Applications fenêtrées

Table des matières

L'application Bonjour

Pour Java une application fenêtrée est une classe dérivée de la classe Frame qui représente les fenêtres. Sa méthode main devra construire une instance de la classe et l'afficher.

Création de l'application

Déclaration et constructeur

Nous créons une classe bonjour qui étend la classe Frame et qui implémente l'interface WindowListener pour réagir en particulier à l'évènement 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 évènements 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.

Méthode paint

Comme nous désirons afficher l'expression "Bonjour !", nous réécrivons la méthode paint.

```
public void paint(Graphics g) {
 int x=(getSize().width-80)/2;
 int y=(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.

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.

Applet dans une fenêtre

Insérer une applet dans une fenêtre

Les applets qui ne font pas appel à certaines méthodes spécifiques liées au document html les contenant (comme getParameter par exemple) peuvent facilement être incluses dans une application fenêtrée.

Il suffit de créer une instance de l'applet, puis d'appeler successivement ses méthodes init et start.

Utilisons l'applet Euro créée lors d'un précédent TD pour la transformer en application fenêtrée AppEuro.

Constructeur

```
public AppEuro() {
 setTitle("Conversion Euro");
 Euro a=new Euro();
 add(a);
 a.init();
 a.start();
}
```

Méthode main

```
public static void main(String args[]) {
 System.out.println("Chargement en cours...");
 AppEuro app=new AppEuro();
 app.setBounds(200,100,300,200);
 app.show();
}
```

Fermeture de la fenêtre

Il reste enfin à gérer l'évènement fermeture de la fenêtre. Cela peut se faire en utilisant l'interface WindowListener. Nous allons utiliser une autre méthode qui consiste à créer une nouvelle classe incluse dans la classe AppEuro (une innerClass) qui sera chargée de gérer cet évènement. Il s'agira d'une classe de type WindowAdapter qui implémente l'interface WindowListener; nous n'aurons ainsi qu'à redéfinir la méthode windowClosing.

Nous ajouterons donc les lignes suivantes pour définir la classe :

```
class ferme extends WindowAdapter {
 public void windowClosing(WindowEvent e) {
  dispose();
  System.exit(0);
 }
}
```

Il suffira ensuite d'indiquer dans le constructeur que les évènements fenêtre sont gérés par une instance de la classe ferme en ajoutant la ligne :

```
        addWindowListener(new ferme());
```

Remarques :

- lors de la compilation, un nouveau fichier AppEuro$ferme.class est généré
- cette méthode peut être utilisée pour gérer d'autres types d'évènements en utilisant les classes ComponentAdapter, ContainerAdapter, FocusAdapter, KeyAdapter, MouseAdapter, MouseMotionAdapter, WindowAdapter.

Créer un fichier jar

Un bon fonctionnement de l'application nécessite la présence des fichiers AppEuro.class, AppEuro$ferme et Euro.class dans le même répertoire. Il est possible de les regrouper dans un unique fichier compressé d'extension .jar en utilisant l'utilitaire jar fournit avec le JDK.

Le programme RealJ permet de réaliser cette opération de façon très simple grâce à la commande Project/Create jar... Le fichier créé contient les fichiers AppEuro.class, AppEuro$ferme.class et Euro.class, ainsi qu'un fichier "manifeste" qui permet de préciser le fichier contenant la méthode main de l'application.

Le fichier manifest.txt qui sert à créer le fichier manifeste du fichier jar contient la ligne :
```
Main-Class: AppEuro
```

Si le JDK 1.2 est installé, le fichier AppEuro.jar obtenu peut être exécuté directement par double-clic.

A partir d'une console DOS on obtient le même résultat avec la ligne de commande :
```
jar cmf manifest.txt fichier.jar *.class
```

Utiliser un menu

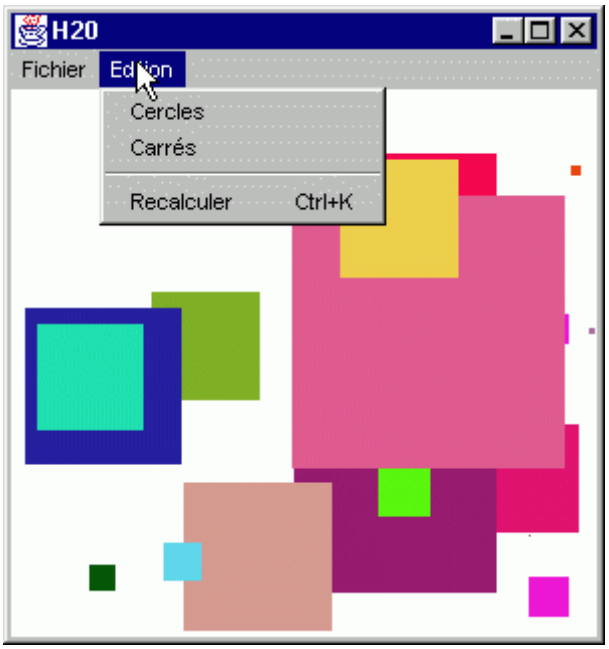
Principes de base

Pour ajouter un menu à une application on utilise 3 classes :

- la classe **MenuBar** qui représente la barre de menu et qui sera associée à l'application par l'intermédiaire de sa méthode setMenuBar.
- la classe **Menu** qui représente un menu déroulant qu'on ajoute à la barre de menu en utilisant la méthode add.
- la classe **MenuItem** qui représente une ligne d'un menu déroulant et qui déclenche des évènements de type ActionEvent.

Il faudra donc définir des écouteurs pour chaque élément de type MenuItem. Si l'application joue le rôle d'écouteur, elle devra implémenter l'interface ActionListener.

Exemple



Reprenons le composant LeDessin créé dans un précédent TD. Il dessine 20 carrés ou 20 cercles de positions, tailles et couleurs fixées au hasard. Il dispose des méthodes modeCercle et modeCarre, et il répond à la commande "Changer" en recalculant les positions, tailles et couleurs.

Nous allons créer une application qui affiche ce composant et permet de le gérer par l'intermédiaire d'un menu.

Construction du menu

Le menu sera construit dans le constructeur de l'application qui en outre contiendra un champ dessin de type LeDessin.

