

## TP : Java avancé

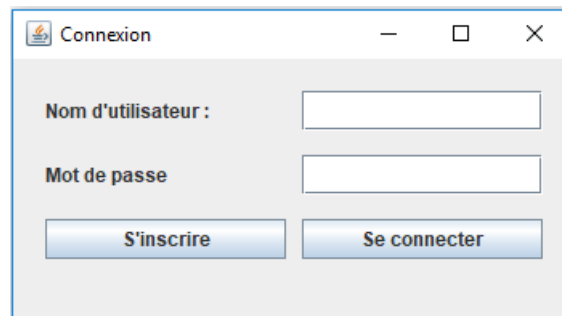
---

### Exercice 1

Nous voudrions, à travers ce mini-projet, gérer les élèves d'une école. Notre application doit nous permettre, après une première phase d'authentification/inscription, d'ajouter/éditer ou chercher un élève dans un fichier texte `eleve.txt`. Je vous conseille fortement de bien lire la totalité de l'énoncé avant de commencer à coder.

### Authentification

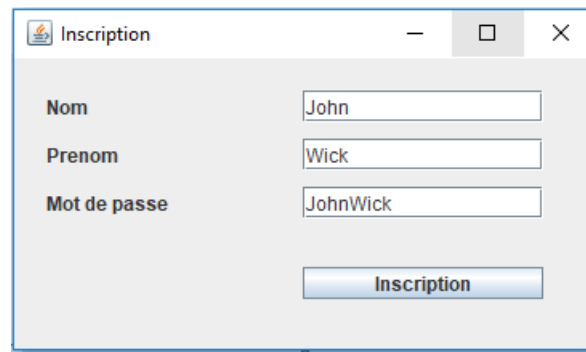
Pour accéder aux différentes fonctionnalités de notre application, il faut s'authentifier. Un utilisateur doit donc saisir son nom et son mot de passe qui existent dans un fichier `utilisateur.txt` (on verra plus tard sa structure), sinon il doit s'inscrire. Si la saisie est correcte, la fenêtre **Gestion élève** sera



affichée. Sinon, il a le droit à deux saisies erronées (un message d'erreur est affiché lors de chaque mauvaise saisie). La fenêtre d'authentification doit se fermer à la troisième saisie incorrecte.

### Inscription

Pour s'inscrire, l'utilisateur doit saisir son nom, son prénom et son futur mot de passe. En cliquant sur le bouton **Inscription** (initialement désactivé, activé après la saisie des trois champs), une ligne sera ajoutée dans le fichier `utilisateur.txt` et la fenêtre **Authentification** sera affichée de nouveau (l'utilisateur pourra donc utiliser les données qu'il vient de saisir). La ligne, qui sera ajoutée dans le fichier `utilisateur.txt`, devra contenir les champs suivants (séparés par le caractère espace) : nom prénom mot-de-passe. Attention : pour simplifier, nous ne traitons ni les noms ni les prénoms



composés.

## Gestion élève

Notre fenêtre **Gestion élève** comporte deux boutons **Ajouter** et **Chercher**. Un clic sur le premier nous mène vers la fenêtre `ajouterEleve` et un clic sur le second vers la fenêtre `chercherEleve`.

La fenêtre `ajouterEleve` doit nous permettre la saisie de trois champs texte `num`, `nom` et `prenom` ainsi que deux boutons radios (`JButtonRadio`) pour choisir le `sexe` de l'élève (par défaut `féminin`). Dans cette fenêtre, il faut aussi trois boutons **Ajouter** (initialement désactivé), **Effacer** (initialement désactivé aussi) et **Quitter**.

Après avoir saisi les trois champs `num`, `nom` et `prenom`, le bouton **Ajouter** sera activé. Pour ajouter un élève, il nous faut donc les 4 informations (numéro unique, nom, prénom et sexe masculin ou féminin). En cliquant sur le bouton **Ajouter**, nous devons vérifier si le numéro saisi existe déjà dans notre fichier `eleve.txt`. Dans l'affirmative, nous devons afficher un message d'erreur. Sinon une nouvelle ligne sera ajoutée dans le fichier `eleve.txt` et un message d'information sera affiché. Dans les deux cas, notre formulaire sera vidé (les trois zones textes `numero`, `nom` et `prenom`).

