

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is light green. They are positioned diagonally, with the blue one partially covering the green one.

Projet Web Dynamique

Bibliothèque

Interface Accueil



Interface Adhérents

Gestion des
Adhérents

Fonctionnalités:

- Lister(1)
- Ajouter (2)
- Modifier (3)
- Supprimer (4)

The screenshot shows a web application titled "GESTION DES ADHERENTS" with a "Home" link. Below the title is a button labeled "Ajouter un Adhérent" with a red circle containing the number 2. Below this is a table with 11 columns: N°, Nom, Prenom, Date de Naissance, Adresse, E-Mail, Date d'Adhesion, Téléphone, Catégorie, and two empty columns. The table contains three rows of data. To the right of the table are two circular icons: one with a pencil (labeled 3) and one with a trash can (labeled 4). A red circle with the number 1 is positioned below the table. The background of the interface is a blurred image of book spines.

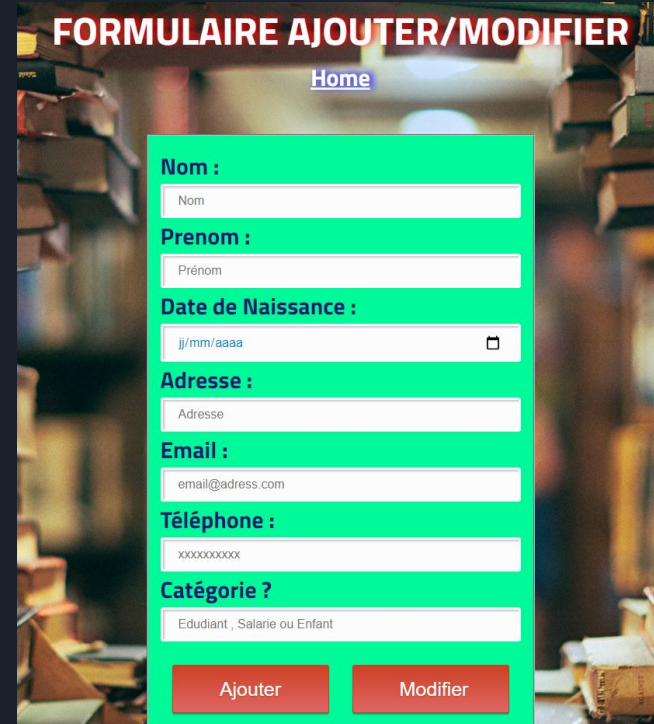
N°	Nom	Prenom	Date de Naissance	Adresse	E-Mail	Date d'Adhesion	Téléphone	Catégorie		
1	Peschard	Francois	1986-01-06	22 rue 14 juillet	francois@gmail.com	2020-11-18	0143624433	Salarie		
2	Wick	John	1974-05-03	213 rue de montmartre	JohnWick@gmail.com	2020-12-17	0632456756	Etudiant		
3	Puyol	Carlos	1978-04-13	2 rue de la paix	CarlosPuyol@gmail.com	2018-08-08	0198535357	Etudiant		

Formulaire Adhérents

Formulaire pour ajouter un Adhérent avec les différents attributs (définis dans le cahier des charges).

Le formulaire sera validé uniquement si chaque champs est rempli et s'il est conforme au format demandé.

Cela est permis grâce aux différents pattern définit pour chaque champs, ainsi que la fonction **required** (qui oblige de remplir le champs)




FORMULAIRE AJOUTER/MODIFIER

[Home](#)

Nom :

Prenom :

Date de Naissance :

Adresse :

Email :

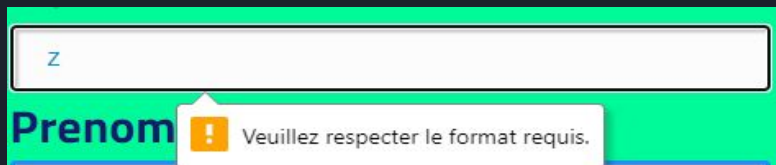
Téléphone :

Catégorie ?

