JAVA

Applications Graphiques

E. ADAM

UPHF - INSA

E. ADAM - UPHF -- Java : Fenêtres en SWING

p: 1

Applications graphiques en Java

Java AWT & SWING
 Fenêtres de dialogue

Accès à l'écran

•GetDefaultToolkit() permet d'accéder aux caractéristiques de l'écran

- Exemple :
 - Dimension tailleEcran =
 java.awt.Toolkit.getDefaultToolkit().getScreenSize();
 - int hauteur = (int)tailleEcran.getHeight();
 - int largeur = (int)tailleEcran.getWidth();

•Centrer une fenêtre sur la fenêtre parente (ou l'écran) :

- Exemple :
 - maFenetre.setLocationRelativeTo(maFenetre.getParent());
 - maFrame.show()

•GraphicsEnvironment permet d'accéder à l'ensemble des environnements graphiques

• Exemple, liste des écrans disponibles :

```
GraphicsEnvironment gEnv = GraphicsEnvironment.getLocalGraphicsEnvironment();
GraphicsDevice[] devices = gEnv.getScreenDevices();
```

```
for (int i=0; i<devices.length; i++) {
    System.out.println("ecran possible : " + devices[i].getIDstring() ); }</pre>
```

```
GraphicsDevice defaultDevice = GraphicsEnvironment.
    getLocalGraphicsEnvironment().getDefaultScreenDevice();
System.out.println("Ecran principal : " + defaultDevice.getIDstring() );
```

•GraphicsEnvironment permet d'accéder à l'ensemble des environnements graphiques

```
    <u>Accéder au second écran :</u>
```

```
GraphicsEnvironment gEnv = GraphicsEnvironment.getLocalGraphicsEnvironment();
GraphicsDevice[] devices = gEnv.getScreenDevices();
```

```
// second écran = n°1
GraphicsConfiguration configGraphique = null;
if (devices.length>1) {configGraphique =
devices[1].getDefaultConfiguration();}
```

```
// lancer une JFrame sur le second écran :
JFrame frame = new JFrame(configGraphique);
```

•GraphicsEnvironment permet d'accéder à l'ensemble des environnements graphiques

- S'informer sur la mémoire de la carte graphique

GraphicsEnvironment gEnv = GraphicsEnvironment.getLocalGraphicsEnvironment();
GraphicsDevice[] devices = gEnv.getScreenDevices();

// nombre d'octets libres dans la mémoire de la carte graphique int bytes = devices[0].getAvailableAcceleratedMemory();

•GraphicsEnvironment permet d'accéder à l'ensemble des environnements graphiques

- Passer en mode plein écran

GraphicsDevice myDevice =

GraphicsEnvironment.getLocalGraphicsEnvironment().getDefaultScreenDevice();

Fenetre maFenetre;

```
if( myDevice.isFullScreenSupported() ) {
   Try { myDevice.setFullScreenWindow(myWindow);
   /** ... */
   }
   finally { myDevice.setFullScreenWindow(null); }
}
Else { System.err.println("Le 'Plein écran' n'est pas supporté");}
```

Également possibilité de changer la résolution !!

Création d'une fenêtre

contenant un texte <u>dessiné</u> et un bouton

•Taille fenêtre = 1/4 taille écran

•Fenêtre centrée

Utilisation de Java Swing

Nombreuses solutions...

🚔 Un exemple simple	
Mon Texte	
ОК	

Java Swing : disposition des éléments

Java Swing propose de nombreuses disposition (layout) permettant de choisir où ajouter des éléments à un conteneur (JPanel).

- Par défaut les éléments sont ajoutés en file (FlowLayout)
- Le BorderLayout décompose l'élément en 4 petites zones (North, East, South, West) autour d'une large zone (Center)
- Le GridLayout permet de disposer des éléments en grille

- ...

Un conteneur peut contenir des conteneur ayant des Layout différentes



Panneau sur lequel « dessiner » : la classe MonJPanel

```
class MonJPanel extends JPanel
    //lien vers la fenêtre appelante
    private JFenetre fen = null;
    public MonJPanel(JFenetre fen)
        fen = fen;
        setLayout(new BorderLayout());
    //ajout d'un bouton
        JButton bOk = new JButton("OK ...");
        add(bOk, BorderLayout.SOUTH);
    //ajout d'une gestion d'événements au bouton
        MaGestionEvent myListener = new MaGestionEvent();
        bOk.addActionListener(myListener);
```



