

TP Application fenêtrée java

L'application Bonjour

Le principe des applications fenêtrées en Java est proche de celui de QT en C++. Il y a 2 bibliothèques principales AWT et SWING (compatible entre elles). On trouve des conteneurs (JFrame, JPanel,..) qui contiennent des composants (JLabel, JButton,...) et d'autres conteneurs éventuellement .

Event et Listener

Certains composants peuvent envoyer un *event* . Pour être pris en compte il faut associer un composant listener, qui comporte une méthode pour gérer l'*event*.

Mise en œuvre :

- on peut utiliser l'interface adéquat. une interface est une espèce de classe virtuelle , avec des méthodes virtuelles de gestion des *event* qu'il faudra définir. On hérite d'une interface avec l'instruction *implements* (et non *extends*).
- Le générateur de code de NetBeans, génère une classe virtuelle avec la méthode de gestion de l'événement qui appelle une méthode que vous n'avez plus qu'à compléter ...

Création de l'application :Bonjour

Déclaration et constructeur

On crée une classe bonjour qui étend la classe Frame et qui implémente l'interface WindowListener pour réagir en particulier à l'événement fermeture de la fenêtre.

```
import java.awt.*;
import java.awt.event.*;

public class Bonjour extends Frame
    implements WindowListener {

    public Bonjour() {
        setBackground(Color.white);
        setSize(300,200);
        setTitle("Essai");
        addWindowListener(this);
    }
}
```

Nous avons utilisé le constructeur pour choisir une couleur de fond (setBackground), une taille (setSize) et un titre (setTitle). Enfin nous redirigeons les événements liés à la fenêtre vers elle-même.

Interface WindowListener

L'interface WindowListener nous oblige à définir 5 méthodes.

```
public void windowClosing(WindowEvent e) {
    System.exit(0);
}

public void windowClosed(WindowEvent e) {}
```

```

public void windowDeiconified(WindowEvent e) {}

public void windowIconified(WindowEvent e) {}

public void windowActivated(WindowEvent e) {}

public void windowDeactivated(WindowEvent e) {}

public void windowOpened(WindowEvent e) {}

```

Seule la méthode `windowClosed` est complétée pour provoquer l'arrêt de l'application lors de la fermeture de la fenêtre.

Ajout d'un Button

Ajouter dans le constructeur :

```

this.setLayout(null); // Pas de placement auto dans le frame(svt on évite)
JButton b1 = new JButton();
b1.setText("Bonjour");
b1.setSize(100, 100); // on dimensionne
b1.setLocation(10,10); // on positionne
this.add(b1); // ajout dans le frame

```

Designer – génération code

Netbean comporte un générateur de code qui génère `initcomponent()`.

C'est une façon rapide de coder les écran !

C'est une façon rapide d'installer les gestionnaires *d'évent*.

Méthode paint

La méthode `paint()` peut être définie pour personnaliser l'affichage (respecter le prototype)

Si nous désirons dessiner un trait et un rond nous réécrivons la méthode `paint`.

```

public void paint(Graphics g) {
    int x=(this.getSize().width-80)/2;
    int y=(this.getSize().height-20)/2;
    g.drawString("Bonjour !", x,y);
}

```

Les coordonnées d'affichage sont calculées pour que le texte soit à peu près centré dans la fenêtre.

Méthode main

La méthode `main` affiche un message, crée une instance de la fenêtre et l'affiche.

```

public static void main(String args[]) {
    System.out.println("Chargement en cours ...");
    bonjour b=new bonjour();
    b.show();
}

```

Le fichier `bonjour.java` peut être compilé et exécuté.

Associer une icône à la fenêtre

Par défaut, les fenêtres créées ont une icône représentant une tasse de café (Java oblige). On peut leur associer une autre icône fournie sous forme d'image gif de dimension 16x16 avec l'instruction :

```
setIconImage(Toolkit.getDefaultToolkit().getImage("fic.gif"));
```

Exécution de l'application

A partir d'une console DOS

Il suffit d'entrer la ligne de commande :

```
java bonjour
```

La fenêtre de l'application s'ouvre; elle peut être déplacée, redimensionnée ou réduite dans la barre de tâches. On quitte l'application en fermant la fenêtre. On peut bien entendu, créer un `.bat` pour assurer le lancement de cette façon.

En utilisant un raccourci

On commence par créer un raccourci vers le fichier `bonjour.class` dans son propre dossier. On édite ensuite les propriétés de ce raccourci (clic droit, menu Propriétés). Dans l'onglet Raccourci, on complète la ligne de saisie Cible en la faisant commencer par `jview` (programme Windows).

On peut aussi écrire `javaw bonjour`. Le programme `javaw` permet d'exécuter une application java sans ouvrir de console DOS.