Boîtes de dialogue

Table des matières

Boîte de message

La classe MsgBox

Cette classe représente une boîte de dialogue destinée à afficher un message et contenant un bouton "OK" seul ou accompagné par un bouton "Annule".

Déclarations

MsgBox dérive de la classe Dialog. La variable id permettra de déterminer avec quel bouton la boîte de dialogue a été refermée par l'utilisateur.

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

class MsgBox extends Dialog implements ActionListener {
 boolean id=false;  //permet de connaître le bouton utilisé
 Button ok,can;
```

Constructeur

Le constructeur fait d'abord appel au constructeur hérité de la classe Dialog. Il est pour cela nécessite d'indiquer quel objet de type Frame sera parent de la boîte de dialogue. On insère ensuite un Label initialisé avec le paramètre msg pour le message, puis un ou deux boutons selon la valeur du paramètre okcan. On termine en affichant la boîte de dialogue.

```
 MsgBox(Frame fr, String msg, boolean okcan) {
  //constructeur hérité
  super(fr, "Message", true);
  //gestionnaire de positionnement
  setLayout(new BorderLayout());
  //ligne de message
  add(BorderLayout.CENTER,new Label(msg,Label.CENTER));
  //boutons
  Panel p=new Panel();
  p.setLayout(new FlowLayout());
  ok=new Button(" OK ");
  p.add(ok);
  ok.addActionListener(this);
  if (okcan) {
   can=new Button("Annule");
   p.add(can);
   can.addActionListener(this);
  }
  add(BorderLayout.SOUTH,p);
  //dimensions et positionnement
  pack();
  Dimension d=getToolkit().getScreenSize();
  setLocation(d.width/3,d.height/3);
  //affichage
  setVisible(true);
 }
```

Gestion des évènements liés aux boutons

L'appui sur un bouton va fermer la boîte de dialogue en la rendant invisible. Le bouton "OK" mettra le champ id à true et le bouton "Annule" le mettra à false.

```
public void actionPerformed(ActionEvent e) {
 if(e.getSource()==ok) {
  id=true;
  setVisible(false);
 }
 else if(e.getSource()==can) {
  id=false;
  setVisible(false);
 }
}
```

Deux méthodes statiques utiles

Les deux méthodes affMsg et affQuest facilitent l'utilisation de la classe MsgBox. La méthode affMsg affiche un message avec le bouton "OK" seul. La méthode affQuest affiche un message avec les boutons "OK" et "Annule", elle renvoie true si le bouton "OK" a été utilisé et false dans les autres cas.

```
public static void affMsg(Frame fr, String msg) {
 MsgBox message=new MsgBox(fr, msg, false);
 message.dispose();
}

public static boolean affQuest(Frame fr, String msg) {
 MsgBox message=new MsgBox(fr, msg,true);
 boolean rep=message.id;
 message.dispose();
 return rep;
}
```

Exemple d'utilisation dans une applet

Ecrivons une applet contenant uniquement un bouton provoquant l'affichage d'une boîte de message.

Le problème est de trouver l'objet de classe Frame qui sera parent de la boîte de dialogue. On l'obtient en utilisant la méthode **getParent** jusqu'à ce qu'elle fournisse effectivement un objet de classe Frame.

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

public class MsgBoxDemo extends Applet
 implements ActionListener {

 public void init() {
  setLayout(new BorderLayout());
  Button b=new Button(" Message ");
  b.addActionListener(this);
  add(BorderLayout.CENTER,b);
```

```
 }

 public void actionPerformed(ActionEvent e) {
  //on cherche un objet de type Frame contenant l'applet
  Frame fr=null;
  Component parentCourant=this;
  while (parentCourant!=null && fr==null) {
   if (parentCourant instanceof Frame) fr=(Frame)parentCourant;
   else parentCourant=parentCourant.getParent();
  }
  MsgBox.affMsg(fr, "Bonjour dans une boîte de message !");
 }

}
```

Sélection de fichiers

La classe FileDialog

La classe FileDialog, dérivée de la classe Dialog, permet d'obtenir les boîtes de sélection de fichiers fournies par le système.