```
public class AppCarre extends Frame
 implements ActionListener, WindowListener {

 LeDessin dessin;

 public AppCarre() {
  MenuItem mi;
  setTitle("H20");
  addWindowListener(this);
  dessin=new LeDessin();
  add(dessin,BorderLayout.CENTER);

  //construction du menu
  MenuBar mb=new MenuBar();
  setMenuBar(mb);
  //menu Fichier
  Menu m1=new Menu("Fichier");
  mb.add(m1);
  //item Quitter
  mi=new MenuItem("Quitter");
```

```
 mi.addActionListener(this);
 m1.add(mi);
 //menu Edition
 Menu m2=new Menu("Edition");
 mb.add(m2);
 //item Cercles
 mi=new MenuItem("Cercles");
 mi.addActionListener(this);
 m2.add(mi);
 //item Carrés
 mi=new MenuItem("Carrés");
 mi.addActionListener(this);
 m2.add(mi);
 //séparateur
 m2.addSeparator();
 //item Recalculer
 mi=new MenuItem("Recalculer");
 mi.setActionCommand("Changer");
 //raccourci clavier pour cet item
 mi.setShortcut(new MenuShortcut(KeyEvent.VK_K));
 mi.addActionListener(dessin);
 m2.add(mi);
}
```

La barre de menu contient deux menus déroulants : Fichier et Edition.

Le menu Fichier contient uniquement la commande Quitter qui permet de mettre fin au programme.

Le menu Edition contient les commandes Cercles et Carrés qui permettent de choisir le type de dessin désiré, un séparateur et la commande Recalculer.

Pour chaque item du menu on définit un écouteur grâce à la méthode addActionListener. Les items Cercles et Carrés envoient leurs commandes à l'application, l'item Recalculer s'adresse au composant dessin. Comme celui-ci réagit à la commande "Changer" nous avons utilisé la méthode setActionCommand pour redéfinir la commande par défaut qui est "Recalculer".

Enfin nous avons associé un raccourci clavier (combinaison Ctrl K) à la commande Recalculer en utilisant la méthode setShortcut.

Réaction aux commandes du menu

La réaction aux commandes du menu se fait dans la méthode actionPerformed.

```
public void actionPerformed(ActionEvent e) {
 if (e.getActionCommand().equals("Cercles"))
  dessin.modeCercle();
 else if (e.getActionCommand().equals("Carrés"))
  dessin.modeCarre();
 else if (e.getActionCommand().equals("Quitter"))
  System.exit(0);
}
```

On ne fait pas référence à la commande Recalculer qui est déjà gérée par le composant dessin.

Méthode main

La méthode main construit une instance de la fenêtre de l'application et lui fixe une taille.

```
public static void main(String args[]) {
 System.out.println("Chargement en cours...");
 AppCarre fr=new AppCarre();
 fr.setBounds(50,100,300,300);
 fr.show();
}
```

La méthode setBounds permet de définir la position et la taille initiales de la fenêtre.

Pour terminer, il reste à écrire les méthodes liées à l'interface WindowListener de façon à gérer la case de fermeture de la fenêtre.

Exercices

1. Loto
   Construire un composant dérivé de Panel qui permette d'effectuer un tirage du Loto. Utiliser ce composant pour faire une applet et une application.

2. Calculette
   Créer une calculette utilisable comme applet et comme application.