

**INSTITUT DE LA FRANCOPHONIE POUR L'INFORMATIQUE  
(IFI)**

\*\*\*\*\*



**RAPPORT DE TRAVAUX PRATIQUE 1**

**GESTION DES TACHES**

**Rédigé par :**

BIAKOTA BOMBIA Herbert Cephas – P21

## **Introduction :**

Dans le cadre du premier travail pratique (TP1) de génie logiciel, il nous a été demandé de réaliser **Un petit gestionnaire de tâches** afin de nous rappeler les concepts de la modélisation avec UML et programmation **orientée objet** avec Java et en plus, de nous familiariser avec un environnement de développement intégré (IDE) « libre » (Open Source) à l'exemple d'ECLIPSE. Ainsi ce présent rapport décrit les étapes de conception et de réalisation de l'application.

## **Spécification:**

Les spécifications de l'application à réaliser sont comme suits :

Nous devons réaliser un petit gestionnaire de tâches pour une équipe de travail. Ce gestionnaire fournit à l'utilisateur les fonctionnalités suivantes:

1. Créer, modifier, supprimer, ajouter une tâche
2. Créer, modifier, supprimer, ajouter un membre
3. Assigner une tâche à un membre
4. Chercher et afficher tous les tâches assignées à un membre (par son ID)
5. Chercher et afficher tous les tâches en fonction de leur status (avec le nom du assigné)

Une tâche est composée de ces informations suivantes :

- ID
- Nom
- Une description
- Status : nouveau, en-progrès, terminé.

Un membre est composé de ces informations suivantes :

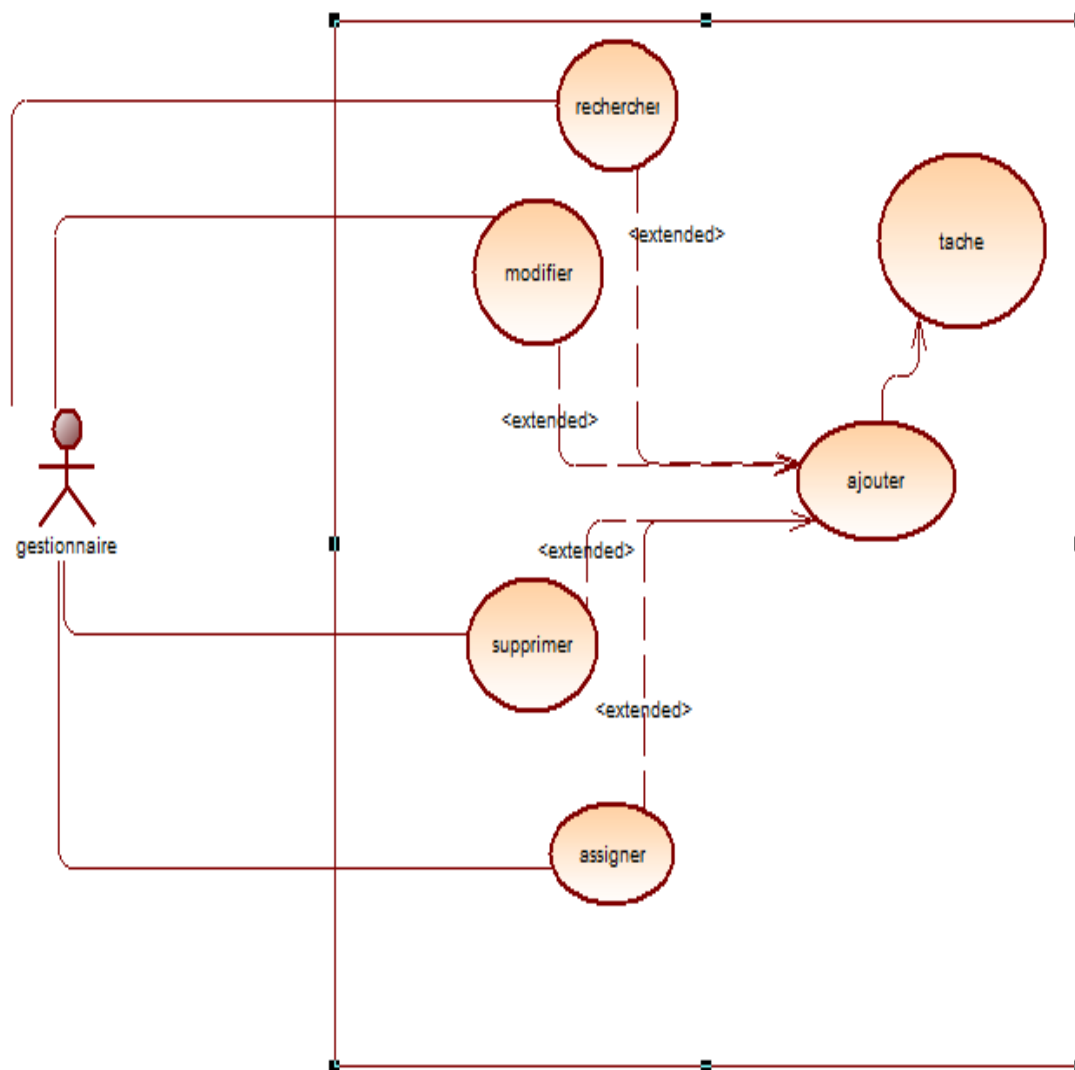
- ID
- Nom.

### **1.2 Contraintes de programmation**

- Il faut programmer de façon orientée objet
- Il faut assurer que la communication entre l'utilisateur et le système est conviviale.
- Les consignes de codage doivent être respectées

## Conception :

La conception de notre application est décrite par les diagrammes ci-après :



