

Travaux pratiques gestion événementielle

Interface graphique et événements

1 OBJECTIF

Mettre en place une petite interface graphique avec gestion d'événements

2 SUPPORT

Nous allons nous intéresser au clavier de la borne et de la prise en compte des touches du clavier.
Un message sur l'afficheur indiquera le code entré et sa validité

3 RÉALISATION DE L'I.H.M.

Voici le visuel de l'ihm que vous devez réaliser



3.1 Création d'un Projet

- Créez un nouveau projet nommé **SimClavier** dans votre arborescence de Netbeans, de classe principale **.SimClavier**
 - A l'intérieur de ce package, créez une classe visuelle **FenClavier** qui va hériter de JFrame :

Ce sera la fenêtre principale. Donnez-lui pour titre « **Clavier** ».

Parmi les propriétés, positionner :

- non redimensionnable
- dimension 220 X 250
- couleur de fond gris clair (Color.LIGHT_GRAY **ou à votre convenance**)
- Gestionnaire de placement null

Sur ce panneau, placez-y :

- une zone de saisie de texte (JTextField) de nom d'instance **textMess** avec pour caractéristiques
 - message d'origine : une chaîne de caractère vide
 - de couleur à votre convenance
 - non éditable
 - police et de dimension de la police à votre convenance

3.2 Panneau du clavier

- Les touches du clavier se trouveront dans un container de type **JPanel** nommé **fondClavier** ayant les propriétés :
 - couleur noire
 - dimension 120 x 160
 - gestionnaire de placement `GridLayout` et avec pour caractéristiques :
 - 3 colonnes
 - 4 lignes
 - horizontal gap 1
 - vertical gap 1

1. Placez les différentes touches qui seront de type *JButton*

3.3 Les touches

3.3.1 Propriétés d'une touche

Celle-ci aura les propriétés :

- Couleur de fond : blanc.
- Dimension 26 * 26 (taille préférée)
- Texte au centre de la touche
- Police au choix (par ex sans sérif 16)
- Marge autour du libellé du bouton de 2
- Texte centré horizontalement
- Message d'aide **appuyer**
- Bordure au choix.

Il y a 12 touches numérotées de 0 à 9, plus les touches * et #.

Vous donnerez un nom à chacune des touches rappelant leur ordre (touche0 pour la première, etc.).

4 GESTION DES ÉVÉNEMENTS

4.1 Objectif

Il nous reste maintenant à prendre en compte les événements liés au clavier c'est-à-dire principalement l'appui sur les touches.

4.2 Prise en compte de l'appui touche

Lorsqu'une touche est appuyée, un événement est généré et peut être intercepté par un écouteur. Nous allons prendre en compte ce fonctionnement.

L'action à effectuer lors du clique sur une touche étant la même pour tous les chiffres (afficher ce chiffre dans l'afficheur), nous allons passer par un gestionnaire d'événements.

- Créez une classe interne à la classe **FenClavier** en recopiant le code suivant:

```
public class GestionnaireTouche implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent e) {
        // recuperation de l'objet à l'origine de l'evenement
        Object src = e.getSource();
        // si l'objet est un bouton
        if (src instanceof JButton) {
            // recuperer le texte du bouton
            String texteDeLaTouche = ((JButton) src).getText();
            // ajouter le texte du bouton au texte deja present
            // dans la zone textMess

        }
    }
}
```

- Complétez le code fourni afin de réaliser la mise à jour de la zone *textMess*.
- Il reste à associer le gestionnaire de touche à toutes les touches numérique. Cela se fait dans le constructeur de la classe **FenClavier**.
- Voici un exemple pour la touche "0" (à placer après l'appel de la méthode *initComponents*) :
`jButton0.addActionListener(new GestionnaireTouche());`
- Ajoutez un événement pour le clique sur la touche "#" avec la souris (il faut se positionner sur le composant en l'occurrence ici la touche puis avec le menu contextuel, ajouter un événement pour la souris et générer un adaptateur pour cet événement.
 - Cliquer sur "#" devra effacer le contenu de la zone *textMess*.
- Ajoutez un événement pour le clique sur la touche "*" avec la souris (il faut se positionner sur le composant en l'occurrence ici la touche puis avec le menu contextuel, ajouter un événement pour la souris et générer un adaptateur pour cet événement.
 - Cliquer sur "*" devra tester si le contenu de la zone *textMess* est égal à la chaîne "1234".
 - Si oui, la couleur de fond de *textMess* passera au vert.
 - Si non, la couleur de fond de *textMess* passera au rouge.
 - Dans tous les cas, le contenu de *textMess* est vidé.

Astuce :

Pour récupérer le texte d'un JButton ou d'un JTextField utilisez la méthode `getText()` de l'objet.