La classe MonJPanel (suite) :

```
//fonction appelée automatiquement à chaque « repaint »
public void paintComponent(Graphics q)
        super.paintComponent(q);
        g.drawString("Mon Texte", 25, 25);
//classe interne pour l'écoute d'événements
class MaGestionEvent implements ActionListener
//fonction lancée automatiquement lors d'événements (clic, ..)
public void actionPerformed(ActionEvent event) {
fen.dispose(); }
```



La classe JFenetre :

```
//Fenêtre graphique qui va contenir le panneau
class JFenetre extends Jframe {
    public JFenetre()
        Toolkit tk = Toolkit.getDefaultToolkit();
        Dimension tailleEcran = tk.getScreenSize();
        int largeurEcran = tailleEcran.width;
        int hauteurEcran = tailleEcran.height;
        setTitle("Un exemple simple");
        setSize(largeurEcran/4, hauteurEcran/4);
        setLocationRelativeTo(null);
        MonJPanel lePanneau = new MonJPanel(this);
```

```
Container support = getContentPane();
support.add(lePanneau);
```



```
E. ADAM - UPHF -- Java : Fenêtres en SWING
```

Les importations :

```
import java.awt.*; // graphisme de base
import javax.swing.*; // graphisme évolué
import java.awt.event.*; // gestion d'événements de base
```

•« Lanceur » de fenêtre :

```
public class LanceFenetre {
   public static void main(String[] args) {
   JFenetre fen = new JFenetre();
   //en réaction à la fermeture de la fenêtre, on quitte le programme
   fen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   //ne pas oublier de rendre la fenetre lisible
    fen.setVisible(true);
```



Java Swing : un exemple simple et court

On peut condenser l'écriture pour obtenir le même résultat :

```
class MonJPanel extends JPanel
```

```
private JFenetre fen = null;
public MonJPanel(JFenetre _fen)
{
  fen = _fen;
  setLayout(new BorderLayout());
  JButton bOk = new JButton("OK ...");
  add(bOk, BorderLayout.SOUTH);
  bOk.addActionListener(e -> fen.dispose());
  }
```

Java Swing : un exemple simple et court

```
class JFenetre extends JFrame
```

```
public JFenetre()
```

```
setTitle("Un exemple simple");
setSize(400, 200);
setLocationRelativeTo(null);
```

```
MonJPanel lePanneau = new MonJPanel(this);
Container support = getContentPane();
support.add(lePanneau);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);
```

```
public static void main(String[] args) {
  JFenetre fen = new JFenetre();
```

Evénements

•Chaque composant graphique (Jcomponent) peut recevoir une écoute d'événements

- Clavier (KeyListener)
- Souris (MouseListener, MouseWheelListener, ...)
- Générique (ActionListener)
- Fenêtre (WindowListener)
- Entres autres ...

•Chacune de ces classes contient des méthodes lancées par java en fonction du type d'événements perçus

```
class MonJPanelEcouteur extends JPanel
```

```
JTextArea zoneTexte;
```

```
public MonJPanelEcouteur(JFenetreEcouteuse fen)
```

```
setLayout(new BorderLayout());
```

```
JButton b0k = new JButton("OK ..."); add(b0k, BorderLayout.SOUTH);
```

```
zoneTexte = new JTextArea();
add(zoneTexte, BorderLayout.CENTER);
zoneTexte.append("Coucou !!!!\n");
zoneTexte.setEditable(false);
//écoute événement de type action
b0k.addActionListener(e -> fen.dispose());}
```

```
l/écoute événement de type souris
zoneTexte.addMouseListener(new MouseInputAdapter() {
public void mouseEntered(MouseEvent e) { addLigne("souris sur zone de texte"); }
 public void mouseExited(MouseEvent e) { addLigne("souris sortie de la zone ");}
  });
  //écoute événement de type touche de clavier
zoneTexte.addKeyListener(new KeyAdapter() {
  Override
  public void keyTyped(KeyEvent e) {addLigne("touche captee = " + e.getKeyChar());}
});
public void addLigne(String ligne){
  if(zoneTexte!=null) zoneTexte.append(ligne+"\n");
```

```
class JFenetreEcouteuse extends JFrame
```

```
public Jfenetre() {
  setTitle("Un exemple simple");
  setSize(400, 300);
```

```
setLocationRelativeTo(null);
```

```
var lePanneau = new MonJPanelEcouteur(this);
Container support = getContentPane();
support.add(lePanneau);
```

