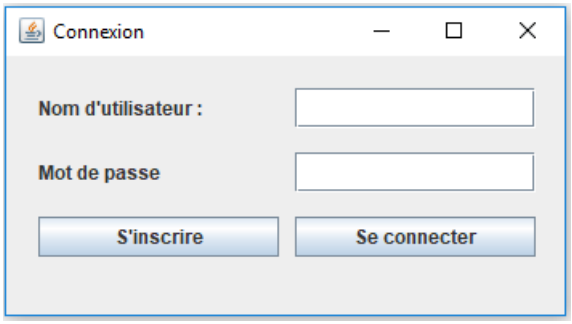
Formation Java - Lille, 28/03/2019 - Achref El Mouelhi

TP 5 : Swing, JDBC et DAO

Exercice 1

Nous voudrons, à travers ce mini-projet, gérer les utilisateurs d'une future application. Nous avons exactement deux types d'utilisateurs : `simple` ou `administrateur`.

- Au lancement de l'application, une fenêtre `Connexion` s'affiche.



- Si l'utilisateur a un compte, alors il peut saisir son nom d'utilisateur (son nom) et son mot de passe (son prénom) pour se connecter. Sinon, il peut cliquer sur `S'inscrire` pour aller sur la fenêtre d'inscription.
- Si l'utilisateur saisit un nom d'utilisateur et/ou un mot de passe qui n'existe(nt) pas dans la base de données, alors un message d'erreur sera affiché. Après la troisième tentative, on ferme la fenêtre `Connexion`.

- On a deux types d'utilisateurs : `simple` ou `administrateur`.
- Un administrateur peut réaliser les opérations suivantes :
  - ▶ créer un utilisateur `simple` ou `administrateur`,
  - ▶ modifier ses données (`nom`, `prénom`, `sexe`),
  - ▶ modifier le compte d'un utilisateur simple,
  - ▶ modifier le type d'un utilisateur de `simple` en `administrateur`,
  - ▶ supprimer son compte, ou
  - ▶ supprimer le compte d'un utilisateur simple
- Un utilisateur simple peut réaliser les opérations suivantes :
  - ▶ modifier ses données (`nom`, `prénom`, `sexe`),
  - ▶ supprimer son compte

- En aucun cas (ajout ou modification), on n'accepte d'insérer des valeurs nulles ou vides dans la base de données.
- Pour s'inscrire, il faut saisir le nom et le prénom et choisir son sexe dans une liste déroulante (`JComboBox`). En validant, un utilisateur simple sera ajouté dans la base de données. La fenêtre `Connexion` sera affichée et la fenêtre `Inscription` sera fermée.
- Pour chaque passage d'une fenêtre à une autre, la fenêtre précédente doit être fermée.
- En cas de connexion réussie, une fenêtre `Accueil` adaptée à chaque type d'utilisateur sera affichée.
- La fenêtre `Accueil` contient des boutons dont chacun correspond à une opération autorisée pour l'utilisateur courant. En cliquant sur le bouton, une fenêtre permettant de réaliser l'opération mentionnée sera affichée.
- Chaque fois qu'on affiche le numéro d'un utilisateur, il est non modifiable.
- Voici le script SQL qui permet de créer la base de données à utiliser dans ce mini-projet

```sql
create database swingJdbc;

use swingJdbc;

create table utilisateur(
numero int primary key auto_increment,
nom varchar(30),
prenom varchar(30),
sexe varchar(1),
type varchar(1)
);

insert into utilisateur (nom,prenom,sexe,type) values
("wick","john","h","a"),
("james","carol","f","s");
```

- `sexe` peut prendre comme valeur soit `f` (pour femme) ou `h` (pour homme)
- `type` peut prendre comme valeur soit `a` (pour administrateur) ou `s` (pour simple)

Étapes :

1. Créer la base de données
2. Créer un projet Maven (`File > New > Other > Maven Project`)
3. Ajouter une dépendance pour le connecteur MySQL
4. Modifier la version de JSE et choisir la 8 (`Properties > Java Build Path`)
5. Créer le bean `Utilisateur` avec un constructeur sans paramètre, un constructeur avec tous les paramètres et un constructeur avec `nom` et `prénom`
6. Créer le DAO correspondant à la classe `Utilisateur` contenant les 5 méthodes habituelles + une méthode `public Utilisateur findByNomAndPrenom(String nom, String prenom)` qui retourne un objet `Utilisateur` correspondant aux nom et prénom passés en paramètre si ces deux derniers existent dans la base de données, `null` sinon.
7. Créer les interfaces graphiques avec Swing et tester les séparément.