**Fig1 : Diagramme de cas d'utilisation**

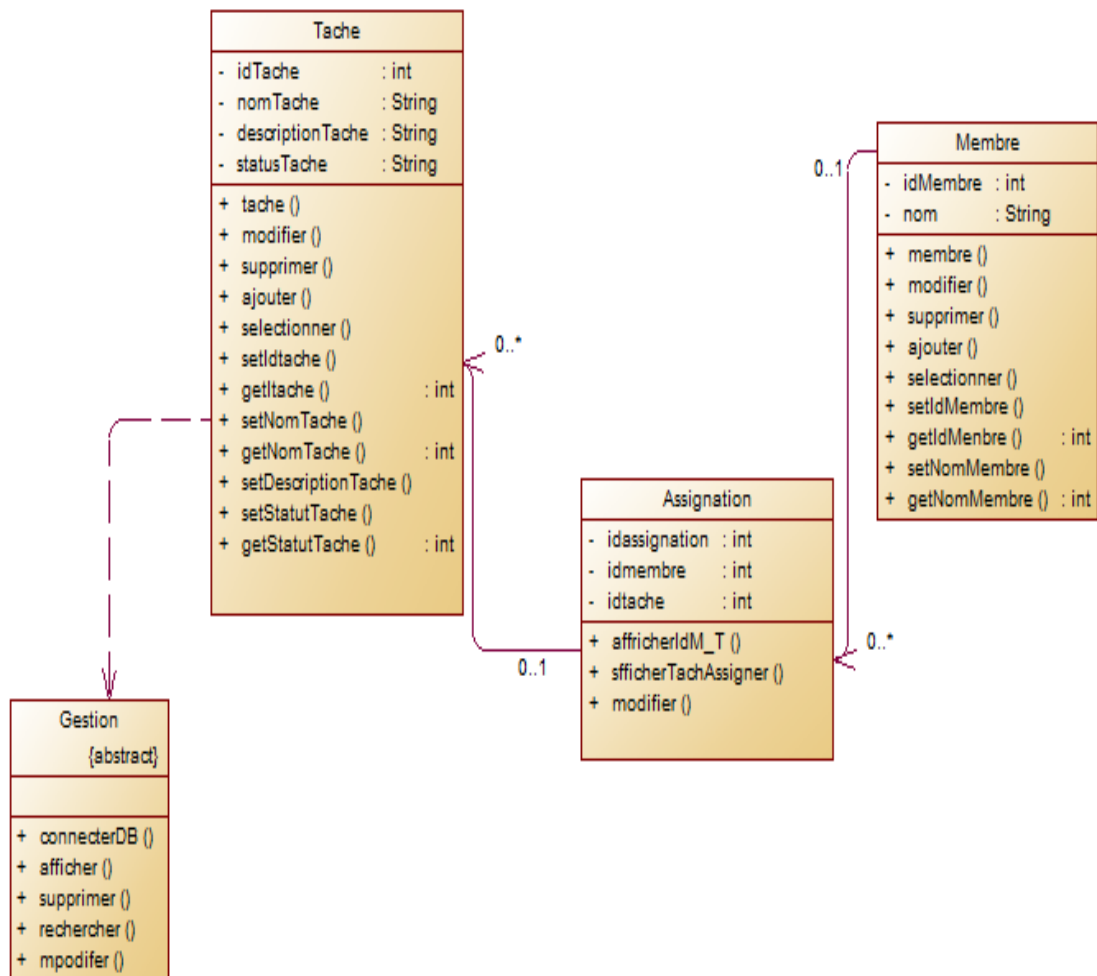


Fig2 : Diagramme de classe

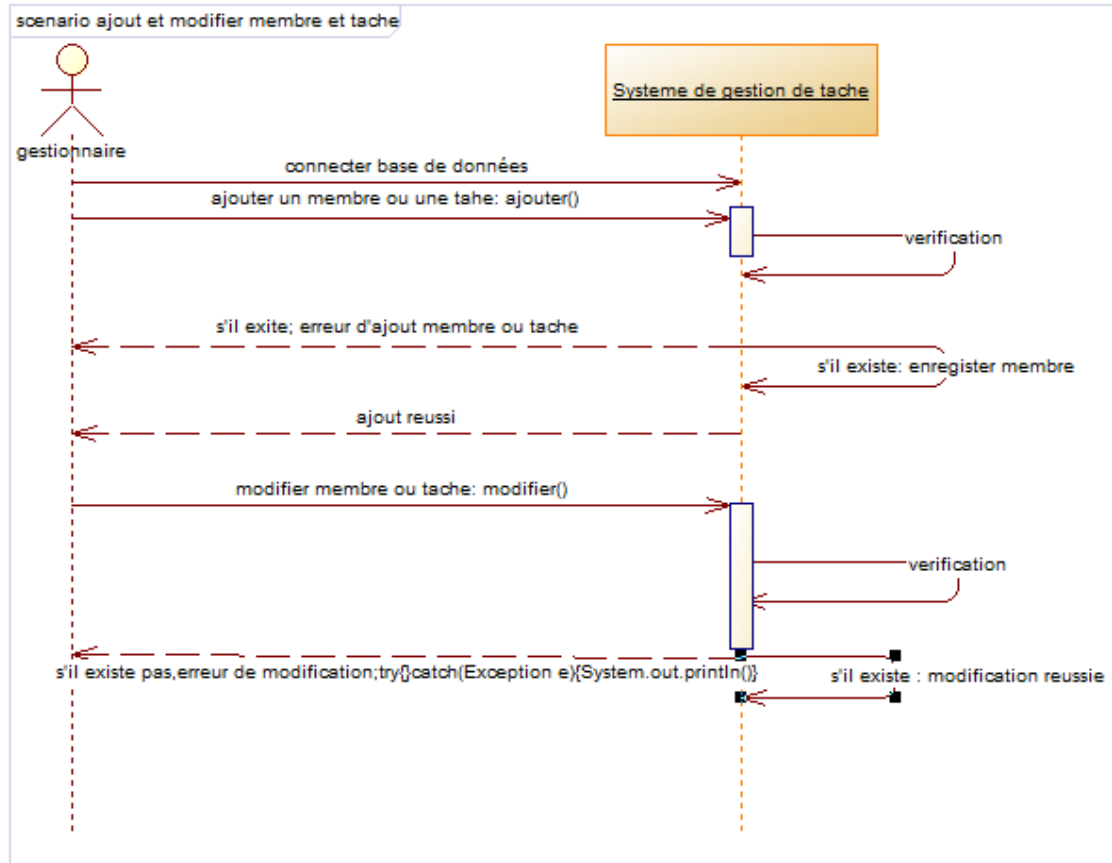
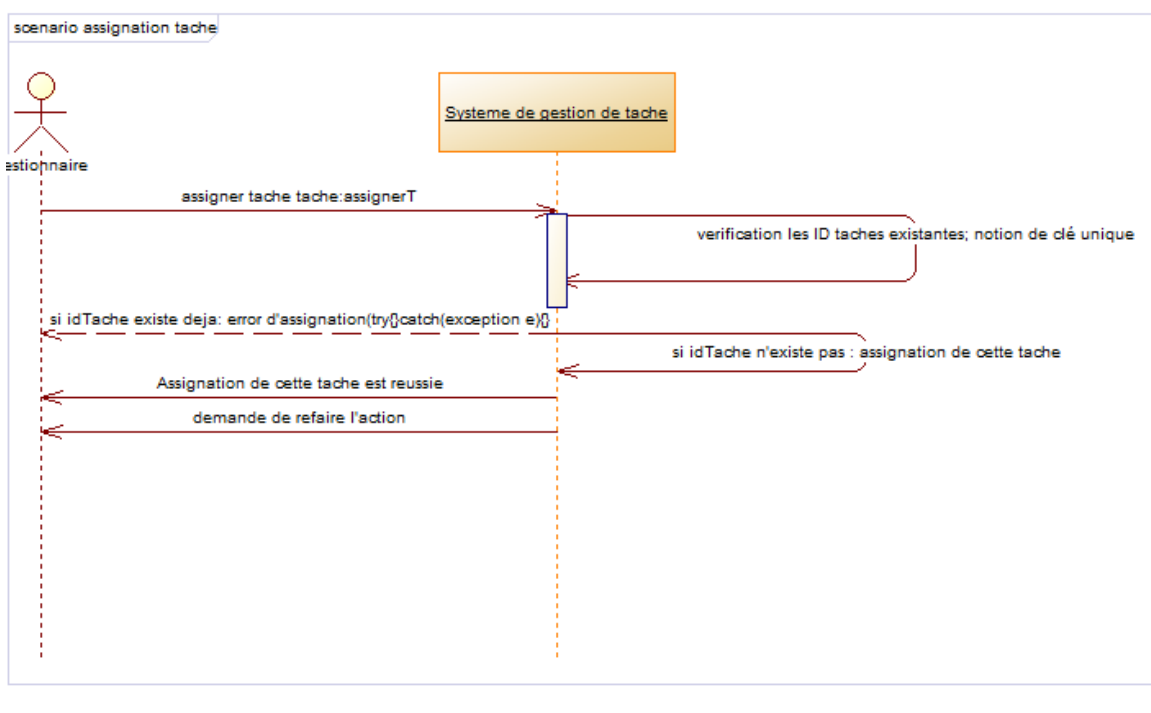


