

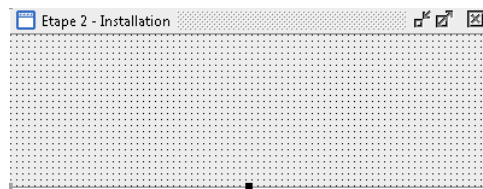
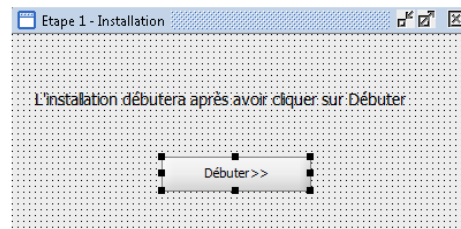
ANNEXE 9 – Les JDialog

Une fenêtre de dialogue est une fenêtre avec moins d'options qu'un `JFrame` habituel. Elle ne peut pas s'exécuter et est habituellement dépendante d'une `JFrame` qu'on appellera son « parent ».

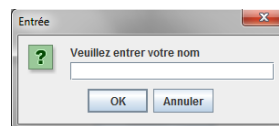
Les fenêtres de dialogue peuvent prendre différentes formes :

- A) **pour des fenêtres de dialogue simples**, on peut utiliser la classe `JOptionPane`, qui crée en fait des `JDialogs` suivant certains modèles, entre autres :
- `JOptionPane.showMessageDialog` : communique une information à l'utilisateur
 - `JOptionPane.showInputDialog` : demande à l'utilisateur d'entrer une réponse dans un champ texte
 - Retourne une `String`, valant `null` si l'utilisateur clique sur Annuler
 - `JOptionPane.showConfirmDialog` : demande une question de confirmation , avec 3 boutons (Oui / Non / Annuler) pour répondre

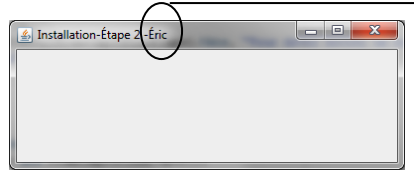
EXERCICE 1 : Dessinez deux `JFrame` représentant les premières étapes de l'installation du logiciel ImpotPlus :



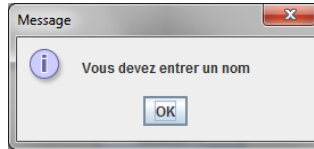
- Dans le premier `JFrame`, gérez le clic de souris de manière à afficher un `JOptionPane` ayant l'apparence suivante :



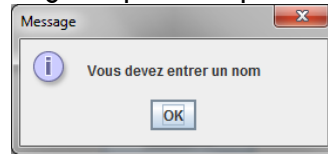
- Si l'utilisateur clique sur Annuler, le `JOptionPane` disparaît (*trivial*)
- Si l'utilisateur entre son nom puis clique sur OK, faites disparaître la première fenêtre et démarrez la deuxième fenêtre en affichant le nom dans le titre de la deuxième fenêtre :



- Si l'utilisateur n'entre pas son nom et clique sur OK, faites apparaître un JOptionPane :



Et l'utilisateur reprend le processus...



B) pour des fenêtres de dialogues spécialisées, des classes existent :

- JFileChooser : ouvre une fenêtre de style explorateur
- JColorChooser : ouvre une fenêtre permettant de choisir une couleur parmi un spectre large

→ on a le choix d'appliquer un de ces objets sur un JPanel ou un JFrame ou (plus simple) de la faire afficher à l'aide de la méthode showDialog.

C) finalement on peut créer nos propres fenêtres de dialogues en créant des JDialogs dans JDeveloper au lieu des JFrames (New → Client Tier → Swing /AWT → Dialog)

Les JDialogs

→ le principal constructeur à utiliser lorsqu'on veut créer un JDialog à partir d'un JFrame (il est d'ailleurs généré automatiquement lorsqu'on crée le fichier JDialog) prend 3 paramètres :

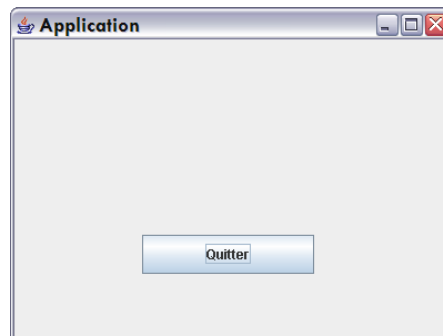
- une référence à la fenêtre parent (le frame d'où origine le JDialog)
 - comment référer au JFrame lorsqu'on est dans une classe d'écouteur ?

