

TP2 - Sélection du sucre par événements

Le but de ce TP est de mettre en pratique l'ajout d'événements sur des composants et des fenêtres. Nous allons réaliser une modale qui permet de savoir si l'utilisateur souhaite du sucre avec son café.

1. Création de la fenêtre de sélection

Créez une classe **YesNoModal** qui étend de `JFrame`.

Cette classe possède :

- un attribut `label` de type `JLabel`
- un attribut `buttonYes` de type `JButton`
- un attribut `buttonNo` de type `JButton`
- un attribut `buttonCancel` de type `JButton`
- un attribut `result` de type `Boolean`
- un attribut `labelText` de type `String`

Le constructeur de cette classe :

- prendra en paramètre un texte qui sera stocké dans l'attribut `labelText`
- appellera la méthode `initComponents`
- appellera la méthode `initEventListeners`
- se chargera de mettre la valeur `null` dans `result`
- fera un appel au `pack` pour adapter la taille de la fenêtre aux composants

`initComponents` se charge :

- d'initialiser la valeur du label avec le paramètre du constructeur
- de mettre "Oui" dans le premier bouton
- de mettre "Non" dans le deuxième
- de mettre "Annuler" dans le troisième
- de mettre le label dans un `JPanel`
- de mettre les 3 boutons dans un autre `JPanel`
- d'ajouter le panel du label en haut de la fenêtre
- d'ajouter le panel des boutons en bas de la fenêtre

`initEventListeners` se charge :

- d'ajouter un `eventListener` sur le bouton "Oui", qui met `true` dans `result` puis ferme la fenêtre
- d'ajouter un `eventListener` sur le bouton "Non", qui met `false` dans `result` puis ferme la fenêtre
- d'ajouter un `eventListener` sur le bouton "Annuler", qui ferme la fenêtre

Pour fermer la fenêtre vous devez utiliser la méthode `dispose`

Ajoutez ensuite une fonction `getResult` qui retourne le contenu de l'attribut `result`

2. Ajout de la feature de sélection du sucre

Cette partie se fait dans la classe **MachineACafe** ou **Distributeur** si vous n'avez pas dépassé la partie 3 du TP1.

Ajoutez trois attributs pour gérer le sucre des cafés :

- un attribut `sugarButton` de type `JButton`
- un attribut `sugarLabel` de type `JLabel`
- un attribut `sugarModal` de type `YesNoModal`

Dans `initComponents` :

- initialisez le `JButton` avec comme texte "Sélectionner le sucre"
- initialisez le `JLabel` avec comme texte "Sucre :"
- ajoutez ces deux attributs dans un `JPanel`
- ajoutez ce `JPanel` à fenêtre de façon à ce qu'il apparaisse en dessous du contenu déjà existant
- initialisez `sugarModal` avec comme texte "Voulez-vous du sucre ?" passé en paramètre

Afin de gérer l'affichage du choix du sucre, ajoutez une méthode `updateLabelSugar` qui se charge de :

- récupérer le résultat dans la modale de sélection du sucre (via la fonction `getResult`)
- si le résultat est `true` , remplacer le texte du label du sucre par "Sucre : oui" puis faire un `pack`
- si le résultat est `false` , remplacer le texte du label du sucre par "Sucre : non" puis faire un `pack`
- si le résultat est `null` , ne rien faire
- faire un appel au `pack` pour ré-adapter la taille de la fenêtre

Ajoutez une méthode `initEventListeners` qui sera appelée après `initComponents` dans le constructeur.

`initEventListener` se charge d'ajouter un **ActionListener** sur le bouton de sélection du sucre.

Cet événement doit :

- rendre visible la fenêtre de sélection du sucre
- ajouter un **WindowListener** sur cette fenêtre afin de détecter le choix de l'utilisateur

Pour réaliser cette action (via le **WindowListener**) vous devez passer en paramètre un **WindowAdapter** qui va surcharger la méthode `windowDeactivated` (appelée suite à la fermeture de votre modale via `dispose`).

Dans votre implémentation de `windowDeactivated` , après avoir appelé `super.windowDeactivated` vous appellerez `updateLabelSugar` .

Testez.