

Les bases de la programmation graphique

Stephane Malandain – POO - Java

I. Premières fenêtres ...

- La classe **Jframe** du paquetage *javax.swing*
- Exemple :

```
import javax.swing.* ;  
public class fenetre1 {  
    public static void main (String args[])  
    { JFrame fen = new JFrame() ;  
      fen.setSize (300, 150) ;  
      fen.setTitle ("Ma premiere fenetre") ;  
      fen.setVisible (true) ;  
    }  
}
```

I. Premières fenêtres ...

- Exemple avec une fenêtre personnalisée :

```
import javax.swing.* ;

class MaFenetre extends JFrame
{ public MaFenetre ()    // constructeur
  { setTitle ("Ma premiere fenetre") ;
    setSize (300, 150) ;
  }
}

public class fenetre2
{ public static void main (String args[])
  { JFrame fen = new MaFenetre() ;
    fen.setVisible (true) ;
  }
}
```

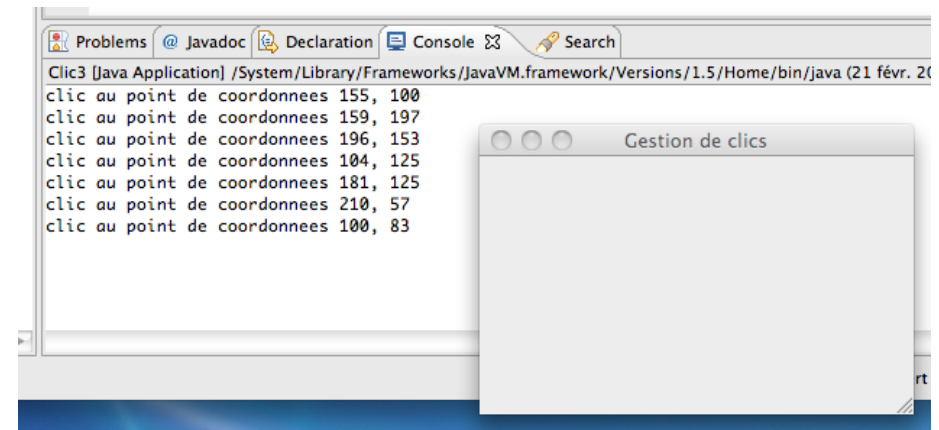
II. Gestion des événements

```
import javax.swing.* ;
import java.awt.event.* ;
class MaFenetre extends JFrame implements MouseListener
{ MaFenetre ()      // constructeur
  { setTitle ("Gestion de clics") ;
    setBounds (10, 20, 300, 200) ;
    addMouseListener (this); // la fenetre sera son propre écouteur
  }
  public void mouseClicked(MouseEvent ev)
  { int x = ev.getX() ;
    int y = ev.getY() ;
    System.out.println ("clic au point de coordonnees " + x + ", " + y ) ;
  }
  public void mousePressed (MouseEvent ev) {}
  public void mouseReleased(MouseEvent ev) {}
  public void mouseEntered (MouseEvent ev) {}
  public void mouseExited  (MouseEvent ev) {}
}
public class Clic2
{ public static void main (String args[])
  { MaFenetre fen = new MaFenetre() ;
    fen.setVisible(true) ;
  }
}
```

II. Gestion des événements

```
import javax.swing.* ;
import java.awt.event.* ;
class MaFenetre extends JFrame
{ MaFenetre ()      // constructeur
  { setTitle ("Gestion de clics") ;
    setBounds (10, 20, 300, 200) ;
    addMouseListener ( new MouseAdapter()
      { public void mouseClicked(MouseEvent ev)
        { int x = ev.getX() ;
          int y = ev.getY() ;
          System.out.println ("clic au point de coordonnees " + x + ", " + y ) ;
        }
      } ) ;
  }
}

public class Clic3
{ public static void main (String args[])
  { MaFenetre fen = new MaFenetre() ;
    fen.setVisible(true) ;
  }
}
```



II. Gestion des événements en général

- A une catégorie **Xxx**, un objet écouteur **XxxEvent** associé par une méthode **addXxxListener**.
- Si plusieurs méthodes, on les redéfinie toutes OU on peut faire appel à une classe dérivée d'une classe adaptateur **XxxAdapter** et ne fournir que les méthodes qui nous intéressent.
- Objet écouteur : Soit Objet lui-même, soit n'importe quel objet.
- Un même événement peut disposer de plusieurs écouteurs.

