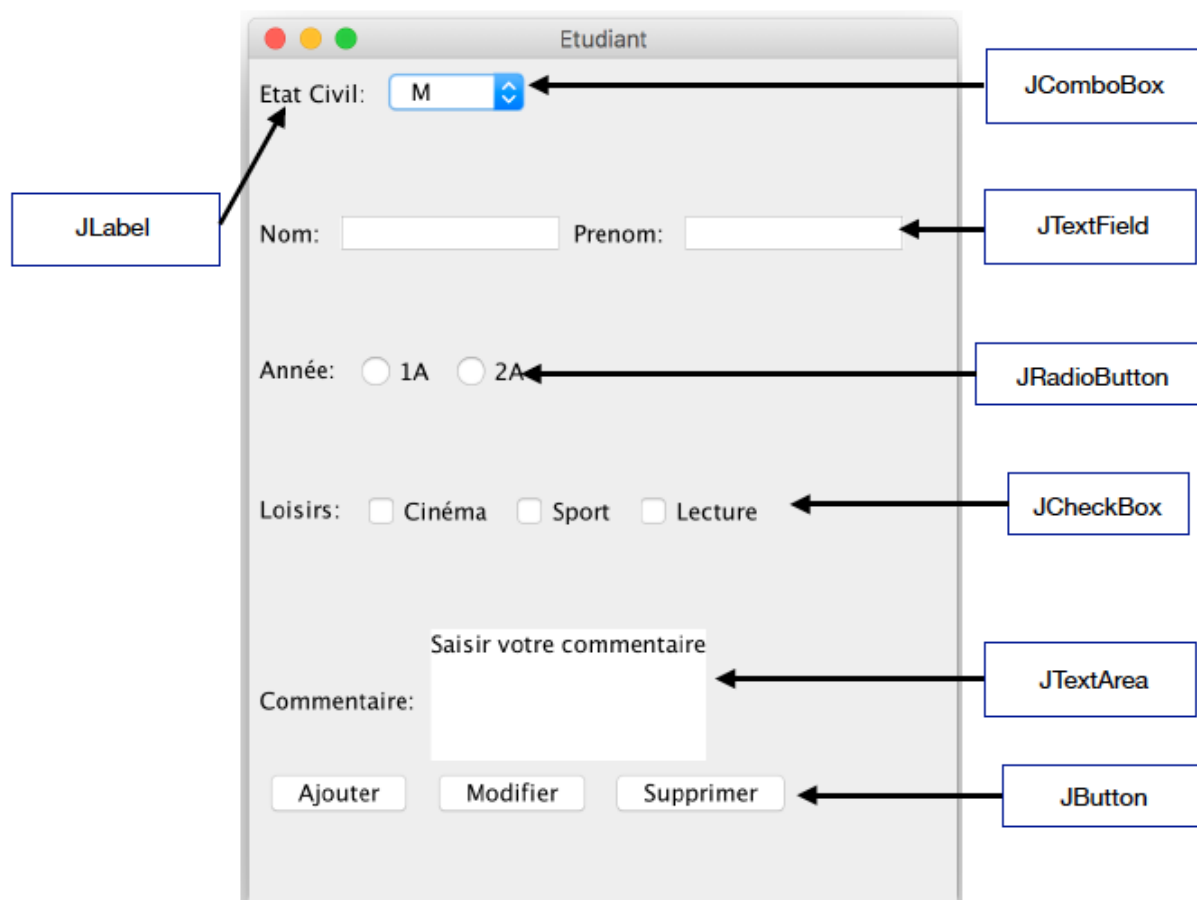


## Java Avancé – TP1 Fenêtre Etudiant

*Durée : 1 séance, pgm + CR à envoyer pour début de séance 2*

L'objectif du TP est de **programmer une interface H/M à la main**, sans l'aide de l'IDE.  
Les TP suivant utiliseront l'assistant WYSIWYG pour développer des interfaces plus poussées.  
ICI ON NE GÈRERA PAS LES CLIC SUR LES BOUTONS.



Vous allez développer une interface composée d'une fenêtre (JFrame) avec :

- 6 JLabel
- 1 JComboBox
- 2 JTextField
- 2 JRadioButton, exclusif
- 3 JCheckBox, multiple
- 1 JTextArea
- 3 JButton

1. Commencer définir un **nouveau projet** (Java application), et spécifier dans le code créé que la classe hérite de JFrame.

Que dit l'IDE ? Classe non définie ? Il faut importer les classes inconnues.  
*Observez dans quel package JAVA se trouve l'API Swing.*

2. **Déclarer** tous les composants nécessaires, sans les instancier pour l'instant (*les composants sont mentionnés sur la fenêtre proposée p.1*).
3. Créer une **méthode privée** `initComponents()` qui va instancier tous ces composants, définir le comportement de la fenêtre à sa fermeture. Regarder la Javadoc pour voir comment se définit la `ComboBox`, la `JTextArea`. *Expliquez.*
4. Créer le **constructeur** qui simplement appelle votre méthode `initComponents()`.
5. Créer le **main()**, instancier une fenêtre, définir sa taille optimale (méthode `pack()`) et la rendre visible. Faites tourner. *Que constatez-vous ?*
6. Récupérer le `contentPane` de la fenêtre principale.
7. Définir le `layout` au niveau du conteneur « implicite » de la fenêtre. Dans cet exemple, on a choisi d'utiliser un `GridLayout` avec 6 lignes et 1 colonne, dans lequel on va ajouter les 6 panneaux (un par ligne).
8. Définir ensuite chacun des panneaux, qui sont des `JPanel`, en ajoutant les composants dans le bon ordre. Par exemple pour la 1<sup>ère</sup> ligne :

```
pann1= new JPanel();
pann1.add(lbl_etatCivil); // d'où l'intérêt de bien nommer ses variables !
pann1.add(cbb_etatCivil);
```