{

}

```
addWindowListener(new WindowAdapter() {
   @Override
    public void windowClosing(WindowEvent e) { dispose();}
   Override
    public void windowActivated(WindowEvent e){
       lePanneau.addLigne("fenetre en action"); }
    Override
    public void windowDeactivated(WindowEvent e){
       lePanneau.addLigne("la fenetre perd le focus... ");
    });
    setVisible(true);
    public static void main(String[] args) {
    JFenetreEcouteuse fen = new JFenetre();
```

Evénements temporisés

- La librairie **Swing** propose un **Timer** spécifique capable d'émettre périodiquement des évènements de type **ActionEvent**
- Un Timer dispose de fonctions comme :
 - start()
 - stop()
 - restart()
 - setDelay(int top)
 - ..
- Exemple de création :
 - Timer timer = new Timer(200, e -> action())
 - Lance action() toutes les 200ms

Java Swing : Déplacement

Création d'une fenêtre contenant :

Une image passant sur une autre image

Utilisation d'un TIMER



Définition d'une zone à rafraîchir !!!

class PanneauImages extends JPanel <mark>implements ActionListener //oblige à l'écoute d'évènement</mark>

/**timer donnant le tempo pour le defilement de l'image*/

javax.swing.Timer timer;

/**image mobile, image fixe*/

BufferedImage imgMobile;

/**largeurs, hauteurs en pixel du panneau et de l'image mobile*/

int largeur, hauteur, largeurImage, hauteurImage;

/**déplacement en pixels de l'image par top*/

int pas=1;

/**durée en ms entre chaque déplacement*/

int top=5;

/**position actuelle de l'image*/

int x, y;

/**zone correspondant a l'image mobile*/

Rectangle zoneImgMobile;

PanneauImages()

```
setPreferredSize(new Dimension(largeur, hauteur));
setBackground(Color.blue);
//lecture des images
try{
 imgMobile = ImageIO.read(new File("./Lettre2.gif"));
 imgFixe = ImageIO.read(new File("./Lettre.jpg")); }
catch (Exception e){ System.out.println("probleme au moment du chargement des images"); System.exit(0); }
//recuperation des proprietes
largeur = (int)(imgFixe.getWidth()*1.5);
hauteur = (int)(imgFixe.getHeight()*1.5);
this.setPreferredSize(new Dimension(largeur, hauteur));
largeurImage = imgMobile.getWidth(); hauteurImage = imgMobile.getHeight();
x = -largeurImage - 5; y = hauteurImage / 3;
zoneImgMobile = new Rectangle(0, 0, 0, 0);
//creation du Timer et lien vers l'action a realiser
timer = new javax.swing.Timer(top, e->decaleImage());
```

```
/**decale la petite image vers la droite.<br>
```

* demande la (re)ecriture uniquement sur la zone touchee par le deplacement*/

```
public void decaleImage()
```

```
{
    //décalage en x
    x += pas;
    if (x > largeur) x = -largeurImage;
    //définition de la zone touchée par le déplacement
    Rectangle nouveauRectangle = new Rectangle(x, y, largeurImage , hauteurImage);
    Rectangle zoneRepaint = nouveauRectangle.union(zoneImgMobile);
    zoneImgMobile = nouveauRectangle;
    //ordre de réécriture localisée
    repaint(zoneRepaint.x, zoneRepaint.y, zoneRepaint.width, zoneRepaint.height);
    //tentez de remplacer la ligne ci-dessus par un simple repaint et observez
    // repaint();
```

```
/**gestion des événements sur les boutons*/
public void actionPerformed(ActionEvent e)
```

```
if (e.getActionCommand().equals("arreter")) timer.stop();
else if (e.getActionCommand().equals("reprendre")) timer.restart();
```

```
/**fonction appelee automatiquement a chaque repaint, meme local*/
@Override
public void paintComponent(Graphics g)
{
    super.paintComponent(g);
    g.drawImage(imgFixe, 0, 0, largeur, hauteur, this);
    g.drawImage(imgMobile, x, y, largeurImage, hauteurImage, this);
}
```

Java Swing : Défilement - Fenêtre

public class ImagePassante extends JFrame implements ActionListener { //la fenetre doit être à l'écoute
JButton arret = new JButton("arret");
JButton reprise = new JButton("reprendre");

```
PanneauImages ardoise;
```

```
ImagePassante() {
  ardoise = new PanneauImages();
  JPanel panneauBoutons = new JPanel();
  arret.setActionCommand("arreter");
  //un composant peut avoir plusieurs gestionnaires d'événements
  //ici le panneau (appelé ardoise)
  arret.addActionListener(ardoise);
  // et la fenêtre elle même
  arret.addActionListener(this);
  reprise.setActionCommand("reprendre");
  reprise.addActionListener(ardoise);
  reprise.addActionListener(ardoise);
  reprise.setEnabled(false);
  panneauBoutons.add(arret); panneauBoutons.add(reprise);
```