II. Un premier composant : Le bouton

- On crée un objet bouton avec le constructeur de la classe *Jbutton*
- On introduit le composant dans la partie contenu d'un objet de type *JFrame* avec la méthode *getContentPane*.
- La méthode *add* de la classe *Container* permet d'ajouter le composant.

```
Jbutton monBouton;  
...  
monBouton = new Jbutton ("Test"); // création d'un bouton avec  
                                   // l'étiquette "TEST"  
Container c = getContentPane();  
c.add(monBouton);  
...
```

III. Gestion du bouton avec un écouteur

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

class Fen1Bouton extends JFrame implements ActionListener
{ public Fen1Bouton ()
  { setTitle ("Premier bouton") ;
    setSize (300, 200) ;
    monBouton = new JButton ("ESSAI") ;
    getContentPane().setLayout(new FlowLayout()) ;
    getContentPane().add(monBouton) ;
    monBouton.addActionListener(this);
  }
  public void actionPerformed (ActionEvent ev)
  { System.out.println ("action sur bouton ESSAI") ;
  }
  private JButton monBouton ;
}

public class Bouton2
{ public static void main (String args[])
  { Fen1Bouton fen = new Fen1Bouton() ;
    fen.setVisible(true) ;
  }
}
```


III. Gestion de plusieurs composants

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

class Fen2Boutons extends JFrame implements ActionListener
{
    public Fen2Boutons ()
    { setTitle ("Avec deux boutons") ;
      setSize (300, 200) ;
      monBouton1 = new JButton ("Bouton A") ;
      monBouton2 = new JButton ("Bouton B") ;
      Container contenu = getContentPane() ;
      contenu.setLayout(new FlowLayout()) ;
      contenu.add(monBouton1) ;
      contenu.add(monBouton2) ;
      monBouton1.addActionListener(this);
      monBouton2.addActionListener(this);
    }
}
```

...

III. Gestion de plusieurs composants

```
...
public void actionPerformed (ActionEvent ev)
{ if (ev.getSource() == monBouton1)
    System.out.println ("action sur bouton numero 1") ;
  if (ev.getSource() == monBouton2)
    System.out.println ("action sur bouton numero 2") ;
}

private JButton monBouton1, monBouton2 ;
}

public class Boutons2
{ public static void main (String args[])
  { Fen2Boutons fen = new Fen2Boutons() ;
    fen.setVisible(true) ;
  }
}
```

III. Gestion de plusieurs composants

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

class Fen2Boutons extends JFrame
{
    public Fen2Boutons ()
    { setTitle ("Avec deux boutons") ;
      setSize (300, 200) ;
      monBouton1 = new JButton ("Bouton A") ;
      monBouton2 = new JButton ("Bouton B") ;
      Container contenu = getContentPane() ;
      contenu.setLayout(new FlowLayout()) ;
      contenu.add(monBouton1) ;
      contenu.add(monBouton2) ;
      EcouteBouton1 ecout1 = new EcouteBouton1() ;
      EcouteBouton2 ecout2 = new EcouteBouton2() ;
      monBouton1.addActionListener(ecout1) ;
      monBouton2.addActionListener(ecout2) ;
    }
    private JButton monBouton1, monBouton2
}
...
```

III. Gestion de plusieurs composants

```
...
Class EcouteBouton1 implements ActionListener
{ public void actionPerformed (ActionEvent ev)
  { System.out.println ("action sur bouton 1") ;
  }
}

Class EcouteBouton2 implements ActionListener
{ public void actionPerformed (ActionEvent ev)
  { System.out.println ("action sur bouton 2") ;
  }
}

public class Boutons3
{ public static void main (String args[])
  { Fen2Boutons fen = new Fen2Boutons() ;
    fen.setVisible(true) ;
  }
}
```

IV. Dessiner ...

```
import javax.swing.* ;
import java.awt.* ;
import java.awt.event.* ;
class MaFenetre extends JFrame
{ MaFenetre ()
    { setTitle ("Traces de clics") ;
      setSize (300, 150) ;
      pan = new Panneau() ;
      getContentPane().add(pan) ;
    }
    private Panneau pan ;
}
```

IV. Dessiner ...

```
class Paneau extends JPanel
{ final int MAX = 100 ;
  public Paneau ()
  { abs = new int[MAX] ; ord = new int[MAX] ;
    nbclics = 0 ;
    addMouseListener (new MouseAdapter()
      { public void mouseClicked (MouseEvent e)
        { if (nbclics < MAX)
          { abs[nbclics] = e.getX() ;
            ord[nbclics] = e.getY() ;
            nbclics++ ;
            repaint() ;
          }
        }
      }) ;
  }
  public void paintComponent (Graphics g)
  { super.paintComponent(g) ;
    for (int i = 0 ; i < nbclics ; i++)
      g.drawRect (abs[i], ord[i], 5, 5) ;}
  private int abs[], ord[] ;
  private int nbclics ;
}
```

IV. Dessiner ...

```
public class TrClics2
{ public static void main (String args[])
  { MaFenetre fen = new MaFenetre() ;
    fen.setVisible(true) ;
  }
}
```

IV. Dessiner à la volée.

```
import javax.swing.* ;
import java.awt.* ;
import java.awt.event.* ;
class MaFenetre extends JFrame
{ MaFenetre ()
    { setTitle ("Traces de clics") ;
      setSize (300, 150) ;
      pan = new JPanel() ;
      getContentPane().add(pan) ;
      pan.addMouseListener (new EcouteClic(pan)) ;
    }
    private JPanel pan ;
}
```


IV. Dessiner à la volée.