8. (Optionnelle) Sauvegarder toutes les connexions dans un fichier `log.txt`. Chaque ligne de ce fichier contient les `nom` et `prénom` de l'utilisateur connecté + la date de connexion.

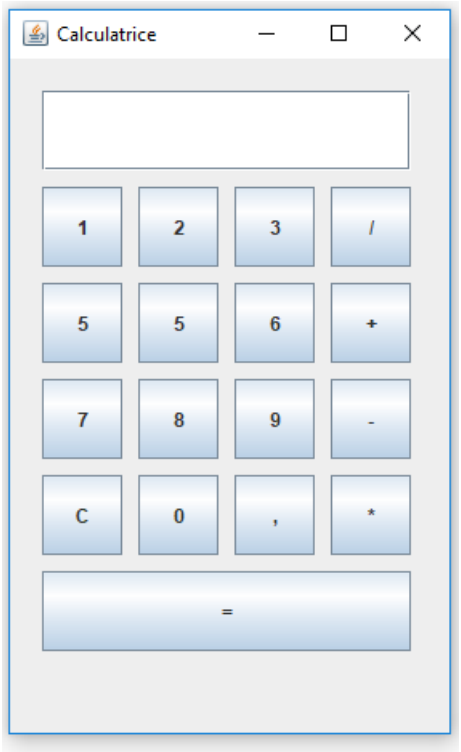
```
LocalDateTime localdate = LocalDateTime.of(2019, 3, 28, 10, 30,30);
DateTimeFormatter dateTimeFormatter = DateTimeFormatter.ofPattern("dd-MM-yyyy hh:mm:ss");

System.out.println(localdate.format(dateTimeFormatter));
// affiche la date formattée 28-03-2019 10:30:30
```

Hypothèse

- On suppose que les nom et prénom sont uniques

Exercice 2

Cet exercice consiste à réaliser une calculatrice comme le montre la figure ci-dessous.



- Cette calculatrice doit permettre d'effectuer des opérations arithmétiques sur des nombres réels.
- Le bouton `C` sera utilisé pour effacer tout le contenu de la zone texte
- Le bouton `sup` sera utilisé pour effacer le dernier caractère ajouté dans la zone texte
- Pour mieux contrôler la saisie, il faut seulement autoriser la saisie de valeurs dans la zone de texte via les touches de la calculatrice.
- Le code permettant de créer la fenêtre est donnée dans la page suivante
- On ne peut pas utiliser le bouton virgule s'il a déjà été utilisé pour la saisie en cours

```
public class Calculette extends JFrame{
  JTextField text = new JTextField();
  JButton but1 = new JButton("1");
  JButton but2 = new JButton("2");
  JButton but3 = new JButton("3");
  JButton but4 = new JButton("5");
  JButton but5 = new JButton("5");
  JButton but6 = new JButton("6");
  JButton but7 = new JButton("7");
  JButton but8 = new JButton("8");
  JButton but9 = new JButton("9");
  JButton but0 = new JButton("0");
  JButton plus = new JButton("+");
  JButton moins = new JButton("-");
  JButton fois = new JButton("*");
  JButton div = new JButton("/");
  JButton point = new JButton(",");
  JButton clean = new JButton("C");
  JButton equal = new JButton("=");

  public void afficherCalculette() {
    text.setBounds(20, 20, 230, 50);
    but1.setBounds(20, 80, 50, 50);
    but2.setBounds(80, 80, 50, 50);
    but3.setBounds(140, 80, 50, 50);
    div.setBounds(200, 80, 50, 50);
    but4.setBounds(20, 140, 50, 50);
    but5.setBounds(80, 140, 50, 50);
    but6.setBounds(140, 140, 50, 50);
    plus.setBounds(200, 140, 50, 50);
    but7.setBounds(20, 200, 50, 50);
    but8.setBounds(80, 200, 50, 50);
    but9.setBounds(140, 200, 50, 50);
    moins.setBounds(200, 200, 50, 50);
    clean.setBounds(20, 260, 50, 50);
    but0.setBounds(80, 260, 50, 50);
    point.setBounds(140, 260, 50, 50);
    fois.setBounds(200,260, 50, 50);
    equal.setBounds(20, 320, 230, 50);
    this.add(but1);
    this.add(but2);
    this.add(but3);
    this.add(but4);
    this.add(but5);
    this.add(but6);
    this.add(but7);
    this.add(but8);
    this.add(but9);
    this.add(but0);
    this.add(clean);
    this.add(plus);
    this.add(moins);
    this.add(fois);
    this.add(div);
    this.add(text);
    this.add(point);
    this.add(equal);
    this.setSize(290, 460);
    this.setTitle("Calculatrice");
    this.setLayout(null);
    this.setVisible(true);
  }
}
```