Ajouter **Modifier**

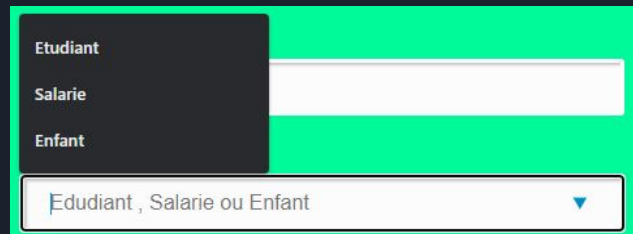
Exemple de vérification de formulaire

Ici on vérifie si le champ est bien rempli au bon format.



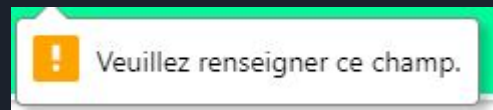
A form field labeled "Prenom" containing the letter "z". Below the field, a white error message box with an orange exclamation mark icon displays the text "Veuillez respecter le format requis."

On vérifie que les catégories renseigné sont ceux définis en amont par le cahier des charges.



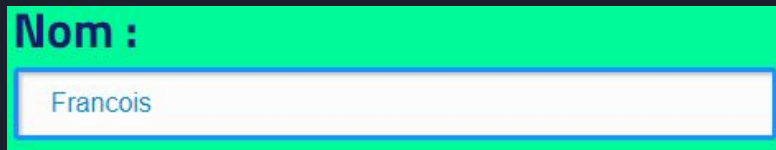
A form field with a dark grey dropdown menu. The menu is open, showing three options: "Etudiant", "Salarie", and "Enfant". Below the menu, a white text box contains the text "Etudiant , Salarie ou Enfant" and a small blue downward arrow icon.

On vérifie que le champ est renseigné



A form field with a white error message box containing an orange exclamation mark icon and the text "Veuillez renseigner ce champ."

Et une fois que le champ est renseigné avec le bon format, l'input est de couleur bleu.



A form field labeled "Nom :" containing the text "Francois". The field has a blue border, indicating a successful validation state.

Modification d'un adhérent



En utilisant ce bouton, il nous permet de récupérer tous les attributs et de les réinsérer dans le même formulaire d'ajout pour modifier directement n'importe quels champs (appart l'id car il est unique)



Grâce à ce lien *Home* il nous permet de revenir au début du logiciel



FORMULAIRE AJOUTER/MODIFIER

[Home](#)

Nom :

Prenom :

Date de Naissance :

Adresse :

Email :

Téléphone :

Catégorie ?


On observe que le formulaire a bien récupéré chaque valeur

Suppression d'un adhérent

1	Peschard	Francois	1986-01-06	22 rue 14 juillet	francois@gmail.com	2020-11-18	0143624433	Salarie		
2	Wick	John	1974-05-03	213 rue de montmartre	JohnWick@gmail.com	2020-12-17	0632456756	Etudiant		
3	Puyol	Carlos	1978-04-13	2 rue de la paix	CarlosPuyol@gmail.com	2018-08-08	0198535357	Etudiant		
31	Ibrahimovic	Zlatan	1970-03-12	Milan A.C	Zlatan@gmail.com	2020-12-20	0123456789	Salarie		



Ce bouton permet de supprimer l'adhérent choisi !