```
class EcouteClic extends MouseAdapter
{ public EcouteClic (JPanel pan)
  { this.pan = pan ;
  }
  public void mouseClicked (MouseEvent e)
  { int x = e.getX(), y = e.getY() ;
    Graphics g = pan.getGraphics() ;
    g.drawRect (x, y, 5, 5) ;
    g.dispose();
  }
  private JPanel pan ;
}

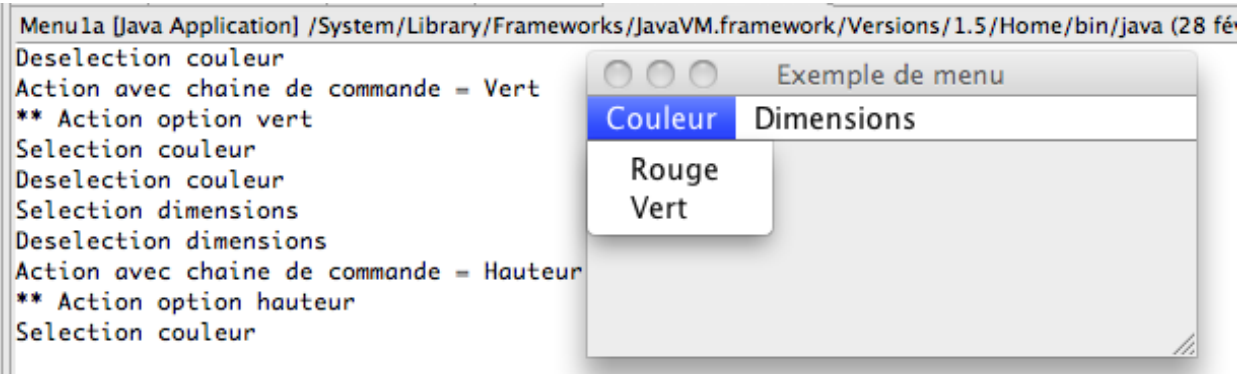
public class TrClics1
{ public static void main (String args[])
  { MaFenetre fen = new MaFenetre() ;
    fen.setVisible(true) ;
  }
}
```

V. Les menus.

```
import java.awt.*;
import java.awt.event.* ;
import javax.swing.* ;
import javax.swing.event.* ;
class FenMenu extends JFrame implements ActionListener,
    MenuListener
{ public FenMenu ()
  { setTitle ("Exemple de menu") ;
    setSize (300, 150) ;
    /* creation barre des menus */
    barreMenus = new JMenuBar() ;
    setJMenuBar(barreMenus) ;
    /* creation menu Couleur et ses options Rouge et Vert */
    couleur = new JMenu ("Couleur") ;
    barreMenus.add(couleur) ;
    couleur.addMenuListener(this) ;
    rouge = new JMenuItem ("Rouge") ;
    couleur.add(rouge) ;
    rouge.addActionListener (this) ;
```

V. Les menus.

```
vert = new JMenuItem ("Vert") ;
couleur.add(vert) ;
vert.addActionListener (this) ;
    /* creation menu Dimensions et ses options Hauteur et
    Largeur */
dimensions = new JMenu ("Dimensions") ;
barreMenus.add(dimensions) ;
dimensions.addMenuListener(this) ;
largeur = new JMenuItem ("Largeur") ;
dimensions.add(largeur) ;
largeur.addActionListener (this) ;
hauteur = new JMenuItem ("Hauteur") ;
dimensions.add(hauteur) ;
hauteur.addActionListener (this) ;
}
```



V. Les menus.

```
public void actionPerformed (ActionEvent e)
{
    Object source = e.getSource() ;
    System.out.println ("Action avec chaine de commande = "
                        + e.getActionCommand() ) ;
    if (source == rouge)    System.out.println ("** Action option
rouge") ;
    if (source == vert)    System.out.println ("** Action option
vert") ;
    if (source == largeur) System.out.println ("** Action option
largeur") ;
    if (source == hauteur) System.out.println ("** Action option
hauteur") ;
}

public void menuSelected (MenuEvent e)
{
    Object source = e.getSource() ;
    if (source == couleur) System.out.println ("Selection
couleur") ;
    if (source == dimensions) System.out.println ("Selection
dimensions") ;
}
```

V. Les menus.

```
public void menuDeselected (MenuEvent e)
{
    Object source = e.getSource() ;
    if (source == couleur) System.out.println ("Deselection
couleur") ;
    if (source == dimensions) System.out.println ("Deselection
dimensions") ;
}
public void menuCanceled (MenuEvent e)
{
    System.out.println ("Cancel ") ;
}
private JMenuBar barreMenus ;
private JMenu couleur, dimensions ;
private JMenuItem rouge, vert, largeur, hauteur ;
}
public class Menula
{
    public static void main (String args[])
    {
        FenMenu fen = new FenMenu() ;
        fen.setVisible(true) ;
    }
}
```