E. ADAM

Java Swing : Défilement - Fenêtre

```
//le panneau principal est formaté selon un "BorderLayout"
Container interieur = getContentPane();
interieur.setLayout(new BorderLayout());
//panneau des boutons au nord
interieur.add(panneauBoutons, BorderLayout.NORTH);
//panneau graphique au centre
interieur.add(ardoise, BorderLayout.CENTER);
interieur.setBackground(Color.white);
pack();
setVisible(true);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
ardoise.timer.start();
```

Java Swing : Défilement - Fenêtre

```
/**
 * ecoute d'événements sur les boutons.
 * ici comportement de type « bouton radio » : l'activation de l'un desactive l'autre
*/
 public void actionPerformed(ActionEvent e) {
 if (e.getActionCommand().equals("arreter")) {
   arret.setEnabled(false);
   reprise.setEnabled(true);
 if (e.getActionCommand().equals("reprendre")) {
   arret.setEnabled(true);
   reprise.setEnabled(false);
/**programme principal*/
public static void main(String[] argv) {
 ImagePassante monCadre = new ImagePassante();
```

E. ADAM

Java Swing propose des fenêtre pré-programmées : les JOptionPane Import javax.swing.JOptionPane ;

•JOptionPane.showMessageDialog(null, "Attention au message d'alerte", "alerte !", JOptionPane.ERROR_MESSAGE);

- Fenêtre affichant un simple message, avec un tag de type Erreur



Fenêtre avec choix :		
<pre>var rep = JOptionPane.showConfirmDia </pre>	alog(null,	TAR DARA NES NO ODTION)
if (rep==J0ptionPane.YES_OPTION) Sy	stem.out.println("yes	<pre>selectionne");</pre>
if (rep==J0ptionPane.NO_OPTION) Sys	<pre>tem.out.println("no s Choisissez</pre>	electionne");
	Formatter tout le disque ?	
	No Yes	

Fenêtre avec choix et boutons personnalisés:

```
Object[] options = { "OUI", "Annuler" };
var rep = JOptionPane.showOptionDialog(null, "Propager un virus à l'échelle mondiale...",
"Confirmez", JOptionPane.DEFAULT_OPTION, JOptionPane.QUESTION_MESSAGE, null, options,
options[0]);
if (rep==0) System.out.println("Oui selectionne");
if (rep==1) System.out.println("Annuler selectionne");
Confirmez
Propager un virus à l'échelle mondiale...
```

Annuler

OUL

Fenêtre avec choix en liste : :

```
Object[] choix = { "Pizza", "Steak Vegan", "Hamburger" };
```

Object repasChoisi = JOptionPane.*showInputDialog*(null, "Choisissez un menu", "Déjeuner",

```
JOptionPane.INFORMATION_MESSAGE, null, choix, choix[0]);
```



Fenêtre avec choix et zone de texte de saisie ::

String texte = JOptionPane.showInputDialog("entrez votre age : ");

votre age :	
Cancel	ОК
	votre age : Cancel

Fenêtre orientée fichier

```
JFileChooser chooser = new JFileChooser();
int returnVal = chooser.showOpenDialog(null);
                                                                                   Open
if(returnVal == JFileChooser.APPROVE_OPTION) {
                                                                                                  diner
     System.out.println("Vous avez sélectionné ce fichier : " +
                                                                                      Name
                                                                                                         ~
                                                                                                            Date Modified
                                                                                    Fourchettes.java
                                                                                                            lundi 22 octobre 2018 17:07
               chooser.getSelectedFile().getName());
                                                                                    LeDiner.iava
                                                                                                            lundi 21 octobre 2019 22:34
                                                                                    Philosophe.java
                                                                                                            lundi 22 octobre 2018 20:05
}
                                                                                             File Format: All Files
                                                                                                                     Cancel
                                                                                                                              Open
```

Fenêtre orientée couleurs

var couleur = JColorChooser.showDialog(null, "Choisissez une couleur", Color.BLUE);

Choisissez une couleur
Swatches HSV HSL RGB CMYK
Non-state Non-state
Sample Text Sample Text