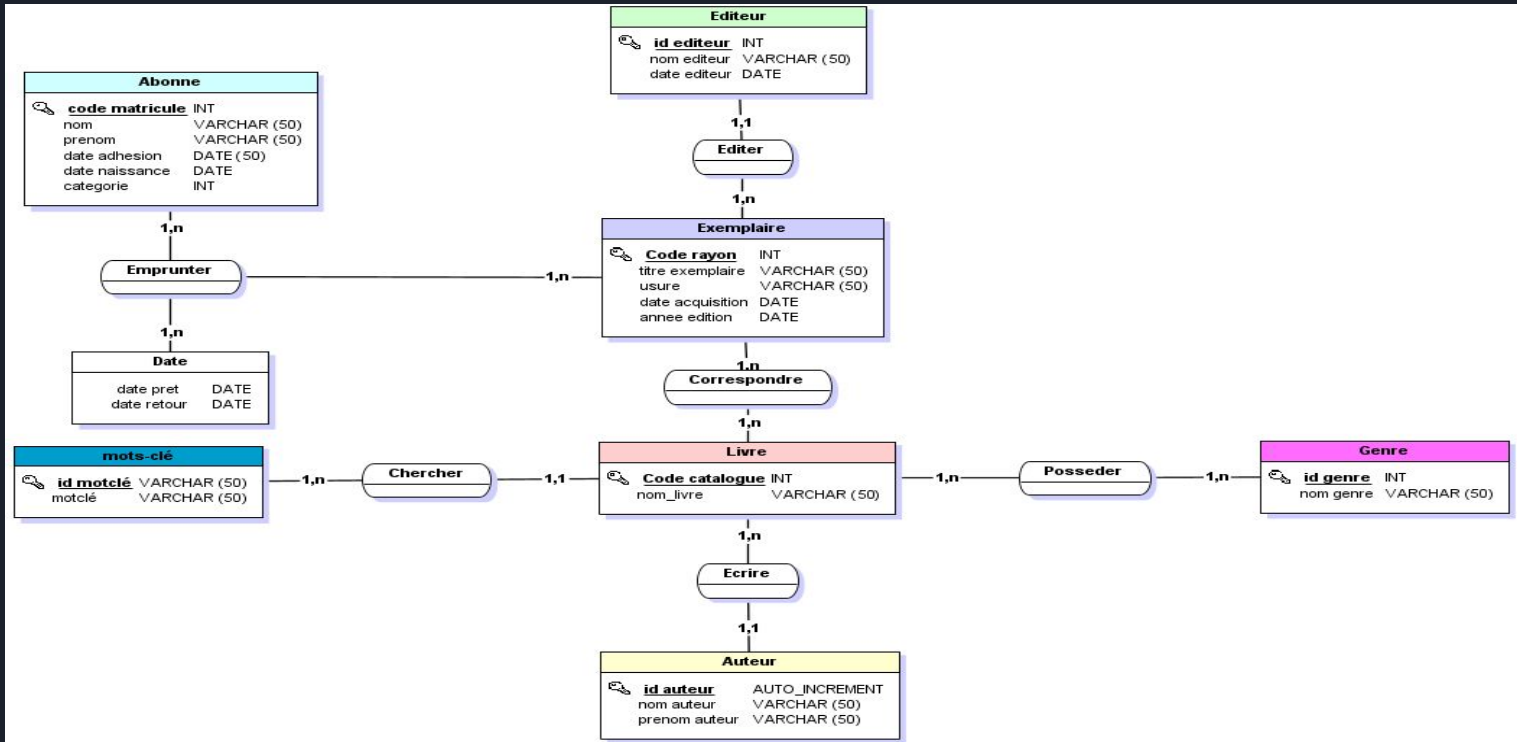
3	Puyol	Carlos	1978-04-13	2 rue de la paix	CarlosPuyol@gmail.com	2018-08-08	0198535357	Etudiant		
---	-------	--------	------------	------------------	-----------------------	------------	------------	----------	---	---

On observe que la dernière ligne 31 a bien été supprimé !