*** cette référence au JFrame est très importante quand on veut passer des informations de la fenêtre de dialogue au JFrame principal (Transtypage)

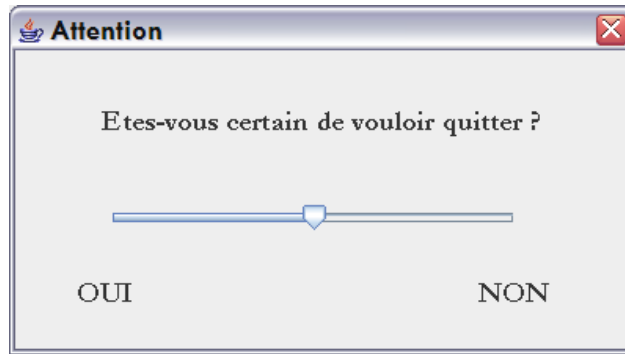
explications :

- le titre du `JDialog` (une `String`)
- le paramètre `modal` : si ce paramètre vaut `true`, l'apparition d'une fenêtre de dialogue bloquera toutes les autres fenêtres de l'application. Si faux, on peut revenir sur les autres fenêtres de l'application pour les utiliser
- On peut bien entendu créer ses propres constructeurs dans un `JDialog` et dessiner l'interface, faire la gestion d'événements de la même manière qu'un `JFrame`

EXERCICE 2 : Réaliser une application contenant un `JFrame` représentant la fin de l'application, comme ceci :



Cliquer sur le bouton `Quitter` devra entraîner l'apparition d'une fenêtre de dialogue (`JDialog`) qui aura l'apparence suivante :



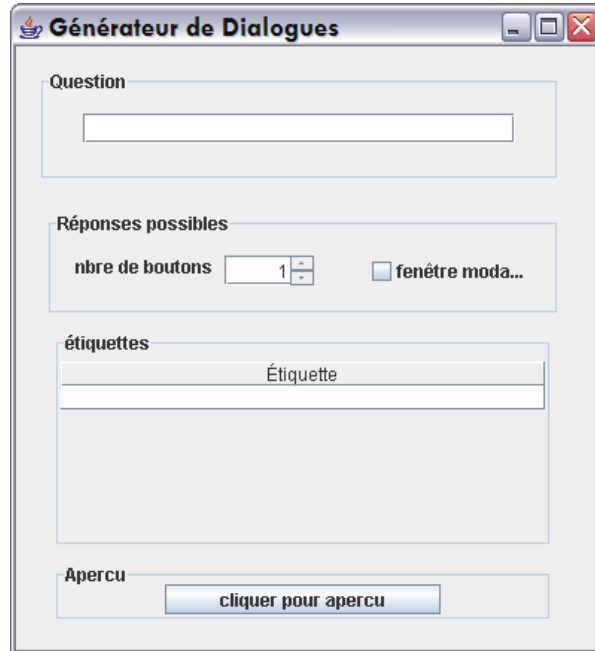
Pour un brin d'originalité, j'ai utilisé un `JSlider` plutôt que des boutons. Comme vous le deviner, si l'utilisateur amène le curseur vers OUI, l'application au complet sera fermée. S'il clique sur NON, le `JDialog` sera fermé mais l'application (et le `JFrame`) sera présent.

En cherchant sur le web et/ou dans l'API et /ou la fenêtre de propriétés, tentez de résoudre les problèmes suivants en construisant votre travail. Nous reviendrons là-dessus en classe.

- A) Comment établir des valeurs minimum et maximum pour le `JSlider` ?
- B) Comment faire en sorte que l'utilisateur ne puisse pas laisser le curseur au $\frac{3}{4}$ de l'échelle ?
- C) À quel type d'écouteur répond un `JSlider` ?
- D) Quelle méthode fait partie de cette interface-écouteur ?
- E) Comment faire en sorte que l'événement se déclenche uniquement lorsque l'utilisateur a choisi une des deux options plutôt que dès qu'il bouge d'un millimètre le curseur du `JSlider` ? (j'ai trouvé la réponse sur le web)
- F) Comment faire pour fermer l'application ?
- G) Comment fermer le `JDialog` sans fermer l'application ?

EXERCICE 3 : Nous créerons ici une application qui permettra de générer un `JDialog` à partir des demandes de l'utilisateur, contenues dans un `JFrame`

Créez d'abord le `JFrame` suivant :



- L'utilisateur écrira donc sa question, le nombre de boutons qu'il veut pour permettre les réponses possibles et écrira dans la `JTable` les étiquettes qui se retrouveront sur les boutons. Une gestion d'événement sur le `JSpinner` (nombre de boutons) ajoutera une rangée à la `JTable` à chaque augmentation du `JSpinner`.
- Un `JCheckBox` sera également présent pour permettre à l'utilisateur de choisir si sa fenêtre de dialogue sera modale ou non.
- Un résultat possible suite à un clic sur le bouton aperçu :

The 'Générateur de Dialogues' window is a Java Swing application. It has a title bar with a standard icon and window controls. The main area is divided into three sections: 'Question', 'Réponses possibles', and 'étiquettes'. The 'Question' section contains a text field with the text 'Voulez-vous jouer au scrabble ?'. The 'Réponses possibles' section contains a label 'nbre de boutons' followed by a JSpinner set to the value '3', and a checked checkbox labeled 'fenêtre moda...'. The 'étiquettes' section contains a list box with three items: 'Oui', 'Non', and 'Peut-être'. At the bottom, there is an 'Aperçu' (Preview) section with a button labeled 'cliquer pour aperçu'.

The generated dialog box is titled 'attention' and has a close button in the title bar. It contains the same question 'Voulez-vous jouer au scrabble ?' and three buttons labeled 'Oui', 'Non', and 'Peut-être' arranged horizontally.

- Pour ce faire, vous devrez créer vos boutons dynamiquement, dans le code, dépendant du choix de l'utilisateur
- Le JSpinner fonctionne presque pareillement que le JSlider
- Bon travail !