Fig3 : Diagramme de séquences (méthode ajouter : membre ou taches)



**Fig4 : Diagramme de séquences (méthode Assignment : taches)**

## Implémentation et tests :

### Implémentation :

Le programme que nous avons développé sous Linux (14.04) avec le langage de programmation **JAVA** et grâce à l'IDE Eclipse.

D'après notre conception nous avons décidé d'implémenter notre application avec 7 classes et d'utiliser la base de données avec la mysql

NB : à priori nous devons installer mysql pour charger la base de données **gestiontache1.sql** et importer la librairy **com.mysql.jdbc\_5.1.5**

La classe **Membre** qui permet d'effectuer les opérations liées au membre tel que : ajouter, modifier, supprimer et rechercher

La classe **Tache** qui permet d'effectuer les manipulations possibles concernant les tâches nous avons : ajouter, supprimer, modifier, rechercher

La classe **Assignment** permet de gérer l'assignation des tâches au membre : assigner, modifierTacheAssigner

La classe **Menu** permet de l'affichage de la page d'accueil contenant le menu

La classe **Sousmenu** permet de gérer l'appel aux différentes méthodes de la classe membre ou tâche

Classe **Gestion** est la classe abstraite sur laquelle sont héritées les classes filles

Classe **Principal** est la classe contenant notre fonction principale

### Résultat attendu :

Les tests réalisés sur notre application sont résumés par les captures d'écrans suivantes:

-Création et ajout d'un membre

```
BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES
*****CREATION DU JEU DEMINEUR*****
*****MEMU*****
---veuillez faire votre choix---
    1 : pour Ajouter
    2 : pour Modifier
    3 : pour Supprimer
    4 : pour Rechercher
    5 : pour Assigner

choix : 1

*****
---veuillez faire votre choix---
    1 : pour la tâche
    2 : pour le Membre

choix : 2|
```

Fig5.1 choix d'ajout de membre

Ensuite :

```
driver OK
connexion à la base de données reussie
MEMBRES :
|IDENTIFIANT DES MEMBRES|      |NOM DES MEMBRE|
      1                      biakota
      2                      cephas
Entrer le nom du membre : eliodor
ajout reussi

***AFFICHAGE DE LA BASE DE DONNEES****

driver OK
connexion à la base de données reussie
MEMBRES :
|IDENTIFIANT DES MEMBRES|      |NOM DES MEMBRE|
      1                      biakota
      2                      cephas
      3                      eliodor
Voulez-vous enregistrer un autre membre ? (1:oui/ 0: non) :
1
Entrer le nom du membre : perrault
ajout reussi

***AFFICHAGE DE LA BASE DE DONNEES****

driver OK
connexion à la base de données reussie
MEMBRES :
|IDENTIFIANT DES MEMBRES|      |NOM DES MEMBRE|
      1                      biakota
      2                      cephas
      3                      eliodor
      4                      perrault
```

Fig5.2 :



## Création et ajout des taches :

```
BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES
*****CREATION DU JEU DEMINEUR*****
*****MEMU*****
---veuillez faire votre choix---
    1 : pour Ajouter
    2 : pour Modifier
    3 : pour Supprimer
    4 : pour Rechercher
    5 : pour Assigner

choix : 1

*****
---veuillez faire votre choix---
    1 : pour la tache
    2 : pour le Membre

choix : 1
```

Fig6.1 : choix d'ajout de tache

## Ensuite

```
connexion à la base de données reussie
LES TACHES :
|ID TACHES|      |NOM DES TACHES|      | DESCRIPTION|      | STATUT
nom de tache : ecran
description de tache : changement ecran
Statut de tache : nouveau
connexion à la base de données reussie
tache ajoutée avec succes
connexion à la base de données reussie
LES TACHES :
|ID TACHES|      |NOM DES TACHES|      | DESCRIPTION|      | STATUT
1          ecran          changement ecran          nouveau
deconnexion reussie
Voulez-vous Ajouter une autre tache ? (1:oui/ 0: non) :
1
nom de tache : volume
description de tache : augmenter ou diminuer le volume
Statut de tache : nouveau
connexion à la base de données reussie
tache ajoutée avec succes
connexion à la base de données reussie
LES TACHES :
|ID TACHES|      |NOM DES TACHES|      | DESCRIPTION|      | STATUT
1          ecran          changement ecran          nouveau
2          volume          augmenter ou diminuer le volume          nouveau
deconnexion reussie
Voulez-vous Ajouter une autre tache ? (1:oui/ 0: non) :
1
```

Fig6.2

## Modification de tache

```

nom de tache : mines
description de tache : positionner les mines
Statut de tache : nouveau
connexion à la base de données reussie
tache ajoutée avec succes
connexion à la base de données reussie
LES TACHES :


| ID TACHES | NOM DES TACHES | DESCRIPTION                     | STATUT  |
|-----------|----------------|---------------------------------|---------|
| 1         | ecran          | changement ecran                | nouveau |
| 2         | volume         | augmenter ou diminuer le volume | nouveau |
| 3         | mines          | positionner les mines           | nouveau |


deconnexion reussie
Voulez-vous Ajouter une autre tache ? (1:oui/ 0: non) :
1
nom de tache : gameover
description de tache : fin de la partie
Statut de tache : nouveau
connexion à la base de données reussie
tache ajoutée avec succes
connexion à la base de données reussie
LES TACHES :


| ID TACHES | NOM DES TACHES | DESCRIPTION                     | STATUT  |
|-----------|----------------|---------------------------------|---------|
| 1         | ecran          | changement ecran                | nouveau |
| 2         | volume         | augmenter ou diminuer le volume | nouveau |
| 3         | mines          | positionner les mines           | nouveau |
| 4         | gameover       | fin de la partie                | nouveau |


deconnexion reussie
Voulez-vous Ajouter une autre tache ? (1:oui/ 0: non) :
1

