

Programmation IHM en Java TP1 - Fenêtre Clubs Etudiants

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.

DANS CE 1^{er} TP, ON NE GÈRERA PAS LES EVENEMENTS (ex. CLICs SUR LES BOUTONS).

On imagine créer une interface pour la saisie, en début d'année, des préférences en termes de loisirs des étudiants, afin de constituer des clubs qui pourraient fonctionner toute l'année au département.



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

- 7 JLabel
- 5 JComboBox
- 2 JTextField
- 2 JRadioButton, exclusif
- 4 JCheckBox, à choix multiple
- 1 JTextArea
- 3 JButton

1. Commencer par 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** (clic droit : `import xxx`). *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*). Utiliser des noms suffisamment parlant.
3. Créer le **constructeur** qui appelle la méthode `initComponents()` et définit le comportement de la fenêtre à sa fermeture.
4. Créer le **main()**, instancier une fenêtre, définir sa taille optimale (méthode `pack()`) et la rendre visible.
5. Ecrire la **méthode privée initComponents()** qui va instancier tous ces composants. Pour l'image, créer un label à partir d'une instance d'`ImageIcon` comme expliqué en cours. Regarder la **Javadoc** pour voir comment se définit la `ComboBox`, la `JTextArea`. *Expliquez dans le CR.*
6. Dans `initComponents()`, 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 5 lignes et 1 colonne, dans lequel on va ajouter 5 panneaux. Ici chaque ligne est un panneau composé : un pour l'image et la `comboBox` de l'état civil ; un pour Nom, Année, Préférences ; un pour les loisirs et leur niveau, etc. le dernier étant composé des 3 boutons du bas. Vous pouvez aussi procéder différemment : considérer 8 lignes par exemple, le rendu sera différent : à vous de choisir selon ce que vous souhaitez.
8. Définir ensuite chacun des panneaux, qui sont des `JPanel`, en ajoutant les composants dans le bon ordre. Par exemple pour la 1^{ère} ligne :

```
panEtatCivil = new JPanel();
panEtatCivil.add(lbl_etatCivil);
panEtatCivil.add(cbb_etatCivil);
```

Rappel : par défaut, un `JPanel` est en `FlowLayout` centré. Le rendu devrait être dans un premier temps être différent de l'interface proposée : c'est normal.

9. Ajouter ensuite les panneaux au conteneur de la fenêtre, et testez votre code. *Voyez-vous des différences avec la fenêtre proposée en page1 ?*

10. On veut maintenant utiliser un `FlowLayout` avec cadrage à gauche. Vous pouvez utiliser le même `layout` comme gestionnaire de placement pour les panneaux concernés :

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

Ce qui nous permet de définir le panneau pour l'année, les préférences et les niveaux ainsi que le `pannCommentaire` en utilisant notre `FlowLayout` cadré à gauche :

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

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

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- Permettre de saisir **un autre club** que ceux prédéfinis (ajout d'un `textField`):

A screenshot of a Java Swing window. At the top, there is a checkbox labeled "Autre". Below it is an empty text field. At the bottom, there is a dropdown menu with the text "débutant" and a blue arrow icon pointing downwards.

Remarque : **commentez** *pertinemment* les différentes parties de votre code, au fur et à mesure du codage pour ne pas perdre de temps.

Utiliser la **forge de Lyon1** (GITLab) pour stocker votre code source et éviter toute perte de code.

Travail à rendre

Déposer votre CR de TP (réponses aux 4 questions) + code source sur Tomuss dans les colonnes mentionnées (un dépôt par colonne).

En cas de problème écrire à veronique.deslandres@univ-lyon1.fr

Remarques sur Tomuss :

- Vous pouvez récupérer ce que vous avez déjà déposé.
- On ne peut pas effacer de fichier
- Vous pouvez redéposer un fichier, cela remplace l'ancien.