9. Ajouter ensuite les 6 panneaux au conteneur de la fenêtre, et testez à nouveau votre code. *Quelle différence voyez-vous avec la fenêtre proposée en page 1 ?*
10. On veut maintenant utiliser un `FlowLayout` avec cadrage à gauche. Vous pouvez utiliser le même `layout` comme gestionnaire de placement pour *chacun* des panneaux :

```
LayoutManager monLayout=new FlowLayout(FlowLayout.LEFT);
```

Ce qui nous permet de définir le premier panneau (`pann1`) en utilisant notre `FlowLayout` cadré à gauche :

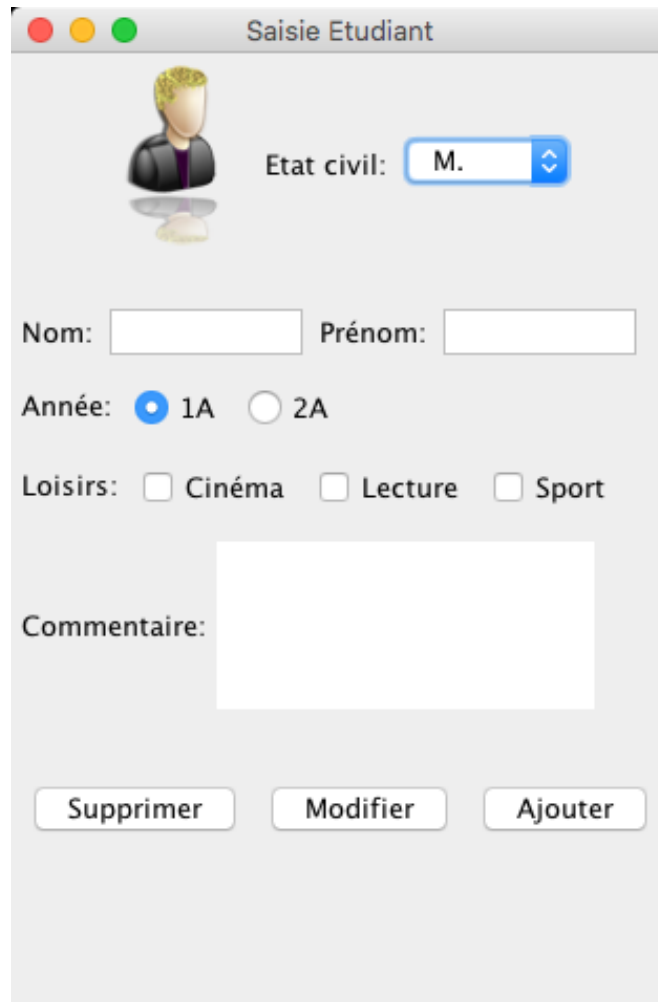
```
pann1 = new JPanel(monLayout);
```

Modifiez votre code en utilisant ce *layout* là où c'est nécessaire pour obtenir la fenêtre proposée.

## FACULTATIF, POUR CEUX QUI ONT FINI

- 1- **Pré cocher** une réponse par défaut pour l'année (1A)
- 2- Faire qu'on puisse **modifier le titre de la fenêtre** à son lancement. *Indice : nouveau constructeur.*

- 3- **Ajouter une image** en haut devant l'état civil. Utiliser un JLabel avec un objet ImageIcon.
- 4- **Modifier les layouts** pour que les lignes du milieu soient plus serrées, comme sur la fenêtre proposée ici :



The image shows a Java Swing window titled "Saisie Etudiant". The window has a standard Mac OS-style title bar with red, yellow, and green window control buttons. The main content area is light gray and contains the following elements from top to bottom:

- A small icon of a person's head and shoulders.
- A label "Etat civil:" followed by a dropdown menu showing "M." with a blue arrow.
- Two text input fields: "Nom:" and "Prénom:".
- A label "Année:" followed by two radio buttons: "1A" (which is selected) and "2A".
- A label "Loisirs:" followed by three checkboxes: "Cinéma", "Lecture", and "Sport", all of which are currently unchecked.
- A text area labeled "Commentaire:".
- Three buttons at the bottom: "Supprimer", "Modifier", and "Ajouter".

## TESTS UNITAIRES

Normalement, chaque méthode et chaque constructeur doit être testée pour tous les cas qui peuvent se présenter. Il faut donc :

- déterminer toutes les *méthodes et constructeurs* à tester
- pour chacun, déterminer les *différents cas à tester*,
- pour chaque cas à tester, précisez les *valeurs de départ* utilisées pour faire le test, les *valeurs de retour* attendues, construire un *test unitaire* pour chaque cas et vérifier que ça passe.

---

## Travail à rendre

Envoyer votre CR de TP (réponses aux 4 questions) + code source par email

à [veronique.deslandres@univ-lyon1.fr](mailto:veronique.deslandres@univ-lyon1.fr)

avec en objet [ASPE] TP1 : votre TP doit être placé dans un répertoire **ayant votre nom**, vous le compressez, et c'est cette archive que vous transmettez.

Remarque : commentez *pertinemment* les différentes parties de votre code.