Passons à la partie Technique

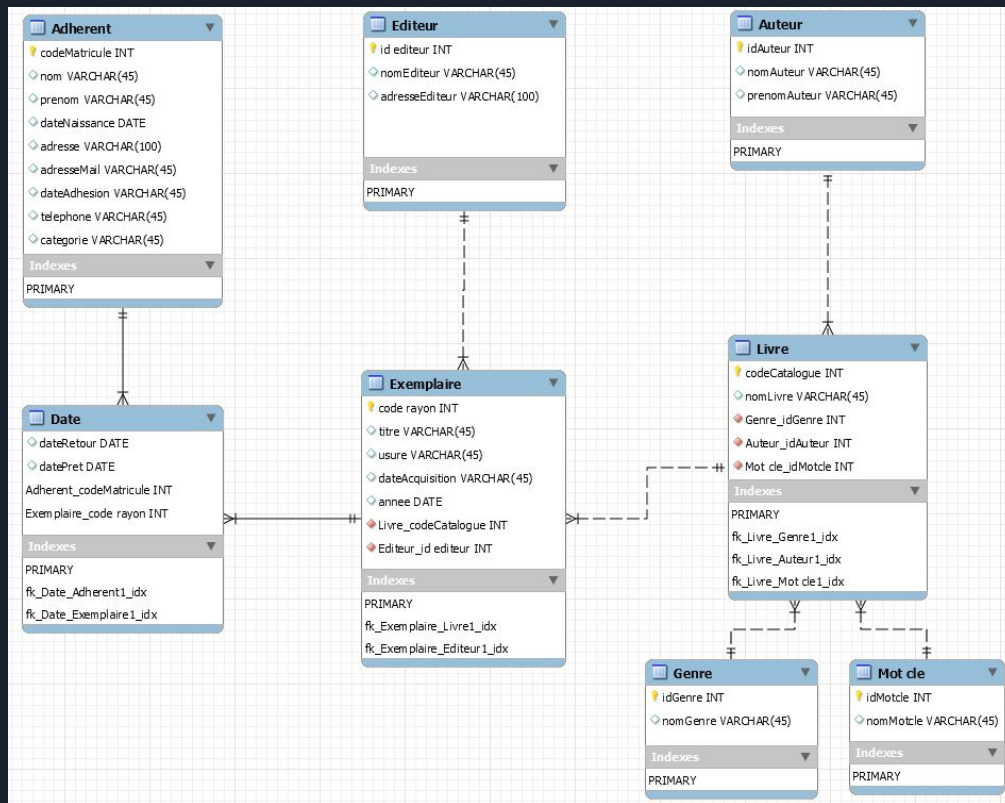
Modèle Conceptuel de Données



Réaliser en fonction du cahier des charges

Modèle Logique de données

Créer grâce à MySQL
Workbench

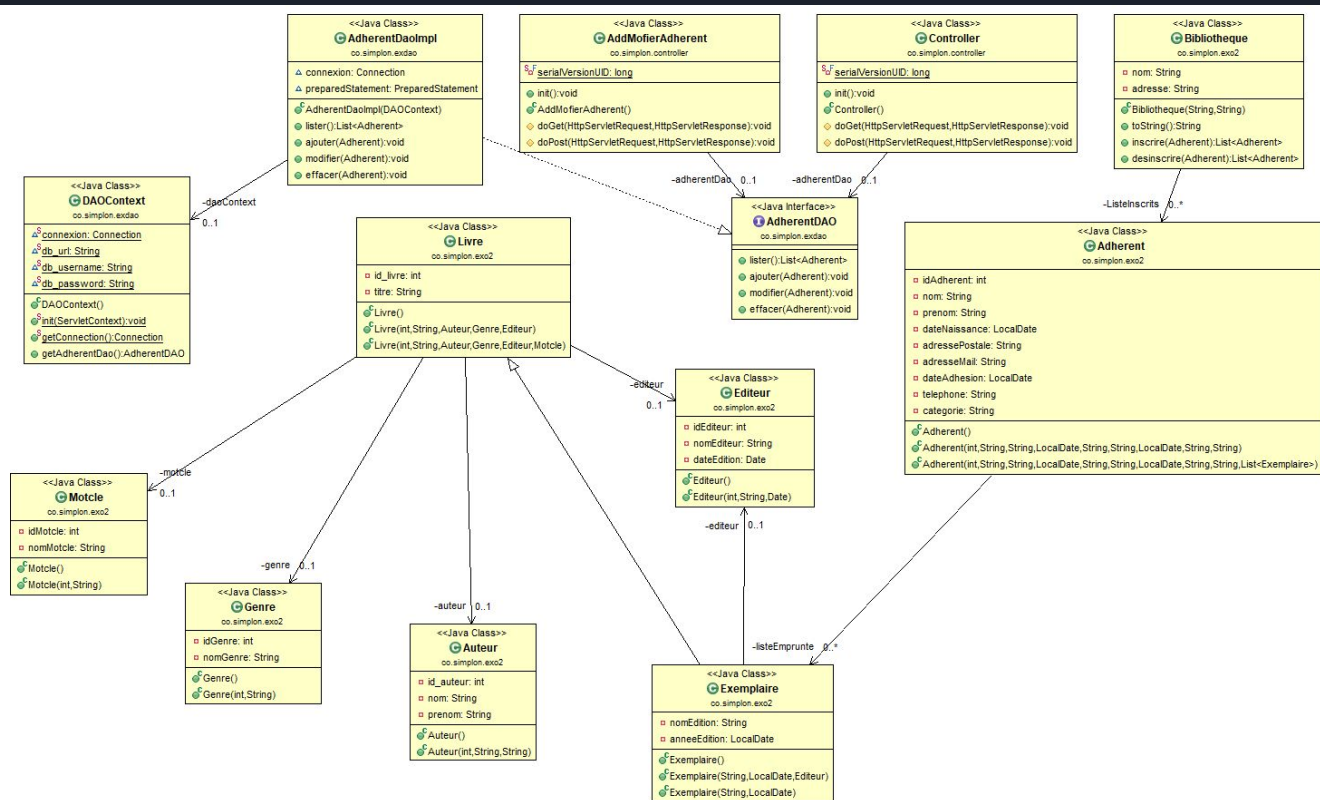


Exemple Table Adhérent

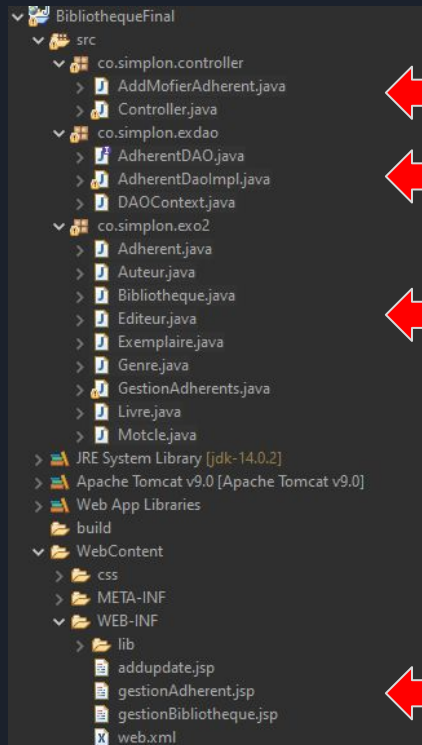
	codeMatricule	nom	prenom	dateNaissance	adresse	adresseMail	dateAdhesion	telephone	categorie
▶	1	Peschard	Francois	1986-01-06	22 rue 14 juillet	francois@gmail.com	2020-11-18	0143624433	Salarie
	2	Wick	John	1974-05-03	213 rue de montmartre	JohnWick@gmail.com	2020-12-17	0632456756	Etudiant
	3	Puyol	Carlos	1978-04-13	2 rue de la paix	CarlosPuyol@gmail.com	2018-08-08	0198535357	Etudiant
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Données insère grâce à un script Sql.

Diagramme de Classe



Utilisation du Modèle Vue Contrôleur



Package qui contient les **Contrôleurs**

Package qui contient connexion à la base de données ainsi que les méthodes pour ajouter lister ect

Package qui contient les **Modèles**

Dossier qui contient les **Vues**

Modèle

Exemple de construction du modèle Adhérents.

Constituer des :

- Attributs
- Constructeurs
- Getters
- Setters

```
public class Adherent {  
  
    private int idAdherent;  
    private String nom;  
    private String prenom;  
    private LocalDate dateNaissance;  
    private String adressePostale;  
    private String adresseMail;  
    private LocalDate dateAdhesion;  
    private String telephone;  
    private String categorie;  
    private List<Exemplaire> listeEmprunte = new ArrayList<Exemplaire>();  
  
    public Adherent() {  
        super();  
    }  
  
    public Adherent(int idAdherent, String nom, String prenom, LocalDate dateNaissance, String adressePostale,  
        String adresseMail, LocalDate dateAdhesion, String telephone, String categorie) {  
        super();  
        this.idAdherent = idAdherent;  
        this.nom = nom;  
        this.prenom = prenom;  
        this.dateNaissance = dateNaissance;  
        this.adressePostale = adressePostale;  
        this.adresseMail = adresseMail;  
        this.dateAdhesion = dateAdhesion;  
        this.telephone = telephone;  
        this.categorie = categorie;  
    }  
}
```

