

Série TP 9

Les boîtes de dialogue. JOptionPane et JFileChooser.

Etapes à suivre

I. Top-Level Containers - JFrame

- Créer un nouveau projet Java sous le nom *IHMTP9*.
- Créer dans celui-ci une classe *MyJFrame* héritant de *JFrame* (extends *JFrame*).
- Pour configurer l'état initial de la fenêtre créée, ajouter une méthode *initJFrame* et appeler celle-ci depuis un constructeur. Utiliser les méthodes suivantes pour l'initialisation :
 - ✓ *setTitle(String title)*
 - ✓ *setSize(int width, int height)*
 - ✓ *setLocationRelativeTo(null)*
 - ✓ *setResizable(boolean resizable)*
 - ✓ *setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)*
- Ajouter une classe main nommée *TestJFrame*. Créer dans cette dernière une instance de la classe *MyJFrame* qui devrait s'exécuter dans l'Event Dispatch Thread (en utilisant *SwingUtilities*).
- Pour afficher cette instance, utiliser la méthode *setVisible(true)*.

II. Les boîtes de dialogue

Il s'agit d'une petite fenêtre pouvant servir à plusieurs choses :

- ✓ afficher une information (message d'erreur, d'avertissement...) ;
- ✓ demander une validation, une réfutation ou une annulation ;
- ✓ demander à l'utilisateur de saisir une information dont le système a besoin ;
- ✓ etc.

Les boîtes de dialogue s'utilisent, à l'exception des boîtes personnalisées, avec l'objet **JOptionPane**. Pour faire une boîte de dialogue personnalisée, vous devez créer une classe héritée de **JDialog** (vu en TP1).

La classe *JOptionPane* fournit des façons simples de créer des boîtes de dialogue élémentaires *modales* (i.e. lorsqu'est active toute interaction avec les autres fenêtres sont bloquées) en spécifiant un message, un titre, une icône, et un type de message ou un type d'option. Si on ne spécifie pas d'icône, des icônes sont fournies par le système en fonction du type de message.

1 Les boîtes d'information

Ces boîtes ne sont pas destinées à participer à de quelconques opérations : elles affichent juste un message à l'attention de l'utilisateur.

Les boîtes de message informatif sont créées avec la méthode ***showMessageDialog*** (Component parentComponent, String message, String title, int messageType) de JOptionPane¹, où :

- ✓ Component parentComponent : correspond au composant parent ;
- ✓ String message : permet de renseigner le message à afficher dans la boîte de dialogue.
- ✓ String title : permet de donner un titre à l'objet.
- ✓ int messageType : permet de savoir s'il s'agit d'un message d'information, de prévention ou d'erreur. Il peut être JOptionPane.ERROR_MESSAGE, JOptionPane.INFORMATION_MESSAGE, JOptionPane.QUESTION_MESSAGE, JOptionPane.WARNING_MESSAGE.

Il existe deux autres méthodes *showMessageDialog()* pour cet objet : une qui prend deux paramètres en moins (le titre et le type de message), et une qui prend un paramètre en plus (l'icône à utiliser).

- Depuis la classe main ***TestJFrame***, créer et afficher les trois boîtes de dialogue d'information ci-dessous comme suit :

```
JOptionPane.showMessageDialog(null, "Message Informatif",  
"Information", JOptionPane.INFORMATION_MESSAGE);  
  
JOptionPane.showMessageDialog(null, "Message Préventif",  
"Attention", JOptionPane.WARNING_MESSAGE);  
  
JOptionPane.showMessageDialog(null, "Message d'Erreur", "Erreur",  
JOptionPane.ERROR_MESSAGE);
```



2 Les boîtes de confirmation

Comme leur nom l'indique, ces dernières permettent de valider, d'invalider ou d'annuler une décision. L'objet JOptionPane est aussi utilisé, mais ce sera cette fois avec la méthode ***showConfirmDialog()***, une méthode qui retourne un entier correspondant à l'option choisie dans cette boîte : Yes (0), No (1), Cancel (2), et Closed (-1).

