- **deleteHoliday** : Permet de supprimer un congé sélectionné. Elle récupère l'ID du congé sélectionné dans la table et le supprime via la méthode deleteHoliday du modèle. Si la suppression réussit, le solde de l'employé est mis à jour et un message de succès est affiché.

```

1         private void deleteHoliday(){
2             int selectedrow = Employe_HolidayView.Tableau1.getSelectedRow();
3             if (selectedrow == -1) {
4                 View.afficherMessageErreur("Veuillez selectionner une ligne.");
5             } else {
6                 int id = (int) Employe_HolidayView.Tableau1.getValueAt( selectedrow, 0);
7                 boolean deletereussi = model_holiday.deleteHoliday(id);
8                 if(deletereussi){
9                     EmployeController.updateSolde(id_employe,solde+ olddaysbetween);
10                    View.afficherMessageSucces("Holiday a bien ete
11                    supprimer.");
12                    displayHoliday();
13                } else {
14                    View.afficherMessageErreur("Holiday n'a pas ete
15                    supprimer.");
16                }
17            }
18        }

```

- **updateHolidaybyselect** : Cette méthode permet de charger les informations d'un congé sélectionné dans les champs de saisie pour les modifier. Elle récupère les informations du congé sélectionné et les affiche dans les champs correspondants de la vue.

```

1         private void updateHolidaybyselect(){
2             int selectedrow = Employe_HolidayView.Tableau1.getSelectedRow();
3             if (selectedrow != -1) {

```

```

4         id = (int) Employee_HolidayView.Tableau1.getValueAt( selectedrow, 0);
5         OldstartDate = String.valueOf(Employee_HolidayView.
           Tableau1.getValueAt(selectedrow, 2));
6         View.remplaireChamps_ho(id, OldstartDate, OldendDate);
7     }
8 }

```

- **updateHoliday** : Permet de mettre à jour un congé existant. Cette méthode vérifie que l'utilisateur a sélectionné un congé à modifier, puis récupère les nouvelles données saisies et les envoie au modèle pour mettre à jour le congé. Un message de succès ou d'erreur est affiché selon le résultat de la mise à jour.

```

1     private void updateHoliday(){
2     if (!test) {
3         View.afficherMessageErreur("Veuillez d'abord
           sélectionner une ligne à modifier.");
4         return;
5     }
6     boolean updateSuccessful = model_holiday.updateHoliday(id, id_employe,
       startDate, endDate, type, targetEmploye);
7     if (updateSuccessful) {
8         View.afficherMessageSucces("Holiday a été modifié avec succès.");
9         displayHoliday();
10        View.viderChamps_ho();
11    } else {
12        View.afficherMessageErreur("Erreur lors de la mise à jour de holiday.");
13    }
14 }

```


Vue Employe HolidayView

Gestion des employes et des congés

Employe Holiday

Nom

Prenom

Email

Telephone

Salaire

Role

Poste

ID	Nom	Prenom	Email	Téléphone	Salaire	Role	Poste	solde
33	koubi	mohamed-yassine	mohamedyassi...	0645593798	15000.0	ADMIN	TEAM_LEADER	25

Ajouter Modifier Supprimer Afficher Create Acconte

Figure 1: Ajouter un employe
le solde de l'employe est 25.

Gestion des employes et des congés

Employe Holiday

Nom de l'employé

Date de debut

Date de fin

Type

ID	nom_employe	date_debut	date_fin	type
71	33 - koubi mohamed-yassine	2024-12-27	2024-12-31	CONGE_PAYE

Ajouter Modifier Supprimer Afficher

Figure 2: Ajouter un holiday le
conge contient 4 days.

Gestion des employes et des congés

Employe Holiday

Nom

Prenom

Email

Telephone

Salaire

Role

Poste

ADMIN

PILOT

ID	Nom	Prenom	Email	Téléphone	Salaire	Role	Poste	solde
33	koubi	mohamed-yassine	mohamedyassi...	0645593798	15000.0	ADMIN	TEAM_LEADER	21

Ajouter Modifier Supprimer Afficher Create Acconte

Figure 3: new solde

Gestion des employes et des congés

Employe Holiday

Nom de l'employé

Date de debut

Date de fin

Type

33 - koubi mohamed-yassine

2024-12-28

2025-01-05

CONGE_PAYE

ID	nom_employe	date_debut	date_fin	type
71	33 - koubi mohamed-yassine	2024-12-27	2024-12-31	CONGE_PAYE

Ajouter Modifier Supprimer Afficher

Gestion des employes et des congés

Employe Holiday

Nom de l'employé: 33 - koubi mohamed-yassine

Date de debut:

Date de fin:

Type: CONGE_PAYE

ID	nom_employe	date_debut	date_fin	type
71	33 - koubi mohamed-yassine	2024-12-27	2024-12-31	CONGE_PAYE
72	33 - koubi mohamed-yassine	2024-12-31	2025-01-05	CONGE_PAYE

Ajouter Modifier Supprimer Afficher

Figure 4: ajouter un conge qui debut au milieu de ancien conge et fini a une date depase le ancien fin date

Gestion des employes et des congés

Employe Holiday

Nom:

Prenom:

Email:

Telephone:

Salaire:

Role: ADMIN

Poste: PILOT

ID	Nom	Prenom	Email	Telephone	Salaire	Role	Poste	solde
33	koubi	mohamed-yassine	mohamedyassi...	0645593798	15000.0	ADMIN	TEAM_LEADER	16

Ajouter Modifier Supprimer Afficher Create Acconte

Figure 5: le solde devient 16 par ce que le conge debut au 31 jusqu'a le 5

Gestion des employes et des congés

Employe | **Holiday**

Nom de l'employé: 33 - koubi mohamed-yassine

Date de debut: 2024-12-28

Date de fin: 2024-12-30

Type: CONGE_PAYE

ID	nom_employe	date_debut	date_fin	type
71	33 - koubi mohamed-yassine	2024-12-27	2024-12-31	CONGE_PAYE
72	33 - koubi mohamed-yassine	2024-12-31	2025-01-05	CONGE_PAYE

Ajouter Modifier Supprimer Afficher

Figure 6: si le conge debut et fini en milieu d'un conge deja existe l'application donner un error

Gestion des employes et des congés

Employe | **Holiday**

Nom de l'employé: 33 - koubi mohamed-yassine

Date de debut:

Date de fin:

Type: CONGE_PAYE

ID	nom_employe	date_debut	date_fin	type
71	33 - koubi mohamed-yassine	2024-12-27	2024-12-31	CONGE_PAYE
72	33 - koubi mohamed-yassine	2024-12-31	2025-01-05	CONGE_PAYE

Ajouter Modifier Supprimer Afficher

Erreur

Holiday n'a pas ete ajoutee.

OK

Figure 7: Error d'ajouter le conge'