TP 10

Les boites de dialogues

L'apparition des boites de dialogues sont dues en général au traitement d'autres évènements. Par exemple, un champ non ou mal rempli. La boite de message invite l'utilisateur à corriger sa saisie. La disparition des boites de dialogues est due en cliquant sur l'un des boutons contenus dans la boite. Donc, pas d'interface à implémenter en principe.

Exercice 1

- Ecrire un programme qui permet d'afficher une simple fenêtre avec une taille, un titre, une couleur que vous aurez choisis. A l'intérieur de cette fenêtre, se trouve une autre fenêtre qui est tout simplement la boite de dialogue dont le titre est "Select an option" ou "Sélectionnez une option" si votre environnement a été traduit en français. Et le message est "Avez-vous bien lu ?".
  Sur la fenêtre graphique, se trouve une étiquette dont le nom est "texte initial".
  Si on clique sur le bouton Oui ou Yes, l'étiquette affiche le message "clic sur bouton Oui ou Yes" Fermez alors la fenêtre graphique. Puis exécutez à nouveau. La boite de confirmation s'affiche à nouveau. Cliquez sur un autre bouton et le message correspondant sera affiché sur l'étiquette.

Exercice 2

- Ecrire un programme qui permet d'afficher une simple fenêtre avec une taille, un titre, une couleur que vous aurez choisis. A l'intérieur de cette fenêtre, se trouve une autre fenêtre qui est tout simplement la boite de confirmation dont le titre est "titre". Et le message est "message de la boite". En plus de ça, on y voit un seul bouton Ok et l'image point d'interrogation dans un carré. Cliquez sur le bouton Ok, la boite disparaît et le message de l'étiquette devient "clic sur bouton Oui ou Yes".

Exercice 3

- Ecrire un programme qui permet d'afficher une simple fenêtre avec une taille, un titre, une couleur que vous aurez choisis. A l'intérieur de cette fenêtre, se trouve une autre qui est tout simplement la boite de saisie dont le titre est "Input" (ou "Entrée" en Français). En plus de ça, on y voit un champ de saisie dans lequel vous pouvez saisir un texte. Le message de la boite de dialogue est "Ici, votre texte". On y voit aussi 2 boutons Ok et Cancel (ou en Français, Ok et Annuler). Entrez un texte et cliquez sur le bouton OK le texte donné sera affiché à la console. Puis exécutez à nouveau cette fois ci cliquez sur Cancel, résultat est ?
  - Changez la valeur du quatrième paramètre (-1) en 0. Puis en 1. Puis en 2. Et enfin en 3. A chaque fois, recompilez puis exécutez. Notez les différences.

Exercice 4

- Ecrire un programme qui permet d'afficher une simple fenêtre avec une taille, un titre, une couleur que vous aurez choisis. A l'intérieur de cette fenêtre on trouve la boite d'entrées. Cette boites contient 2 boutons : OK et Annuler (ou Cancel). En plus de ça, on a une liste de noms dans une boite combo. Le nom présélectionné est en position

spécifié dans le septième paramètre. Si on sélectionne un élément de la liste déroulante puis on clique sur Ok l'élément sélectionné sera affiché à la console.

Exercice 5

- Ecrire un programme qui permet d'afficher une simple fenêtre avec une taille, un titre, une couleur que vous aurez choisis et à l'intérieur de laquelle on trouve la boite d'options. Cette boite contient une liste de boutons disposés sur une ligne. Chaque bouton possède un nom. Le bouton qui possède le focus est en position spécifié dans le septième paramètre. Si on clique sur l'un des boutons puis on clique sur Ok le nom du bouton cliqué sera affiché à la console.

Les Threads

Un thread est, en gros, une séquence d'instructions qui s'exécutent parallèlement aux autres threads. Chaque programme est constitué d'au moins un thread : le thread principal, qui fait tourner votre fonction main(). Les programmes qui utilisent uniquement le thread principal sont *monothreadés*, si vous leur ajoutez un ou plusieurs threads alors ils deviennent *multithreadés*.
Testez et analysez les exemples suivants :

Exemple 1

```
public class NouveauThread1 {
public static void main(String[] args) {
      Thread t = new Thread() {
// on peut nommer le thread new Thread("Mon thread");
public void run() {
      System.out.println("Je suis bien dans le thread: "
                  + Thread.currentThread().getName());
      }
};
t.start(); // à tester avec t.run() donne je suis dans le thread: main

System.out.println("Je suis dans le thread: "
      + Thread.currentThread().getName());
      }
}
```

Exemple 2

```
public class NouveauThread2 extends Thread {
@Override
public void run() {
System.out.println("Je suis bien dans le thread: "
            + Thread.currentThread().getName());

}

public static void main(String[] args) {
Thread t = new NouveauThread2();
t.start(); // à tester avec t.run() donne je suis dans le thread: main

System.out.println("Je suis dans le thread: "
      + Thread.currentThread().getName());
      }
```

```
}
```

Exemple 3

```
public class NouveauThread3 implements Runnable {

@Override
public void run() {
System.out.println("Je suis bien dans le thread: "
            + Thread.currentThread().getName());

}

public static void main(String[] args) {
Thread t = new Thread(new NouveauThread3());
t.start(); // à tester avec t.run() donne je suis dans le thread: main

System.out.println("Je suis dans le thread: "
            + Thread.currentThread().getName());
}
}
```