```

fig7.1

```

Entrer l'identifiant du membre à modifier: 1
Entrer l'identifiant de sa tache : 5
Entrer le nouvel identifiant correspondant du membre à modifier: 2
driver OK
connexion reussi
modification d'assignation effectuée avec succes

resultat de la modification

*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****
driver OK
connexion reussi


| TACHES   | MEMBRE  |
|----------|---------|
| son      | cephas  |
| click    | cephas  |
| gameover | eliodor |
| volume   | daniel  |



*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****
driver OK
connexion reussi


| ID TACHES | ID MEMBRES |
|-----------|------------|
| 5         | 2          |
| 1         | 2          |
| 4         | 3          |
| 2         | 1          |


```

Fig7.2

## Modification de Membre

```
BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES
*****CREATION DU JEU DEMINEUR*****
*****MEMU*****
---veuillez faire votre choix---
    1 : pour Ajouter
    2 : pour Modifier
    3 : pour Supprimer
    4 : pour Rechercher
    5 : pour Assigner

choix : 2

*****
---veuillez faire votre choix---
    1 : pour la tache
    2 : pour le Membre

choix : 2
```

Fig8.1 : choix de modifier un membre

```
driver OK
connexion à la base de données reussie
MEMBRES :
|IDENTIFIANT DES MEMBRES|      |NOM DES MEMBRE|
    1                      biakota
    2                      cephas
    3                      eliodor
    4                      perrault
Entre le nom du membre à modifier: biakota
Entre le nouveau nom du membre à modifier: daniel
driver OK
connexion à la base de données reussie
modification effectuée avec succes

***AFFICHAGE DE LA BASE DE DONNEES***

driver OK
connexion à la base de données reussie
MEMBRES :
|IDENTIFIANT DES MEMBRES|      |NOM DES MEMBRE|
    1                      daniel
    2                      cephas
    3                      eliodor
    4                      perrault
Voulez-vous modifier un autre membre ? (1:oui/ 0: non) :
```

Fig8.2

## Suppression de tache

```
BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES
*****CREATION DU JEU DEMINEUR*****
*****MEMU*****
---veuillez faire votre choix---
    1 : pour Ajouter
    2 : pour Modifier
    3 : pour Supprimer
    4 : pour Rechercher
    5 : pour Assigner

choix : 3

*****
---veuillez faire votre choix---
    1 : pour la tache
    2 : pour le Membre

choix : 1
```

Fig9.1 choix de supprimer taches par ID

```
connexion à la base de données reussie
LES TACHES :
|ID TACHES|      |NOM DES TACHES|      | DESCRIPTION|      | STATUT
1          click          click souris          nouveau
2          volume          augmenter ou diminuer le volume          nouveau
3          mines          positionner les mines          nouveau
4          gameover          fin de la partie          nouveau
5          son          chargement de son          nouveau
6          score          incrementation du score          nouveau

Entrer l'identifiant de la tache à supprimer : 6
connexion à la base de données reussie
tache suppriméeconnexion à la base de données reussie
LES TACHES :
|ID TACHES|      |NOM DES TACHES|      | DESCRIPTION|      | STATUT
1          click          click souris          nouveau
2          volume          augmenter ou diminuer le volume          nouveau
3          mines          positionner les mines          nouveau
4          gameover          fin de la partie          nouveau
5          son          chargement de son          nouveau
deconnexion reussie
Voulez-vous Supprimer une autre tache ? (1:oui/ 0: non) :
```

Fig9.1

Suppression de membre :

```
BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES
*****CREATION DU JEU DEMINEUR*****
*****MEMU*****
---veuillez faire votre choix---
    1 : pour Ajouter
    2 : pour Modifier
    3 : pour Supprimer
    4 : pour Rechercher
    5 : pour Assigner

choix : 3

*****
---veuillez faire votre choix---
    1 : pour la tache
    2 : pour le Membre

choix : 2
```

fig10.1 choix de supprimer un membre par ID

```
driver OK
connexion à la base de données reussie
MEMBRES :
|IDENTIFIANT DES MEMBRES|      |NOM DES MEMBRE|
    1                      daniel
    2                      cephas
    3                      eliodor
    4                      perrault

Entrer l'identifiant du membre à supprimer : 4
driver OK
connexion à la base de données reussie
membre supprimé
***AFFICHAGE DE LA BASE DE DONNEES***

driver OK
connexion à la base de données reussie
MEMBRES :
|IDENTIFIANT DES MEMBRES|      |NOM DES MEMBRE|
    1                      daniel
    2                      cephas
    3                      eliodor
```

Fig10.2

## Recherche de taches pas statut

```
BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES
*****CREATION DU JEU DEMINEUR*****
*****MEMU*****
---veuillez faire votre choix---
    1 : pour Ajouter
    2 : pour Modifier
    3 : pour Supprimer
    4 : pour Rechercher
    5 : pour Assigner

choix : 4

*****
---veuillez faire votre choix---
    1 : pour la tache
    2 : pour le Membre

choix : 1
```

Fig11.1 : choix de rechercher taches par statut

```
connexion à la base de données reussie
LES TACHES :
|ID TACHES|      |NOM DES TACHES|      | DESCRIPTION|      | STATUT|
    1      click      click souris      nouveau
    2      volume      augmenter ou diminuer le volume      nouveau
    3      mines      positionner les mines      nouveau
    4      gameover      fin de la partie      nouveau
    5      son      chargement de son      nouveau

****RECHERCHE DE TACHE EN FONCTION DE LEUR TACHE****
Veuillez entrer le statut des taches : nouveau
Resultat de la recherche :
connexion à la base de données reussie
|TACHES|      |STATUT|
click      nouveau
gameover      nouveau
mines      nouveau
son      nouveau
volume      nouveau

deconnexion reussie
Voulez-vous effecuer une autre recherche ? (1:oui/ 0: non) :
```

Fig11.2 :

Assignment de tache :

```
BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES
*****CREATION DU JEU DEMINEUR*****
*****MEMU*****
---veuillez faire votre choix---
    1 : pour Ajouter
    2 : pour Modifier
    3 : pour Supprimer
    4 : pour Rechercher
    5 : pour Assigner

choix : 5

*****
---veuillez faire votre choix---
    1 : pour Modifier l'assignation
    2 : pour assigner tache à un membre

choix : 2
```

fig12.1 : choix d'assignation de tache aux membres

```
driver OK
connexion à la base de données reussie
MEMBRES :
|IDENTIFIANT DES MEMBRES|      |NOM DES MEMBRE|
    1                      daniel
    2                      cephas
    3                      eliodor
connexion à la base de données reussie
LES TACHES :
|ID TACHES|      |NOM DES TACHES|      | DESCRIPTION|      | STATUT|
    1      click      click souris      nouveau
    2      volume      augmenter ou diminuer le volume      nouveau
    3      mines      positionner les mines      nouveau
    4      gameover      fin de la partie      nouveau
    5      son      chargement de son      nouveau

****LES TACHES ASSIGNEES A CHAQUE MEMBRE****
driver OK
connexion reussi

      |TACHES|      |MEMBRE|

****LES TACHES ASSIGNEES A CHAQUE MEMBRE****
driver OK
connexion reussi

      |ID TACHES|      |ID MEMBRES|
```

fig12.2 :

```

ENTRER L'IDENTIFIANT DU MEMBRE : 1
ENTRER L'IDENTIFIANT DU TACHE : 5
driver OK
connexion reussi
Assignation reussie

resultat de l'assignation

*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****
driver OK
connexion reussi