Le bouton **Effacer** servira à vider tous les champs de formulaire tandis que **Quitter** permettra de fermer la fenêtre de l'application.

La fenêtre `chercherEleve` permettra de chercher un élève selon son numéro. Si le numéro existe dans notre fichier `eleve.txt`, alors il faut ré-afficher le numéro de l'élève correspondant, son nom, son prénom et son sexe dans une nouvelle fenêtre `editerEleve` (cette fenêtre a deux boutons **Modifier** et **Supprimer**). Sinon (si le numéro n'existe pas), un message nous informant de l'inexistence de cet élève sera affiché. Évidemment, nous ne pourrons modifier le `num` d'un élève (vous pourrez utiliser la méthode `setEditable(false)`).

En cliquant sur le bouton **Supprimer**, l'élève correspondant sera supprimé de notre fichier `eleve.txt`

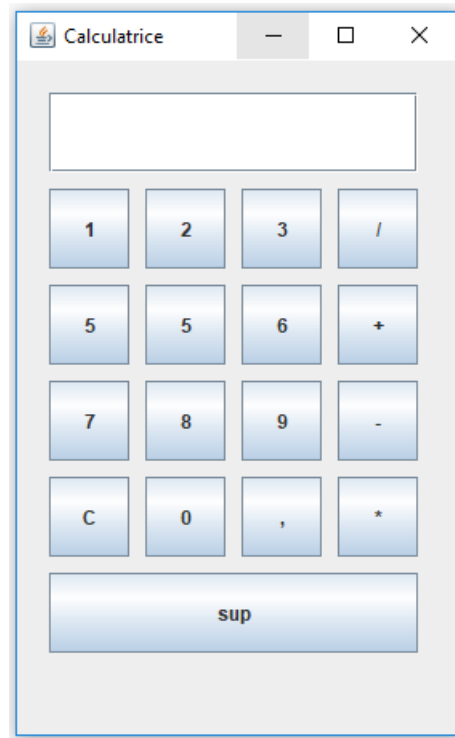
## Consignes

- **Eleve** et **Utilisateur** doivent hériter d'une classe **Personne**
- Pour chacune des deux classes **Eleve** et **Utilisateur**, il faut associer une deuxième classe qui héritera de **JFrame** et qui contiendra les méthodes qui permettront d'afficher les différents formulaires qui seront utilisées pour les fonctionnalités précises dans l'énoncé.
- Les classes **Eleve** et **Utilisateur** doivent contenir seulement les attributs et les méthodes de bases (aucun composant graphique **swing**).
- Les classes Fenêtres de **Eleve** et **Utilisateur** doivent utiliser les classes **Eleve** et **Utilisateur** en appelant les méthodes qui permettent de construire des objets, chercher, modifier supprimer ou ajouter dans un fichier.
- Chaque fois qu'on ouvre une nouvelle fenêtre à partir d'une autre, il faut penser à fermer la fenêtre appelante (pour éviter d'avoir plusieurs fenêtres ouvertes au même temps)
- Pour utiliser les boutons radios, il faut penser à déclarer un groupe de boutons et de lui affecter les boutons radios comme-indiqué dans l'exemple ci-dessous :

```
ButtonGroup group = new ButtonGroup();
JButtonRadio m13 = new JButtonRadio("Marseille");
JButtonRadio l69 = new JButtonRadio("Lyon");
group.add(m13);
group.add(l69);
```
- Pour le champ `num`, il doit contenir exactement trois **chiffres**. Par contre, les champs `nom` et `prenom` ne doivent contenir que des lettres. Pour cela, vous pouvez écrire deux méthodes **static** pour que vous puissiez les appeler pour **Eleve** et **Utilisateur**.
- N'oublions pas d'ajouter chaque fois les commentaires Javadoc et de générer à la fin la documentation de votre mini-projet.

## 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

```

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JTextField;

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 sup = new JButton("sup");

    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);
        sup.setBounds(20, 320, 230, 50);
        this.setLayout(null);
        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(sup);
        this.setVisible(true);
        this.setSize(290, 460);
        this.setTitle("Calculatrice");
        this.setLocationRelativeTo(null);
    }
}

```