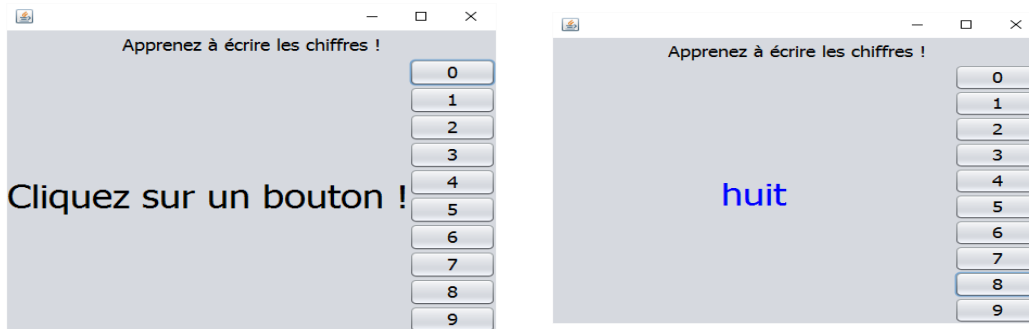


TP1

Exercice 1 :

On souhaite construire une version simplifiée de l'application vue en CM dont l'interface est décrite ci-dessous et qui permet de connaître l'écriture d'un chiffre en lettres lorsque l'on clique sur un bouton contenant un chiffre. Chaque chiffre est écrit en lettres et d'une couleur différente.



Cette interface (JFrame) est organisée en BorderLayout. Elle comporte :

- un panneau en haut (NORTH) organisé en FlowLayout et qui contient un message (JLabel)
- un panneau à droite (EAST) organisé en GridLayout (10,1) et qui contient dix boutons (JButton)
- un panneau au centre (CENTER) organisé en GridLayout(1,1) et qui contient un message (JLabel)

Pour construire cette interface, vous pouvez suivre les étapes suivantes (avec la version Apache NetBeans IDE 12.2)

Etape 1 : Création d'un nouveau projet

File -> **New Project** puis dans la fenêtre qui s'ouvre sélectionnez **Java with ant** et **Java Application** puis **Nextt**.

Dans la fenêtre suivante, choisissez le nom du projet dans : **Project Name** (par exemple *LesChiffres* et le nom du répertoire de travail **Project Location** (par exemple *moduleI2B*). Le répertoire du projet sera alors automatiquement le nom du répertoire de travail (par exemple : *moduleI2B/LesChiffres*).

Vérifiez que la case : « Create Main Class » qui permet la création d'une classe avec un programme principal en mode console, ne soit t pas cochée, sinon décochez la case.

Terminez la création du projet, en cliquant sur **Finish**.

Etape 2 : Ajout d'une classe dérivée de la classe JFrame pour la fenêtre principale

Faites un clic droit sur le nom du projet et choisissez l'option **New -> JFrame Form si cette option est disponible ou bien faites Nouveau -> Other... -> Swing GUI Forms -> JFrame Form**. Dans la fenêtre qui s'ouvre choisissez le nom de la classe dérivée de JFrame (par exemple *LesChiffres*). Une classe avec son design pour dessiner l'interface sont créés. La classe générée comporte un programme principal qui est de la forme :

```
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new LesChiffres().setVisible(true);
        }
    });
}
```

Ce programme principal peut être modifié pour choisir la taille initiale de la fenêtre principale (à l'ouverture), sa position et fixer son titre.

```
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            LesChiffres lc= new LesChiffres();
            lc.setSize(800,800) ;
            lc.setTitle("Les Chiffres");
            lc.setVisible(true);
        }
    });
}
```

Etape 3 : Construction de l'interface.

Dans l'onglet **Navigateur** on voit les éléments constituant l'interface sous forme d'une arborescence de composants : le composant **JFrame** et une entrée pour les composants non visibles « **Autres composants** » initialement vide. Pour choisir la stratégie d'organisation de la fenêtre principale, cliquez droit sur le composant **JFrame**, et choisissez **Set Layout** -> BorderLayout. La stratégie choisie s'affiche en dessous du composant.

Pour ajouter des panneaux dans la fenêtre principale, vous pouvez :

- soit cliquer droit sur le composant JFrame et choisir l'option **Add from palette** -> **Swing Containers** -> **Panel**
- soit faire afficher la palette de composants par l'option **Window** -> **IDE Tools** -> **Palette** (si elle n'est pas déjà visible), et cliquer sur le composant choisi (ici Panel) et le glisser sur l'interface.

En suivant les indications, ajoutez 3 panneaux dans la fenêtre principale. Pour fixer la disposition d'un panneau dans la fenêtre, cliquez droit sur le panneau, choisissez « Propriétés » et fixez la propriété « Direction » qui est dans la catégorie « Layout » à la valeur souhaitée NORTH, EAST ou CENTER.