- Depuis la classe main ***TestJFrame***, créer et afficher une boîte de confirmation comme suit:

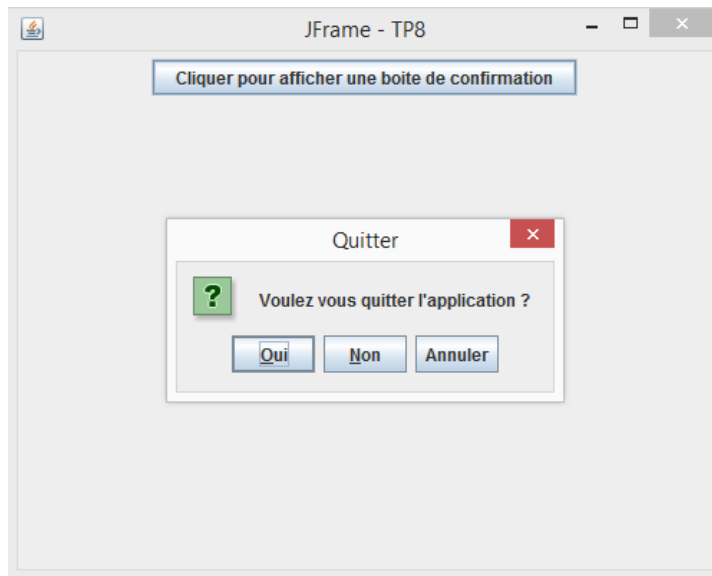
¹ <https://openclassrooms.com/courses/apprenez-a-programmer-en-java/les-menus-et-boites-de-dialogue>

```
int rep = JOptionPane.showConfirmDialog(null,"Voulez vous créer et
afficher une fenêtre MyJFrame ?", "Bonjour",
JOptionPane.YES_NO_OPTION);

if (rep == JOptionPane.YES_OPTION)
    //TODO si oui.

else
    //TODO si non.
```

- Modifier MyJFrame pour ajouter un JButton *button*. Initialiser son texte à "Cliquer pour afficher une boîte de confirmation".
- Modifier *actionPerformed* pour qu'elle crée et affiche la boîte de confirmation YES_NO_CANCEL suivante à chaque clic sur *button* :



3 Les boîtes de saisie

Dans ce type de boîte de dialogue, c'est la méthode *showInputDialog* qui est utilisée, qui retourne une chaîne de caractères.

- Depuis la classe main *TestJFrame*, créer et afficher une boîte de saisie comme suit:

```
JOptionPane jop = new JOptionPane();

jop.showInputDialog(null,"Quel est votre album préféré ?", "Deezer",
JOptionPane.QUESTION_MESSAGE);
```

C'est une boîte de dialogue de type QUESTION_MESSAGE. Le texte sur les boutons et le titre sont par défaut et dépendent de la langue utilisée par le système.

Il est possible d'offrir une liste de réponses possibles avec *showInputDialog* comme suit :

```
String[] bands = {"Iron Maiden", "Dream Theater", "Opeth"} ;

String s = (String) JOptionPane.showInputDialog(null, "Choisissez
le groupe à écouter", "Sélection groupe",
JOptionPane.QUESTION_MESSAGE, null, bands, bands[0]);

if(s!=null)

    //TODO si pas annulé
```

Le nombre de boutons dépend du paramètre *optionType*. Le deuxième paramètre "null" de *showInputDialog* correspond à l'icône à faire passer (ici null=par défaut) et dernier paramètre correspond à la valeur par défaut de la liste déroulante.

4 JFileChooser

Il existe dans javax.swing une classe permettant d'afficher une fenêtre de dialogue destinée à ouvrir ou sauvegarder un fichier en naviguant dans ses répertoires ; cette classe se nomme **JFileChooser**.

- Modifier *MyJFrame* pour ajouter un **JButton** *button2*. Initialiser son texte à "Select File".
- Modifier *actionPerformed* pour qu'elle crée et affiche un **JFileChooser** à chaque clic sur *button2* comme suit :

```
JFileChooser fileChooser = new JFileChooser();

int returnValue = fileChooser.showOpenDialog(null);

if (returnValue == JFileChooser.APPROVE_OPTION) {

    File selectedFile = fileChooser.getSelectedFile();

    System.out.println(selectedFile.getName());

}
```

La méthode *showOpenDialog* affiche la boîte de dialogue à l'écran. Le **JFileChooser.APPROVE_OPTION** est renvoyé lorsque les boutons Oui ou OK sont cliqués. Enfin, le nom du fichier sélectionné est obtenu puis affiché à la console.

Il est possible de filtrer le type de fichiers à afficher selon leur extension comme suit :

```
FileFilter filtr = new FileNameExtensionFilter("text files","txt");

fileChooser.addChoosableFileFilter(filtr);
```