```

TACHES	MEMBRE
son	daniel

```

*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****
driver OK
connexion reussi

```

ID TACHES	ID MEMBRES
5	1

```

deconnexion reussie
Voulez-vous faire une autre assignation ? (1:oui/ 0: non) :
1

```

Fig :12.3 :

```

ENTRER L'IDENTIFIANT DU MEMBRE : 2
ENTRER L'IDENTIFIANT DU TACHE : 1
driver OK
connexion reussi
Assignation reussie

resultat de l'assignation

*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****
driver OK
connexion reussi

```

TACHES	MEMBRE
son	daniel
click	cephas

```

*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****
driver OK
connexion reussi

```

ID TACHES	ID MEMBRES
5	1
1	2

```

deconnexion reussie
Voulez-vous faire une autre assignation ? (1:oui/ 0: non) :
1

```

Fig12.4 :



```

ENTRER L'IDENTIFIANT DU MEMBRE : 1
ENTRER L'IDENTIFIANT DU TACHE : 2
driver OK
connexion reussi
Assignation reussie

resultat de l'assignation

*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****
driver OK
connexion reussi

          |TACHES|
          son
          click
          gameover
          volume

          |MEMBRE|
          daniel
          cephas
          eliodor
          daniel

*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****
driver OK
connexion reussi

          |ID TACHES|
          5
          1
          4
          2

          |ID MEMBRES|
          1
          2
          3
          1

deconnexion reussie
Voulez-vous faire une autre assignation ? (1:oui/ 0: non) :
0

```

Fig12.5

Modification d'assignation de tache :

```

BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES
*****CREATION DU JEU DEMINEUR*****
*****MEMU*****
---veuillez faire votre choix---
      1 : pour Ajouter
      2 : pour Modifier
      3 : pour Supprimer
      4 : pour Rechercher
      5 : pour Assigner

choix : 5

*****
---veuillez faire votre choix---
      1 : pour Modifier l'assignation
      2 : pour assigner tache à un membre

choix : 1

```

Fig :12.6 : choix de modification d'assignation de tache

```

*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****
driver OK
connexion reussi

      | ID TACHES |      | ID MEMBRES |
      | 5        |      | 1          |
      | 1        |      | 2          |
      | 4        |      | 3          |
      | 2        |      | 1          |

*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****
driver OK
connexion reussi

      | TACHES |      | MEMBRE |
      | son   |      | daniel |
      | click |      | cephas |
      | gameover |      | eliodor |
      | volume |      | daniel |

```

fig :12.7

```

Entrer l'identifiant du membre à modifier: 1
Entrer l'identifiant de sa tache : 5
Entrer le nouvel identifiant correspondant du membre à modifier: 2
driver OK
connexion reussi
modification d'assignation effectuée avec succes

resultat de la modification

*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****
driver OK
connexion reussi

      | TACHES |      | MEMBRE |
      | son   |      | cephas |
      | click |      | cephas |
      | gameover |      | eliodor |
      | volume |      | daniel |

*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****
driver OK
connexion reussi

      | ID TACHES |      | ID MEMBRES |
      | 5        |      | 2          |
      | 1        |      | 2          |
      | 4        |      | 3          |
      | 2        |      | 1          |

```

fig12.8

L'implémentation (utilisant JUnit pour le test unitaire)

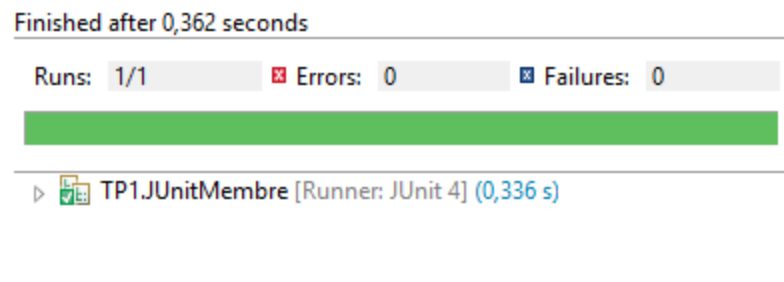


Fig13.1 : Test unitaire sur la méthode afficher() de la classe Membre :

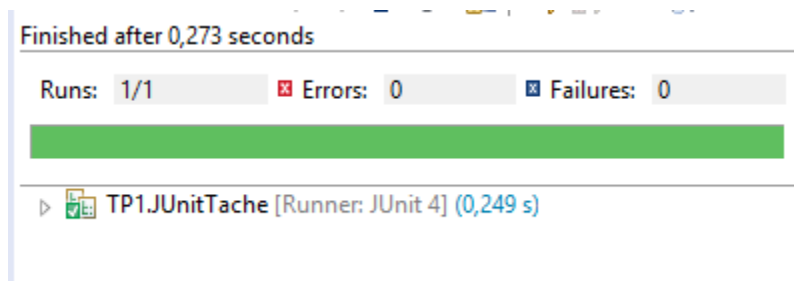


Fig13.2 : image de test unitaire sur la méthode afficher de la classe tache

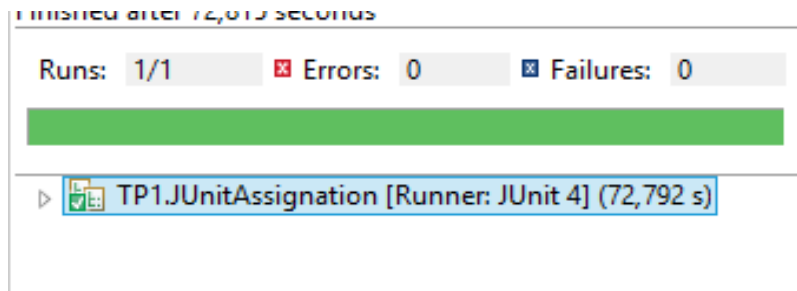
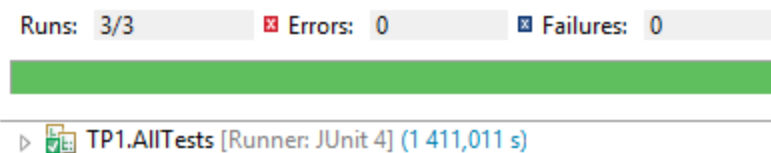


Fig13.2 : test unitaire sur la méthode de la classe tache

Test global :



## **Conclusion :**

A l'issu de ce Travail personnel nous avons pour atteindre les objectifs attendu tout en améliorant notre niveau compréhension de la programmation orientée objet avec java sur eclipse. Ainsi les spécifications mentionnées ci-haut ont été respectées et notre application de gestion de tache (création du jeu le démineur) est fonctionnelle. Pour ce fait il nous reste encore des efforts à fournir dans le sens de l'orientée objet et de la bonne maîtrise du langage java et sans oublier aussi le bonnes pratiques de programmation

Annexes :