Vue

Exemple de vue pour le
formulaire d'adhérent

```
<form method="post" action="AddMofierAdherent" >
<fieldset>
  <div>
    <input class="enjoy-input" type="hidden" name="idA" id="idA" readonly value="{requestScope.idAdherent}" required placeholder="Id" >
    </div>
    <div>
      <label for="nomA">Nom : </label>
      <input class="enjoy-input" type="text" name="nomA" id="nomA" value="{requestScope.nomtest}" required placeholder="Nom" >
    </div>
    <div>
      <label for="prenomA">Prénom : </label>
      <input class="enjoy-input" type="text" name="prenomA" id="prenomA" value="{requestScope.prenomtest}" required placeholder="Prénom" >
    </div>
    <div>
      <label for="datenA">Date de Naissance : </label>
      <input class="enjoy-input" type="date" name="datenA" id="datenA" value="{requestScope.dateNaissancetest}" required placeholder="format : 2020-11-12" >
    </div>
    <div>
      <label for="adresseA">Adresse : </label>
      <input class="enjoy-input" type="text" name="adresseA" id="adresseA" value="{requestScope.adressePostaletest}" required placeholder="Adresse" >
    </div>
    <div>
      <label for="emailA">Email : </label>
      <input class="enjoy-input" type="email" name="emailA" id="emailA" value="{requestScope.adresseMailtest}" required pattern="[a-zA-Z0-9_&#96;+]+@[a-zA-Z0-9_&#96;+]{1}[a-zA-Z]{2,4}" placeholder="email@address.com" >
    </div>
    <div>
      <label for="telA">Téléphone : </label>
      <input class="enjoy-input" type="tel" name="telA" id="telA" value="{requestScope.teltest}" required placeholder="xxxxxxxxxx" pattern="[d(2)d(2)d(2)d(2)d(2)]" >
    </div>
    <div>
      <label>Catégorie ? </label>
      <input class="enjoy-input" type="text" id="catg" name="catg" value="{requestScope.catgtest}" list="h" required pattern="[Ee]tudiant|[Ss]alarie|[Ee]nfant" placeholder="Edudiant , Salarie ou Enfant">
      <datalist id="h">
        <option>Etudiant</option>
        <option>Salarie</option>
        <option>Enfant</option>
      </datalist>
    </div>
    <div class="divbtn">
      <button class="enjoy" name="add" type="submit" id="valid">Ajouter</button>
      <button class="enjoy" name="mod" type="submit">Modifier</button>
    </div>
  </fieldset>
```


Contrôleur

Méthodes exécutés
en fonction de
l'action (par exemple
d'un bouton)

Ces méthodes sont
définie dans mon
DAO.

```
if (request.getParameter("listeAd") != null) {

request.setAttribute("Ladherents", adherentDao.lister());
this.getRequestDispatcher("/WEB-INF/gestionAdherent.jsp").forward(request, response);

}else if (request.getParameter("add1") != null || request.getParameter("add2") != null) {

    if(noma != "" && noma!= null ) {
        Adherent adherent = new Adherent();
        adherent.setNom(noma);
        adherent.setPrenom(prenomA);
        adherent.setDateNaissance(LocalDate.parse(datenA));
        adherent.setAdressePostale(adresseA);
        adherent.setAdresseMail(emailA);
        adherent.setDateAdhesion(dateDay);
        adherent.setTelephone(telA);
        adherent.setCategorie(catg);
        adherentDao.ajouter(adherent);
    }
request.setAttribute("Ladherents", adherentDao.lister());
this.getRequestDispatcher("/WEB-INF/gestionAdherent.jsp").forward(request, response);
```

DAO

```
public interface AdherentDAO {
```

```
List<Adherent> lister();  
void ajouter(Adherent adherent);  
void modifier(Adherent adherent);  
void effacer(Adherent adherent);
```

Interface qui définit les fonctions que nous avons besoin.

L'implémentation
des différent
fonctions.

Ici nous pouvons
voir la méthode
Ajouter qui a été
définie.

```
public List<Adherent> lister() {}  
  
@Override  
public void ajouter(Adherent adherent) {  
    try {  
  
        connexion = DAOContext.getConnection();  
        preparedStatement = connexion.prepareStatement("INSERT INTO adherent(nom, prenom, dateNaissance, adresse, adresseMail, dateAdhesion, telephone, categorie) VALUES(?, ?, ?, ?, ?, ?, ?, ?);");  
        preparedStatement.setString(1, adherent.getNom());  
        preparedStatement.setString(2, adherent.getPrenom());  
        preparedStatement.setDate(3, Date.valueOf(adherent.getDateNaissance()));  
        preparedStatement.setString(4, adherent.getAdressePostale());  
        preparedStatement.setString(5, adherent.getAdresseMail());  
        preparedStatement.setDate(6, Date.valueOf(adherent.getDateAdhesion()));  
        preparedStatement.setString(7, adherent.getTelephone());  
        preparedStatement.setString(8, adherent.getCategorie());  
  
        preparedStatement.executeUpdate();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
}  
  
public void modifier(Adherent adherent) {}  
  
public void effacer(Adherent adherent) {}
```