Constructeur

Le constructeur attend 3 paramètres :

- l'objet Frame parent de la boîte de sélection
- le titre à afficher dans la barre de titre
- le mode de fonctionnement qui est FileDialog.LOAD (ouverture d'un fichier) ou FileDialog.SAVE (sauvegarde d'un fichier)

On pourra donc construire une boîte de sélection de fichiers en écrivant :

```
FileDialog fDial=new FileDialog(this, "Ouvrir", FileDialog.LOAD);
```

en supposant que this représente ici la fenêtre d'une application qui est donc de classe Frame.

Méthodes

Pour utiliser la boîte de sélection de fichiers on fera appel aux méthodes :

- **setVisible(true)** : pour l'afficher
- **getFile()** : pour obtenir le nom du fichier sélectionné ou null si le dialogue a été annulé
- **getDirectory()** : pour obtenir le nom du dossier contenant le fichier

Exemple d'utilisation

Ecrivons une application nommée ChoixFichier qui permet de sélectionner un fichier et qui affiche son nom dans une zone de texte.

Déclarations

On utilise deux variables : zoneTexte qui pointe sur la zone de texte et fDial qui est la boîte de sélection de fichiers.

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

public class ChoixFichier extends Frame
 implements ActionListener {

 TextArea zoneTexte=new TextArea();
 FileDialog fDial;
```

Constructeur

Le constructeur initialise la boîte de sélection de fichiers, insère la zone texte, crée un menu et affiche la fenêtre.

```
 public ChoixFichier() {
  super("Sélection de fichiers");
  fDial=new FileDialog(this, "Choisir", FileDialog.LOAD);
  // Définition et affichage du système de menus
  MenuBar mb=new MenuBar();
  setMenuBar(mb);
  Menu m1=new Menu("Fichier");
  mb.add(m1);
  MenuItem mi1=new MenuItem("Sélection...");
  mi1.setActionCommand("select");
  m1.add(mi1);
  mi1.addActionListener(this);
  MenuItem mi2=new MenuItem("Quitter");
  mi2.setActionCommand("quitter");
  m1.add(mi2);
  mi2.addActionListener(this);
  //zone texte
  add(BorderLayout.CENTER,zoneTexte);
  zoneTexte.setEditable(false);
  //fermeture de la fenêtre
  addWindowListener(new Fermeture());
  //taille et position
  pack();
  setBounds(50,50,300,400);
  //affichage
  setVisible(true);
 }
```

Gestion des évènements du menu

Le menu propose deux commandes : Sélection et Quitter. C'est la commande Sélection qui provoque l'affichage de la boîte de sélection de fichiers.

```
 public void actionPerformed(ActionEvent e) {
  if (e.getActionCommand().equals("quitter")) System.exit(0);
  else if (e.getActionCommand().equals("select")) {
   fDial.setVisible(true);
   String nom=fDial.getFile();
   if (nom!=null) {
    String dir=fDial.getDirectory();
```

```
        zoneTexte.append(dir+nom+"\n");
       }
       else zoneTexte.append("Sélection annulée.\n");
      }
     }
```

Autres méthodes

Il reste à écrire la méthode main et à créer la classe interne Fermeture qui gèrera la fermeture de la fenêtre.

```
    public static void main(String args[]) {
     System.out.println("Ouverture de l'application...");
     ChoixFichier fr=new ChoixFichier();
    }

    /**
     * Gestion de l'événement Case de fermeture
     */
    class Fermeture extends WindowAdapter {
     public void windowClosing(WindowEvent event) {
      System.exit(0);
     }
    }

   }
```

Exercices

1. Modifier la classe MsgBox pour qu'elle puisse afficher des messages sur plusieurs lignes.
2. Créer une classe InputBox qui permet à l'utilisateur d'entrer une chaîne de caractètes.
3. Ecrire une application de type Bloc-Notes.