### **Classe Membre**

```
package TP1;

import java.sql.*;

import java.util.Scanner;

public class Membre extends Assignment{

    private int ID;

    private String nomMembre;

    private String sql;

    String url="jdbc:mysql://localhost/gestiontachetp1";

    String user="root";

    String password="";

    Statement st = null;

    ResultSet rs = null;

    Connection cn = null;

    Scanner sc ;
```

```

public int getID() {
    return ID;
}

public void setID(int iD) {
    ID = iD;
}

public String getNom() {
    return nomMembre;
}

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

@Override
public void connecterDB() {
    // TODO Auto-generated method stub
    try{
        // Chargement du clavier
        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("driver OK");

        cn = DriverManager.getConnection(url,user,password);
        st = cn.createStatement();
        System.out.println("connexion à la base de données
reussie");
    }catch (Exception e){
        System.out.println(" error de connexion à la base de
données!");
    }
}
}

```

```

@Override

public Boolean modifier(){

    int reponse = 1;

    while (reponse == 1) {

        sc = new Scanner(System.in);

        System.out.print("Entre le nom du membre à modifier: ");

        nomMembre=sc.nextLine();

        System.out.print("Entre le nouveau nom du membre à modifier:

");

        String nouveauNomMembre=sc.nextLine();

        connecterDB();

        try{

            sql = "UPDATE membre SET `nommembre` =

'" +nouveauNomMembre +"' WHERE nommembre ='" +nomMembre+"'";

            st.executeUpdate(sql);

            System.out.println("modification effectuée avec

succes");

            afficher();

        }catch(Exception e){

            System.out.println("echec de modification");

        }finally{

            try{

                cn.close();

                st.close();

            }catch (Exception e){

                System.out.println("deconnexion echouée");

            }

        }

        reponse = 2;

```

```

while(reponse != 1 && reponse != 0)

{

    //On demande si la personne veut faire un autre essai

    System.out.println("Voulez-vous modifier un autre membre
? (1:oui/ 0: non) : " );

    reponse = sc.nextInt();

}

return true;

}

@Override

public Boolean afficher() {

    System.out.println("");

    System.out.println("***AFFICHAGE DE LA BASE DE
DONNEES***");

    System.out.println("");

    connecterDB();

    try{

        sql= "SELECT * FROM gestiontachetp1.membre";

        rs= st.executeQuery(sql);

        System.out.println("MEMBRES : ");

        System.out.println("| IDENTIFIANT DES
MEMBRES|\t|NOM DES MEMBRE|");

        while (rs.next()){

            System.out.print("\t");

            System.out.print(rs.getInt("idmembre")+"\t");

```

```

        System.out.print("\t\t\t");

        System.out.print(rs.getString("nommembre")+"\t");

        System.out.println("");

    }

    }

    catch (Exception e){

        System.out.println("echec d'affichage");

    }

    return true;

}

@Override

public void supprimer(){

    sc = new Scanner(System.in);

    System.out.println("");

    System.out.print("Entrer l'identifiant du membre à supprimer
: ");

    ID=sc.nextInt();

    connecterDB();

    try{

        sql ="DELETE FROM gestiontachetpl.membre WHERE
membre.idmembre='"+ID+"' ";

        st.executeUpdate(sql);

        System.out.print("membre supprum ");

        afficher();

    }

    catch(Exception e){

        System.out.print("echec de suppression de ce membre");

    }

}

@Override

```



```

public Boolean ajouter(){

    int reponse = 1;

    while (reponse == 1) {

        sc = new Scanner(System.in);

        System.out.print("Entrer le nom du membre : ");

        nomMembre=sc.nextLine();

        try{

            sql = "INSERT INTO
gestiontachetp1.membre(`idmembre`,`nommembre`) VALUES ('"+ ID
+"', '"+nomMembre+"')";

            st.executeUpdate(sql);

            System.out.println("ajout reussi");

            //connecterDB();

            afficher();

        }catch (Exception e){

            System.out.println("error d'ajout de membre");

        }

        reponse = 2;

        while(reponse != 1 && reponse != 0)

            {

                //On demande si la personne veut faire un autre essai

                System.out.println("Voulez-vous enregistrer un autre
membre ? (1:oui/ 0: non) : " );

                reponse = sc.nextInt();

            }

        }

        return true;

    }

    @Override

    public void rechercher(){

        int reponse = 1;

```

```

while (reponse == 1) {

    sc = new Scanner(System.in);

    System.out.println("");

    System.out.println("Recherche de toutes les tache assigné à
un membre : ");

    System.out.print("Entrer l'identifiant : ");

    ID=sc.nextInt();

    System.out.println("Resultat de la recherche : ");

    connecterDB();

    try{

        sql= "SELECT tache.nomtache, membre.nommembre FROM
gestiontachetp1.tache, gestiontachetp1.membre,
gestiontachetp1.assignment WHERE assignment.idmembre =
membre.idmembre AND assignment.idtache = tache.idtache AND
assignment.idmembre='"+ID+"'";

        rs= st.executeQuery(sql);

        System.out.println("|TACHES|\t\t|MEMBRE|");

        while (rs.next()){

            System.out.println(rs.getString("tache.nomtache")+"\t\t"+
rs.getString("membre.nommembre"));

        }

        System.out.println("");

    }catch (Exception e){

        System.out.println("taches assignées introuvables");

    }finally{

        try{

            cn.close();

            st.close();

            System.out.println("deconnexion reussie");

        }catch (Exception e){

            System.out.println("deconnexion echouée");

        }

    }

}

```

```

        reponse = 2;

        while(reponse != 1 && reponse != 0)
        {
            //On demande si la personne veut faire un autre essai

            System.out.println("Voulez-vous enregistrer une autre
recherche ? (1:oui/ 0: non) : " );

            reponse = sc.nextInt();
        }
    }
}

```

### Classe tache :

```

package TP1;

import java.sql.*;
import java.util.Scanner;

public class Tache extends Gestion {

    private int ID;

    private String nomTache;

    private String description;

    private String statut;

    private int idmembre;

    String sql;

    String url="jdbc:mysql://localhost/gestiontachetp1";

    String user="root";

    String password="";

    Connection cn = null;

    Statement st = null;

    ResultSet rs = null;

    Scanner sc = new Scanner(System.in);

    @Override

```

```

public void connecterDB() {

    // TODO Auto-generated method stub

    try{

        // Chargement du clavier

        Class.forName("com.mysql.jdbc.Driver");

        // System.out.println("driver OK");

        cn =
DriverManager.getConnection(url,user,password);

        st = cn.createStatement();

        System.out.println("connexion à la base de
données reussie");

    }catch (Exception e){

        System.out.println(" error de connexion à la
base de données!");

    }

}

@Override

public Boolean afficher() {

    // TODO Auto-generated method stub

    connecterDB();

    try{

        sql= "SELECT * FROM gestiontachetpl.tache ";

        rs= st.executeQuery(sql);

        System.out.println("LES TACHES : ");

        System.out.println("|ID TACHES|\t|NOM DES
TACHES|\t\t| DESCRIPTION|\t\t\t\t| STATUT");

        while (rs.next()){

            System.out.print("\t");

            System.out.print(rs.getInt("idtache")+"\t");

            System.out.print(rs.getString("nomtache")+"\t\t\t");

