

Enoncé de la première partie du quatrième TD/TP de l'enseignement  
**technologies du Web**  
d'IUP GMI 2  
**(Les applets JAVA)**

Pierre Pompidor

13 avril 2001

## Objectifs fonctionnels

Création d'une **applet java** permettant la consultation des programmes de l'IUP

- L'administrateur pourra définir **l'arbre des programmes** de l'IUP (via des formulaires) par :
  - années (IUP1, IUP2, IUP3) → *premier niveau de l'arbre*
  - semestres → *second niveau de l'arbre*
  - modules d'enseignements → *troisième niveau de l'arbre*
  - chapitres dans ces modules → *feuilles de l'arbre*
- Les utilisateurs pourront :
  - visualiser l'arbre des enseignements dans une applet java
  - cliquer sur une feuille pour avoir des informations sur le contenu du chapitre

## Objectifs pédagogiques

Création d'une classe java particulière

- appelée dans une page web (cad une applet)
- réactive (l'utilisateur peut cliquer sur celle-ci)

# La gestion d'une applet

## L'appel de l'applet dans la page web

```
<applet codebase="../../applets" code="....class" width="..." height="...">
...
<param name="..." value="..."> // passage d'un paramètre à l'applet
...
```

## Importations nécessaires

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

## Définition d'une classe "Applet"

```
public class ... extends Applet
{
}
```

## Récupération des paramètres

```
public void init ()
{
    // pour un paramètre "chaîne de caractères" :
    String parametre = getParameter("nom du paramètre");
    ...

    // pour un paramètre entier :
    int parametre = Integer.parseInt (getParameter("nom du paramètre"));
    ...
}
```

## Une fonctionnalité importante : la création de panels

```
Panel panel2;
final static String SOUS_PANEL_1 = "sous panel 1";
...

// création d'un premier panel qui va contenir des boutons sélectionnant le second panel à afficher
Panel panel1 = new Panel ();
panel1.setLayout (new FlowLayout ()); // les boutons seront affichés les uns à la suite des autres
MonBouton bouton_... = new MonBouton("Sous panel 1", this, panel2);
panel1.add (bouton_...);
// même traitement pour les autres boutons
add ("North", panel1); // l'applet affiche par défaut ses panels en "BorderLayout"

// création d'un second panel qui va contenir les sous-panels sélectionnables
panel2 = new Panel (new CardLayout()); // les sous-panels sélectionnables sont empilés

// création des panels sélectionnables
Panel panel2_1 = new Panel ();
canvas_1 = new MonCanvas (...); // création du canvas supporté par ce panel
panel2_1.add ();
panel2.add (SOUS_PANEL_1, panel2_1);

// même traitement pour les autres panels sélectionnables ...

add ("Center", panel2);
```

# La gestion graphique (dans une classe dérivée d'Applet ou de Canvas)

## L'affichage graphique

```
public void paint (Graphics g)
{
    Dimension d = getSize(); // récupération de la dimension

    g.setColor(Color.black); // sélection d'une couleur

    // affichage d'une chaîne de caractères
    g.drawString ("...", abscisse, ordonnée);

    // Affichage d'un segment de droite
    g.drawLine (abscisse_pt_départ, ordonnée_pt_départ, abscisse_pt_arrivée, ordonnée_pt_arrivée);

    // Affichage d'un rectangle
    g.drawRect (abscisse, ordonnée, largeur, hauteur); // ou g.fillRect (...)

    // Affichage d'un ovale
    g.drawOval (abscisse, ordonnée, largeur, hauteur); // ou g.fillOval (...)

    // Affichage d'un segment d'arc
    g.drawArc (abscisse, ordonnée, largeur, hauteur, angleDeDépart, degrés); // ou g.fillArc (...)
}
```

## L'implémentation de MouseListener : exemple dans le constructeur d'un bouton

```
public MonBouton ("chaîne de caractères à afficher", ...) // Constructeur
{
    super(nom);
    ...
    this.addMouseListener (this); // mise en place de l'écoute d'un événement souris
}
```

## L'implémentation de MouseListener : les méthodes à redéfinir

```
class ... extends Canvas implements MouseListener
{
    public void mousePressed (MouseEvent e) {;}
    public void mouseDragged (MouseEvent e) {;}
    public void mouseReleased (MouseEvent e) {;}
    public void mouseClicked (MouseEvent e) {;}
    public void mouseEntered (MouseEvent e) {;}
    public void mouseExited (MouseEvent e) {;}
}
```

## La récupération des coordonnées d'un clic souris

```
public void mousePressed (MouseEvent e)
{
    int x, y;

    x=e.getX ();
    y=e.getY ();
}
```