Exemple 4

```
class NouveauThread4 extends Thread {
public NouveauThread4(String nom) {
      super(nom);
}

public void run() {
for (int i = 0; i < 10; i++) {
      System.out.println(getName() + " " + i);
try {
// la méthode sleep doit toujours être appelée dans un bloc try...catch
      sleep((long) (Math.random() * 1000));
} catch (InterruptedException e) {
}
System.out.println(getName() + " terminé");
}
}

public static void main(String[] args) {
      new NouveauThread4("Taza").start();
      new NouveauThread4("Casa Blanca").start();
}
}
```

Exemple 5

```
class NouveauThread5 implements Runnable {
public void run() {
for (int i = 0; i < 3; i++) {
      System.out.println(Thread.currentThread().getName() + " " + i);
      try {
            Thread.currentThread().sleep((long) (Math.random() * 1000));
      } catch (InterruptedException e) {
      }
      System.out.println(Thread.currentThread().getName() + " terminé");
}

      System .out. println ("Je suis un thread !");
}
```

```
public static void main(String[] args) {
new Thread(new NouveauThread5(),"Taza").start();
new Thread(new NouveauThread5(),"Casa Blanca").start();
}
}
```

Exemple 6

```
public class NouveauThread6 {
public static void main(String[] args) {
// le traitement encapsulé dans un Runnable
Runnable traitement = new Runnable() { // #1
public void run() {
      System.out.println("Je suis dans le thread: "
                  + Thread.currentThread().getName());
}
};
Thread t1 = new Thread(traitement, "Premier thread"); // #2
t1.start();
Thread t2 = new Thread(traitement, "Second thread");
t2.start();
System.out.println("Je suis dans le thread: "
                  + Thread.currentThread().getName());
}
}
```

Exemple 7

```
import java.awt.*;
import java.text.DateFormat;
import java.util.Calendar;
import java.util.Date;

import javax.swing.*;

public class CadreAvecHeure extends JFrame {
      private String titre;
      private JTextField heures;
      private JTextField minutes;
      private JTextField secondes;

      public CadreAvecHeure(String titre) {
            super(titre);
            this.titre = titre;
            Container c = getContentPane();
            c.setLayout(new FlowLayout());
            this.heures = new JTextField(20);
            this.minutes = new JTextField(20);
            this.secondes = new JTextField(20);

            new Thread(new Runnable() {
                  public void run() {
                        gererAffichageHeure();
                  }
            }).start(); // implémentation de Runnable
            setDefaultCloseOperation(EXIT_ON_CLOSE);
            setSize(600, 400);
            c.setBackground(Color.RED);
            c.add(heures);
            c.add(minutes);
            c.add(secondes);
```

```
            setVisible(true);
      }

      public void gererAffichageHeure() {
            while (true) {
                  try {
                        Thread.sleep(1000);
                  } catch (InterruptedException e) {
                  }
                  Date h = Calendar.getInstance().getTime();
                  String s = DateFormat.getTimeInstance().format(h);
                  setTitle(titre + " " + s);

                  secondes.setText( ""+h.getSeconds());
                  minutes.setText(""+h.getMinutes());
                  heures.setText(""+h.getHours());
            }
      }

      public static void main(String[] args) {
            new CadreAvecHeure("Test affichage heure");
      }
}
```

Exemple 8

```
import java.awt.*;
import java.text.DateFormat;
import java.util.Calendar;
import java.util.Date;

import javax.swing.*;

public class CadreAvecHeure2 extends JFrame {
      private String titre;
      private JTextField heures;
      private JTextField minutes;
      private JTextField secondes;
      boolean go = true;

      public CadreAvecHeure2(String titre) {
            super(titre);
            this.titre = titre;
            Container c = getContentPane();
            c.setLayout(new FlowLayout());
            this.heures = new JTextField(20);
            this.minutes = new JTextField(20);
            this.secondes = new JTextField(20);

            new Thread(new Runnable() {
                  public void run() {
                        gererAffichageHeure();
                  }
            }).start(); // implémentation de Runnable
            setDefaultCloseOperation(EXIT_ON_CLOSE);
            setSize(600, 400);
            c.setBackground(Color.RED);
            c.add(heures);
            c.add(minutes);
```

```
            c.add(secondes);

            setVisible(true);
      }

      public void gererAffichageHeure() {
            while (go) {
                  try {
                        Thread.sleep(1000);
                  } catch (InterruptedException e) {
                  }
                  Date h = Calendar.getInstance().getTime();
                  String s = DateFormat.getTimeInstance().format(h);
                  setTitle(titre + " " + s);

                  secondes.setText( ""+h.getSeconds());
                  minutes.setText(""+h.getMinutes());
                  heures.setText(""+h.getHours());
                  }
      }

      public void arreterLaPendule() {
            go = false;
      }

      public static void main(String[] args) {
      CadreAvecHeure2 p = new CadreAvecHeure2("Test affichage heure");
            p.arreterLaPendule();

      }
}
```