```

```

        System.out.print(rs.getString("descriptiontache")+"\t\t\t\t\t");

        System.out.print(rs.getString("statustache")+"\t\t");

        System.out.println("");

    }

}

catch (Exception e){

    System.out.println("echec d'affichage");

}

return true;

}

//les getters et setters
public int getID() {

    return ID;

}

public void setID(int iD) {

    ID = iD;

}

public String getNom() {

    return nomTache;

}

public void setNom(String nom) {

    this.nomTache = nom;

}

public String getDescription() {

    return description;

}

public void setDescription(String description) {

    this.description = description;

```

```

    }

    public String getStatut() {
        return statut;
    }

    public void setStatut(String statut) {
        this.statut = statut;
    }

    public String getNomTache() {
        return nomTache;
    }

    public void setNomTache(String nomTache) {
        this.nomTache = nomTache;
    }

    public int getIdmembre() {
        return idmembre;
    }

    public void setIdmembre(int idmembre) {
        this.idmembre = idmembre;
    }

    // les methodes

    //mettre à jour la base de donnée

    @Override

    public Boolean modifier( ){

        int reponse = 1;

        while (reponse == 1) {

            System.out.print("ancienne tache : ");

            String ancienNomTache = sc.nextLine();

            System.out.print("ancienne description : ");

            String ancienDescription = sc.nextLine();

```

```

        System.out.print("ancien statut : ");

        String ancienStatut = sc.nextLine();

        System.out.print("Nouvelle tache : ");

        String nouveauNomTache = sc.nextLine();

        System.out.print("Nouvelle description : ");

        String nouvelleDescription = sc.nextLine();

        System.out.print("nouveau status : ");

        String nouveauStatut=sc.nextLine();

        connecterDB();

        try{

            //requete pour la mise à jour des taches

            sql = " UPDATE tache SET
nomtache='"+nouveauNomTache+"',descriptiontache='"+nouvelleDescription+
"',statustache='"+nouveauStatut+"' WHERE nomtache ='"+ancienNomTache+"'
AND descriptiontache='"+ancienDescription+"' AND
statustache='"+ancienStatut+"'";

            st.executeUpdate(sql);

            System.out.println("modification statut reussi");

            afficher();

        }catch (Exception e){

            System.out.println("modification statut echoué !");

        }finally{

            try{

                //fermeture de la connexion à la base

                cn.close();

                st.close();

                System.out.println("deconnexion reussie");

            }catch (Exception e){

                System.out.println("deconnexion echouée");

            }

        }

        reponse = 2;

```

```

        while(reponse != 1 && reponse != 0)
        {
            //On demande si la personne veut faire un autre essai

            System.out.println("Voulez-vous modifier une autre tache
? (1:oui/ 0: non) : " );

            reponse = sc.nextInt();

        }

    }

    return true;

}

//supprimer une information dans la base de données
public void supprimer(){

    int reponse = 1;

    while (reponse == 1) {

        System.out.println("");

        System.out.print("Entrer l'identifiant de la tache à
supprimer : ");

        ID=sc.nextInt();

        connecterDB();

        try{

            sql ="DELETE FROM gestiontachepl.tache WHERE
tache.idtache='"+ID+"'";

            st.executeUpdate(sql);

            System.out.print("tache supprimée");

            afficher();

        }

        catch(Exception e){

            System.out.print("echec de suppression de cette
tache");

        }finally{

            try{

                //fermeture de la connexion à la base

```



```

        cn.close();

        st.close();

        System.out.println("deconnexion reussie");
    }catch (Exception e){
        System.out.println("deconnexion echouée");
    }

    reponse = 2;

    while(reponse != 1 && reponse != 0)
    {
        //On demande si la personne veut refaire l'action
        System.out.println("Voulez-vous Supprimer une autre
tache ? (1:oui/ 0: non) : " );

        reponse = sc.nextInt();
    }
}

}

//ajouter une information dans la base de donnée
public Boolean ajouter(){
    int reponse = 1;
    while (reponse == 1) {
        sc = new Scanner(System.in);
        System.out.print("nom de tache : ");
        nomTache = sc.nextLine();//saisie au clavier
        System.out.print("description de tache : ");
        description = sc.nextLine();//saisie au clavier
        System.out.print("Statut de tache : ");
        statut = sc.nextLine();//saisie au clavier
        connecterDB();
        try{

```

```

        //requete pour l'enregistrment des taches

        sql = "INSERT INTO gestiontachepl.tache
        (`idtache`,`nomtache`,`descriptiontache`,`statustache`) VALUES ('"+ ID
        +"', '"+ nomTache +"', '"+ description +"', '"+ statut +'")";

        st.executeUpdate(sql);

        System.out.println("tache ajoutée avec succes");

        afficher();

    }catch(Exception e){

        System.out.println("erreur ajout tache");

    }finally{

        try{

            //fermeture de la connexion à la base

            cn.close();

            st.close();

            System.out.println("deconnexion reussie");

        }catch (Exception e){

            System.out.println("deconnexion echouée");

        }

    }

    reponse = 2;

    while(reponse != 1 && reponse != 0)

    {

        //On demande si la personne veut faire un autre essai

        System.out.println("Voulez-vous Ajouter une autre tache
        ? (1:oui/ 0: non) : " );

        reponse = sc.nextInt();

    }

}

return true;

}

```

```

//recherche un information dans la base de données

public void rechercher(){

    int reponse = 1;

    while (reponse == 1) {

        sc = new Scanner(System.in);

        System.out.println("");

        System.out.println("*****RECHERCHE DE TACHE EN FONCTION DE
LEUR TACHE*****");

        System.out.print("Veuillez entrer le statut des taches : ");

        statut=sc.nextLine(); //saisie au clavier

        System.out.println("Resultat de la recherche : ");

        connecterDB();

        try{

            //requete pour la recherche des taches par leur
statuts

            sql= "SELECT `nomtache` , `statustache` FROM tache
WHERE statustache ='"+statut+"' ";

            //execution de la requette

            rs= st.executeQuery(sql);

            System.out.println("|TACHES|\t\t|STATUT|");

            while (rs.next()){

                System.out.println(rs.getString("nomtache")+"\t\t"+
rs.getString("statustache"));

            }

            System.out.println("");

        }catch (Exception e){

            System.out.println("taches introuvables");

        }finally{

            try{

                //fermeture de la connexion à la base

                cn.close();

                st.close();

```

```

        System.out.println("deconnexion reussie");
    }catch (Exception e){
        System.out.println("deconnexion echouée");
    }
}

reponse = 2;
while(reponse != 1 && reponse != 0)
{
    //On demande si la personne veut faire un autre essai
    System.out.println("Voulez-vous effectuer une autre recherche
? (1:oui/ 0: non) : " );
    reponse = sc.nextInt();

}

}

}

}

