

## Java Avancé – TP2 Gestion des événements

*Durée : 1 séance, pgm + CR à envoyer pour début de séance 3*

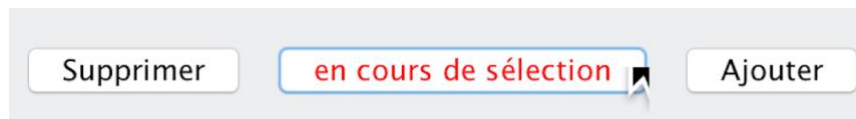
L'objectif du TP est de **programmer la gestion des événements liés à l'interface Etudiant réalise au TP1**, toujours sans l'aide de l'IDE. Nous allons mettre en place les 3 façons de gérer les écouteurs, vues en cours.

Vous allez ajouter les événements suivants à votre interface :

1. Ajouter des événements de types `WindowEvent` pour l'ouverture et la fermeture de la fenêtre. Ici on utilise l'écouteur `WindowListener` en **définissant la classe principale comme étant son propre écouteur**. Vous allez redéfinir toutes les méthodes de l'interface pour fournir un message approprié, par exemple :
  - `windowClosing()` affichera le message « Application Fermée » au moment de la fermeture de la fenêtre, dans la fenêtre *output*.
  - `windowOpened()` ajoutera « de Lyon » au titre de votre fenêtre au moment de l'ouverture de la fenêtre avec la méthode `setTitle()` de `JFrame`. Pour récupérer le titre initial, on utilisera `getTitle()` ;
2. Aurait-on pu utiliser `WindowAdapter` ici ? Quel est le *type d'événement* que prennent toutes ces méthodes en argument ? Comment sont gérés les cas considérés (ouverture, iconification, etc.) ?
3. Faire en sorte que l'image utilisée change selon l'état civil sélectionné. Pour cela, il faut trouver une nouvelle image. Pour l'écouteur d'événement, la `comboBox` fonctionne avec un écouteur d'item. On définira cette fois l'écouteur `ItemListener` **en classe anonyme**. *Quelle unique méthode faut-il donc définir ? Quel événement prend-elle en argument ? Notez la différence avec `WindowListener`.*

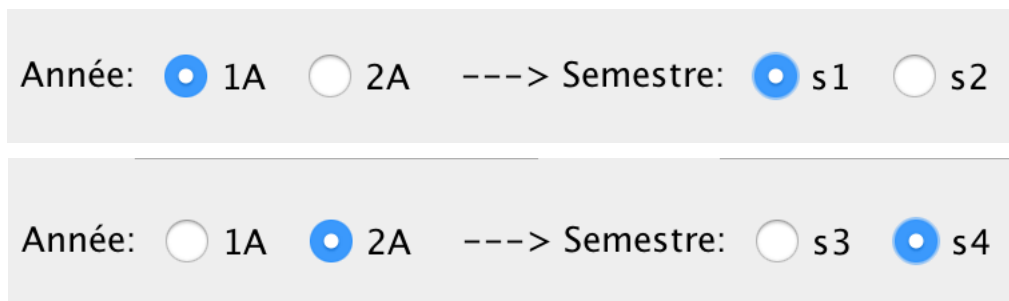
On utilisera la méthode `getItem()` de l'argument pour connaître l'état civil sélectionné.
4. Écrire l'écouteur `MouseListener` **en classe interne** pour changer le texte ainsi que la couleur du texte des boutons Supprimer et Modifier, quand la souris passe sur le bouton. Le texte du bouton doit être « Bouton en train d'être cliqué » et de couleur rouge. Pour changer la couleur du texte, vous pouvez utiliser la méthode `setForeground(Color.red)`. Dans cette question, modifier les deux méthodes suivantes :
  - `mouseEntered()` pour changer le texte du bouton
  - `mouseExited()` pour rendre le bouton à son état initial

Les autres méthodes de `MouseListener` afficheront simplement ce qui se passe pour le bouton dans la fenêtre *output* « bouton : cliqué, pressé, relâché ».



4- Utiliser l'écouteur `ActionListener` en 2<sup>ème</sup> classe interne pour :

- a- Ajouter un lien de dépendance **entre l'année et le semestre** de chaque étudiant. Pour cela, il faut ajouter une ligne dans le formulaire qui contient : un label pour « semestre » et 4 boutons radios qui sont (S1, S2, S3 et S4). Faire en sorte qu'initialement les boutons des semestres soient cachés. Si l'utilisateur sélectionne le bouton radio 1A, les 2 boutons S1 et S2 seront affichés. S'il sélectionne le bouton 2A, les 2 boutons S3 et S4 seront affichés comme le montre la figure suivante :



Année:  1A  2A ---> Semestre:  s1  s2

Année:  1A  2A ---> Semestre:  s3  s4

- b- **Réinitialiser tous les champs de la fenêtre** quand on clique sur le bouton supprimer. Il faut que tous les champs texte soient vides, l'état civil soit « M. » avec l'icône image associé, il faut que tous les boutons soient décochés, les boutons des semestres cachés, et la zone commentaire remise à vide.

- c- **Afficher les informations de l'étudiant** quand on clique sur le bouton « Ajouter » sous la forme suivante :

« M/Mme *prénom nom* est inscrit(e) en 1<sup>ère</sup>/2<sup>ème</sup> année en semestre S1/S2/S3/S4 et a comme loisirs *Cinéma/Sport/Lecture*. Autres informations : (*texte de la zone commentaire*) ». L'affichage reprend les informations saisies et l'information des boutons sélectionnés.

- Dans un premier temps, faites l'affichage dans la console avec la méthode `System.out.println()`.
- Dans un deuxième temps, faites-en sorte que toutes ces informations soient affichées dans une fenêtre `MessageDialog`, qui s'ouvre quand on clique sur le bouton « Ajouter ». Voici la syntaxe (vue au cours 3) :

```
JOptionPane.showMessageDialog(la-fenetre-courante, la-chaine-a-afficher, "titre",  
JOptionPane.INFORMATION_MESSAGE);
```

## FACULTATIF, POUR CEUX QUI ONT FINI

5. Avant d'ajouter le texte récapitulatif de la Q4.c, **vérifier que les champs** sont tous remplis et ne sont pas nuls.
  6. Ajouter du code pour le **bouton modifier** : quand on clique dessus, une boîte de dialogue (*MessageDialog*) s'ouvre mentionnant qu'il faut modifier directement les champs de la fenêtre.
  7. Ajouter du code à la **fermeture de la fenêtre** : afficher une boîte de dialogue pour confirmer qu'on veut réellement quitter (*ConfirmDialog*)
  8. Ajouter du code pour que **l'effet sur les boutons soit facultatif**, par exemple choisi au moment de la création de la fenêtre (nouveau constructeur).
  9. Ajouter du code au **bouton supprimer** : afficher une boîte de dialogue pour confirmer qu'on veut réellement supprimer (*ConfirmDialog*)
- 

## TRAVAIL A RENDRE

Envoyer votre CR de TP + code source **par email** à [veronique.deslandres@univ-lyon1.fr](mailto:veronique.deslandres@univ-lyon1.fr)

avec en objet [ASPE] TP2 : 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.