Fixez ensuite la stratégie choisie pour chaque panneau : FlowLayout pour le panneau du nord, GridLayout pour les 2 autres panneaux. Pour fixer le nombre de lignes et de colonnes pour une organisation en grille, cliquez droit sur la stratégie (GridLayout), et choisissez « Propriétés », vous pouvez alors fixer les valeurs « Rows (lignes) » et « Columns (colonnes) ».

Ajoutez ensuite les composants dans les panneaux, soit en utilisant la palette soit en utilisant le menu contextuel sur le panneau, « Add from palette -> Swing Controls -> Button (ou Label) ».

Vous pouvez ensuite modifier les propriétés de vos composants :

- Modifier la taille de la fonte (font) pour les deux messages (18 pour le message en haut et 36 pour celui du centre) et fixer leurs textes initiaux.
- Modifier le texte sur les boutons pour mettre les chiffres, on peut aussi modifier la taille de la fonte d'écriture et la taille préférée (preferredSize) du bouton.
- Il est également possible de modifier le nom d'un composant (Clic droit sur le composant -> Change Variable Name) si on le souhaite ou son texte (Clic droit sur le composant -> Edit Text). Par exemple, en faisant un clic droit sur le JLabel du centre, choisissez « Change Variable Name » et mettez « Message ».

Etape 4 : Compilation et exécution

Pour fixer le projet par défaut (s'il y en a plusieurs qui sont ouverts), choisissez dans le menu « Run » l'option « Set Main Project » et sélectionnez le projet voulu dans la liste.

Construisez le projet (Run -> Build Project et exécutez-le (Run -> Run Project)

Pour rendre réactif votre programme, il faut abonner les boutons à des écouteurs d'événements et donner les instructions à exécuter dans le gestionnaire de l'évènement associé.

Pour cela vous pouvez :

- Soit cliquer droit sur un bouton, et sélectionner « Properties », dans la fenêtre qui est ouverte, cliquez sur l'onglet « Events », sélectionner la case « actionPerformed » et valider le nom du gestionnaire proposé (par défaut nom du composant suivi de actionPerformed).
- Soit cliquer droit sur le bouton, et sélectionner « Events » puis « Action » et enfin « actionPerformed »
- Soit double-cliquer sur le bouton.

Dans tous les cas, le code de l'abonnement à l'écouteur est généré dans la méthode « initComponents » et l'entête du gestionnaire pour le bouton concerné est également générée dans le programme et doit être complétée.

Par exemple, ici, pour le bouton 1 (nommé jButton1), le code suivant pourra être écrit dans le gestionnaire :

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    // ici, Message désigne le nom du JLabel qui a été renommé lors de la construction de l'interface  
    Message.setForeground(Color.green); // changement de la couleur d'écriture  
    Message.setText("zéro"); // changement du texte à afficher  
}
```

Il faut ajouter les bibliothèques en mettant « import java.awt.* ; » et « import javax.swing.* ; » au début du fichier. Faites le même travail, avec différentes couleurs pour chaque bouton et le texte correspondant au chiffre du bouton.

Exercice 2 :

On souhaite compléter cette application en intégrant de nouvelles fonctionnalités qui permettent de changer la couleur d'écriture du texte (boutons rouge, jaune, vert, orange, bleu) et la taille des caractères (14, 24 ou 36).



L'option « LesChiffres » du menu ne comporte qu'une sous-option « Quitter » qui ferme l'application. Pour construire le menu, il est nécessaire d'ajouter un composant de type « JMenuBar », puis dans cette barre construire un composant de type « JMenu » pour l'option « LesChiffres », et enfin construire un composant de type « JMenuItem » pour la sous-option « Quitter ».

Après l'ajout de la barre de menu, l'ajout des options et des sous-options peut se faire directement dans la fenêtre de design. L'ajout d'une action sur la sous-option « Quitter » se fait de la même façon que sur un bouton.

1. Compléter l'interface de l'application précédente pour ajouter les boutons des couleurs et les boutons des tailles.
2. Ajouter des actions sur les boutons des couleurs. Par exemple, lors du clic sur le bouton rouge nommé BRouge de type JButton, le code du gestionnaire est le suivant :

```
private void BRougeActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    Message.setForeground(Color.red) ;  
}
```

3. Ajouter les actions sur les boutons des tailles. Par exemple, lors du clic sur le bouton de taille 14, le code du gestionnaire du bouton nommé BTaille14 de type JButton, le code du gestionnaire est le suivant :

```
private void BTaille14ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    Message.setFont(new java.awt.Font("Tahoma", 0, 14));  
}
```

4. Ajouter une action sur la sous-option « Quitter » qui permet la fermeture de la fenêtre en utilisant le code suivant pour le gestionnaire.

```
System.exit(0) ;
```

5. Compléter l'application sous une forme au choix pour permettre le changement de la couleur de fond du panneau contenant le message central. Par exemple, si le panneau s'appelle PMessage de type JPanel, le code à exécuter serait le suivant :

```
PMessage.setBackground(new Color(102, 255, 51));  
pour la couleur RVB(102,255,51).
```

```
ou bien PMessage.setBackground(Color.yellow); pour du jaune.
```