Compte rendu TP2 Java

**Elève :** Jérémy LAURENT

**Question1 :** Ajouter des évènements de types WindowEvent pour l’ouverture et la fermeture de la fenêtre. Ici on utiliser l’écouteur WindowListener en définissant la classe principale comme étant son propre écouteur.

**Réponses :**



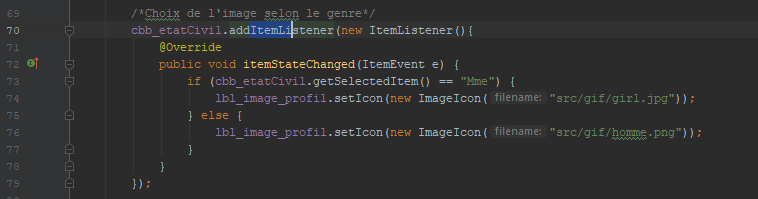
**Question 2 :** Aurait-on pu utiliser WindowAdapter ici ? Quel est le type d’événement que prennent toutes ces méthodes en argument ? Comment sont gérés les cas considérés (ouverture, iconification, etc.) ?

**Réponses :**

* Oui on peut utliser WindowAdapter.
* Les méthodes prendront en compte des évenements de type fenêtre (WindowEvent).

**Question 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.

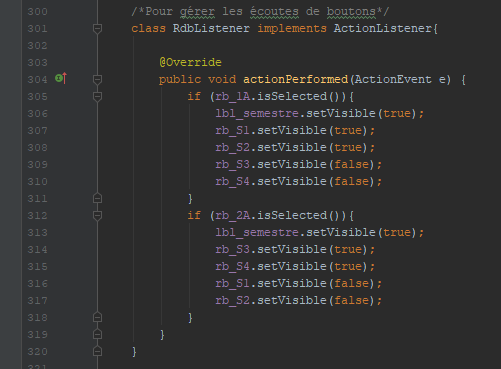
**Réponses :**

* A tester directement avec le programme JAVA FenEtu

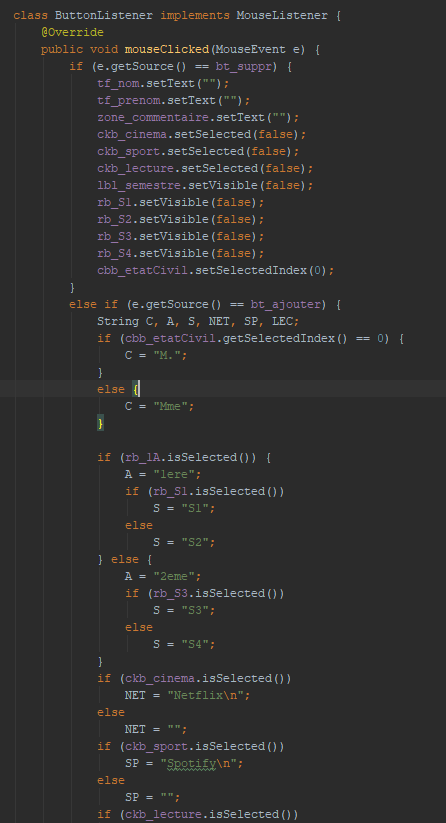
Redéfinition de la méthode ItemStateChanged  🡪 change l’icone en fonction de la ComboBOX et à pour paramètre : itemEven

**Question 4.bis :** Utiliser l’écouteur ActionListener en 2 ème classe interne pour :

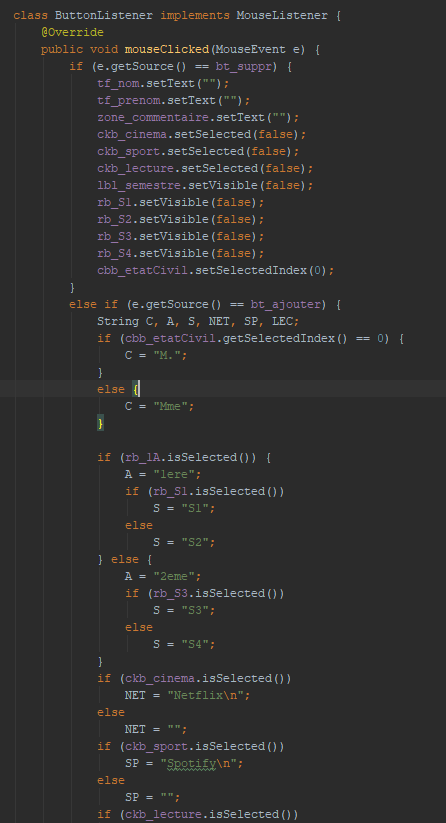
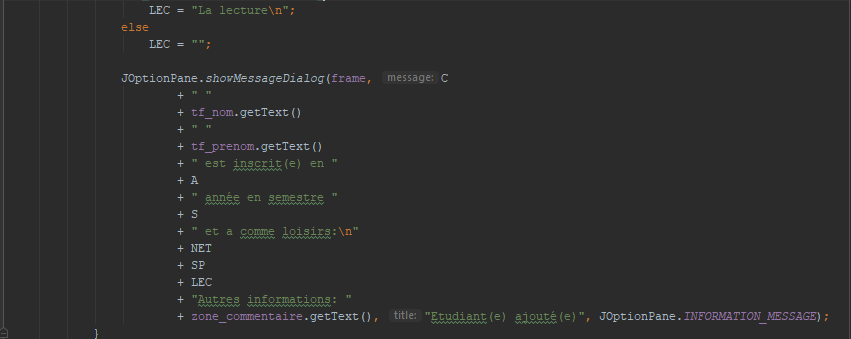
**a-** Ajouter un lien de dépendance entre l’année et le semestre de chaque étudiant. Pour cela, il faut ajouter une ligne dans le formulaire qui contient : un label pour « semestre » et 4 boutons radios qui sont (S1, S2, S3 et S4). Faire en sorte qu’initialement les boutons des semestres soient cachés. 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 :



**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.

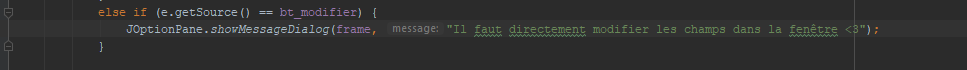


**c-** Afficher les informations de l’étudiant quand on clique sur le bouton « Ajouter » sous la forme suivante : « M/Mme prénom nom est inscrit(e) en 1ère/2ème année en semestre S1/S2/S3/S4 et a comme loisirs Cinéma/Sport/Lecture. Autres informations : (texte de la zone commentaire) ». L’affichage reprend 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);



**Question 5 :** Avant d’ajouter le texte récapitulatif de la Q4.c, vérifier que les champs sont tous remplis et ne sont pas nuls

**Question 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



**Question 7:** Ajouter du code à la fermeture de la fenêtre : afficher une boite de dialogue pour confirmer qu’on veut réellement quitter (ConfirmDialog)