```

## Classe assignation

```

package TP1;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Scanner;

public class Assignation extends Tache {

    String url="jdbc:mysql://localhost/gestiontachetp1";

    String user="root";

    String password="";

    Statement st = null;

    ResultSet rs = null;

```

```

ResultSet rsM = null;

Connection cn = null;

int idassign;

String sql;

Scanner sc = new Scanner(System.in);

public void connecterDB() {

    // TODO Auto-generated method stub

    try{

        // Chargement du clavier

        Class.forName("com.mysql.jdbc.Driver");

        System.out.println("driver OK");


        cn = DriverManager.getConnection(url,user,password);

        st = cn.createStatement();

        System.out.println("connexion reussi");

    }catch (Exception e){

        System.out.println(" error de connexion");

    }

}

public void affichertachAssign(){

    System.out.println("");

    System.out.println("*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****");

    connecterDB();

    try{

        sql= "SELECT tache.nomtache , membre.nommembre FROM
gestiontachetpl.tache, gestiontachetpl.membre,
gestiontachetpl.assignation WHERE membre.idmembre=
assignation.idmembre AND tache.idtache=assignation.idtache ";

        rs= st.executeQuery(sql);

        System.out.println("\t\t\tTACHES\t\t\tMEMBRE");

        while (rs.next()){

```

```
        System.out.println("\t\t\t"+rs.getString("tache.nomtache")+"\t\t\t\t\t\n"+ rs.getString("membre.nommembre")) ;  
  
                //System.out.println(rs.getString("tache.nom  
"));)  
  
    }  
  
    System.out.println("");  
  
}catch (Exception e){  
  
    System.out.println("taches assignées introuvables");  
  
}  
  
}
```

```
public void afficherIdM_T(){  
  
    System.out.println("");  
  
    System.out.println("*****LES TACHES ASSIGNEES A CHAQUE MEMBRE*****");  
  
    connecterDB();  
  
    try{  
  
        sql= "SELECT assignment.idtache,  
assignment.idmembre FROM gestiontachetpl.assignment ";  
  
        rs= st.executeQuery(sql);  
  
        System.out.println("\t\t\t|ID TACHES|\t\t\t|ID MEMBRES|");  
  
        while (rs.next()){  
  
            System.out.println("\t\t\t"+rs.getInt("assignment.idtache")+"\t\t\t\t\t\n"+ rs.getInt("assignment.idmembre")) ;  
  
                //System.out.println(rs.getString("tache.nom  
"));)  
  
        }  
  
        System.out.println("");  
  
    }  
  
    catch (Exception e){  
  
        System.out.println("taches assignées introuvables");  
  
    }
```

```

    }

    public Boolean modifier() {
        int reponse = 1;
        while (reponse == 1) {
            sc = new Scanner(System.in);

            System.out.print("Entrer l'identifiant du membre à
modifier: ");

            int idMA = sc.nextInt();

            System.out.print("Entrer l'identifiant de sa tache :
");

            int idT = sc.nextInt();

            System.out.print("Entrer le nouvel identifiant
correspondant du membre à modifier: ");

            int newIdM = sc.nextInt();

            connecterDB();

            try{

                sql = "UPDATE assignation SET assignation.idmembre
='"+newIdM+"' WHERE assignation.idtache = '"+idT+"' AND
assignation.idmembre='"+idMA+"' ";

                st.executeUpdate(sql);

                System.out.println("modification d'assignation
effectuée avec succes");

                System.out.println("");

                System.out.println("resultat de la modification ");

                afficherTachAssign();

                afficherIdM_T();

            }catch(Exception e){

                System.out.println("echec de modification
d'assignation");

            }finally{

                try{

                    cn.close();

                    st.close();

                }catch (Exception e){

```

```

        System.out.println("deconnexion echouée");
    }
}

reponse = 2;

while(reponse != 1 && reponse != 0)
{
    //On demande si la personne veut faire un autre essai

    System.out.println("Voulez-vous une autre modification
d'assignation de tache ? (1:oui/ 0: non) : " );

    reponse = sc.nextInt();
}

}

return true;
}

public void assignerT(){
    int reponse = 1;
    while (reponse == 1) {

        System.out.print("ENTRER L'IDENTIFIANT DU MEMBRE : ");

        int idM= sc.nextInt();

        System.out.print("ENTRER L'IDENTIFIANT DU TACHE : ");

        int idT=sc.nextInt();

        connecterDB();

        try{

            sql = "INSERT INTO gestiontachetpl.assignment
(idassignation,idmembre,idtache) VALUES ('"+ idassign
+"', '"+idM+'', '"+idT+'')";

            st.executeUpdate(sql);

            System.out.println("Assignation reussie");

            System.out.println(" ");

            System.out.println("resultat de l'assignation

");

            affichertachAssign();

```



```

        afficherIdM_T();
    }catch (Exception e){
        System.out.println("Echec d'assignation");
    }finally{
        try{
            cn.close();
            st.close();
            System.out.println("deconnexion reussie");
        }catch (Exception e){
            System.out.println("deconnexion echouée");
        }
    }
    reponse = 2;
    while(reponse != 1 && reponse != 0)
    {
        //On demande si la personne veut faire un autre essai
        System.out.println("Voulez-vous faire une autre assignation ?
(1:oui/ 0: non) : " );
        reponse = sc.nextInt();
    }
}
}
}
}

```

### La classe Sous Menu

```

package TP1;

import java.util.Scanner;

public class Sousmenu {

    Scanner sc = new Scanner(System.in);

```





```

        System.out.println("\t\t\t\t\t\t\t---veuillez faire votre
choix---");
        System.out.println("\t\t\t\t\t\t\t 1 : pour Modifier
l'assignation ");
        System.out.println("\t\t\t\t\t\t\t 2 : pour assigner tache à
un membre ");
        System.out.print("\t\t\t\t\t\t\t choix : ");

        int choixM = sc.nextInt();
        if (choixM==1){
            mbre.afficher();
            tach.afficher();
            as.afficherIdM_T();
            as.affichertachAssign();
            as.modifier();

        }else if (choixM==2){

            mbre.afficher();
            tach.afficher();
            as.affichertachAssign();
            as.afficherIdM_T();
            as.assignerT();

        }else{
            System.out.println("veuillez respecter les consignes
de choix : ");
        }
    }
}

```

## Classe Menu

```

package TP1;

import java.util.Scanner;

public class Menu {

    Scanner sc;
    Membre mbre;
    Tache tach = new Tache();
    Assignation as=new Assignation();
    Sousmenu Smenu = new Sousmenu();

    public void menu() {
        mbre=new Membre();
        int choixaction = 0;

        sc= new Scanner(System.in);

        int reponse = 1;
        while (reponse == 1) {
            System.out.println("\t\t\t\t\t\t\tBIENVENU SUR LE PROGRAMME DE
GESTION DES TACHES");

```



```
public abstract class Gestion {  
    public abstract Boolean modifier();  
    public abstract void supprimer();  
    public abstract Boolean ajouter();  
    public abstract void rechercher();  
    public abstract void connecterDB();  
    public abstract Boolean afficher();  
}
```