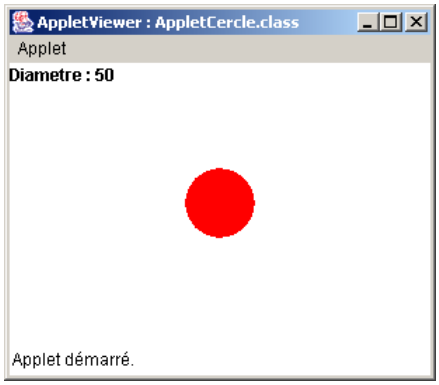
TD JAVA n°9: Les applets en Java

Exercice 1 : Dessin paramétré dans une applet



1. ↳ Ecrire une applet **`AppletCercle`** qui permet d'obtenir l'interface suivante.

On utilisera un **`PanelPrincipal`** que l'on rattachera au contenu de l'applet… Cette classe aura comme attribut : le **`diametre`** du cercle (n'oubliez pas le getteur et le setteur ! ! !)
Le panel se décomposera pour l'instant comme suit :

- Au nord : un `JLabel infoDiametre` donnant la longueur du diametre
- Ailleurs : le dessin d'un cercle plein rouge.

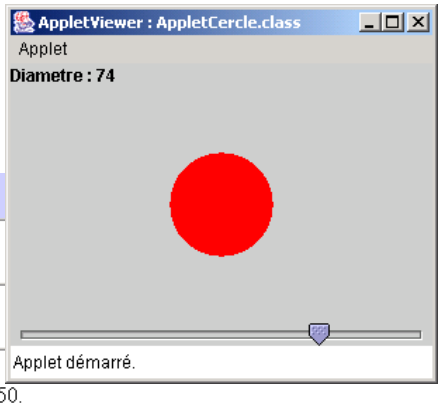
java.awt
Class Graphics

abstract void	**fillArc**(int x, int y, int width, int height, int startAngle, int arcAngle) Fills a circular or elliptical arc covering the specified rectangle.
abstract void	**fillOval**(int x, int y, int width, int height) Fills an oval bounded by the specified rectangle with the current color.
abstract void	**fillPolygon**(int[] xPoints, int[] yPoints, int nPoints) Fills a closed polygon defined by arrays of *x* and *y* coordinates.
void	**fillPolygon**(Polygon p) Fills the polygon defined by the specified Polygon object with the graphics context's current color.
abstract void	**fillRect**(int x, int y, int width, int height) Fills the specified rectangle.
abstract void	**fillRoundRect**(int x, int y, int width, int height, int arcWidth, int arcHeight) Fills the specified rounded corner rectangle with the current color.

↳ Afficher cette **applet** dans une fenêtre de largeur : 300 et hauteur: 200.

2. Modifier l'applet pour que le diamètre du cercle soit fourni par un paramètre figurant dans le fichier HTML. Si une erreur se produit lors de la récupération du paramètre, attribuer au `diametre` une valeur par défaut (par exemple : 45)

3. Proposer un réglage du diamètre grâce à un `JSlider` que vous positionnerez au bas de l'écran.
La plage des valeurs proposées par le `JSlider` ira de 0 à 100.
La valeur initiale est la valeur passée en paramètre dans le fichier HTML.



javax.swing
Class JSlider

Constructor Summary

JSlider() Creates a horizontal slider with the range 0 to 100 and an initial value of 50.
JSlider(BoundedRangeModel brm) Creates a horizontal slider using the specified BoundedRangeModel.
JSlider(int orientation) Creates a slider using the specified orientation with the range 0 to 100 and an initial value of 50.
JSlider(int min, int max) Creates a horizontal slider using the specified min and max with an initial value equal to the average of the min plus max.
JSlider(int min, int max, int value) Creates a horizontal slider using the specified min, max and value.
JSlider(int orientation, int min, int max, int value) Creates a slider with the specified orientation and the specified minimum, maximum, and initial values.

Method Summary

void	**addChangeListener**(ChangeListener l) Adds a ChangeListener to the slider.
int	**getValue**() Returns the sliders value.

javax.swing.event
Interface ChangeListener

void	**stateChanged**(ChangeEvent e) Invoked when the target of the listener has changed its state.

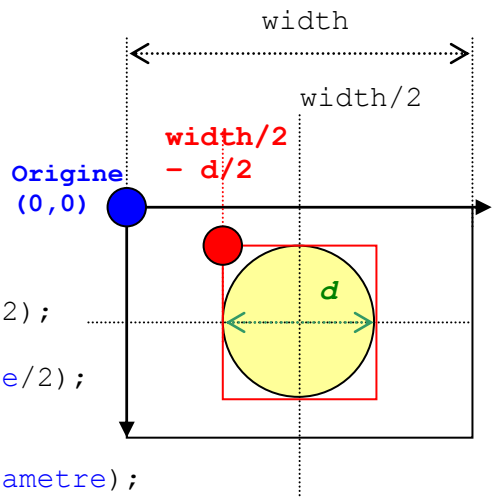
Correction TD JAVA n°9: Les applets en Java

Exercice 1 : Dessin paramétré dans une applet

1. Afin d'assurer la permanence, le dessin est réalisé dans un panneau dont on redéfinit la méthode `paintComponent`.
   - Fichier : PanelPrincipal.java :
   - Fichier : AppletCercle.java :
   - Fichier de lancement de l'applet: TestAppletCercle.html

Fichier : `PanelPrincipal.java` :

```
import javax.swing.*;
import java.awt.*; //pour les couleurs
public class PanelPrincipal extends JPanel
{
private int diametre;
private JLabel infoDiametre;

    // Constructeur permettant de donner une valeur au diametre
    // et d'organiser l'IHM
    public PanelPrincipal (int diametre)
    {
    // valeur du diametre
    this.diametre=diametre;

    // JLabel
    setLayout(new BorderLayout());
    infoDiametre = new JLabel("Diametre : " + diametre );
    add(infoDiametre,BorderLayout.NORTH);
    }

    //Getteur
    public int getDiametre ()
    { return diametre; }

    //Setteur
    public void setDiametre (int nouvDiametre)
    { this.diametre= nouvDiametre; }

    // Dessin d'un cercle ROUGE plein au milieu du panel
    protected void paintComponent(Graphics g)
    {
    super.paintComponent(g);// Ne pas l'oublier !!!
    g.setColor(Color.RED);
    int xCercleTopLeft = (this.getWidth()/2) - (diametre/2);
// x_coin_haut_gauche - rayon
    int yCercleTopLeft  = (this.getHeight()/2) - (diametre/2);
// y_coin_haut_gauche - rayon

    g.fillOval(xCercleTopLeft ,yCercleTopLeft,diametre,diametre);
    }
}
```



Remarque :

Si `xCercleTopLeft` <0 , c'est que le cercle est plus grand que le fenêtre (qui est alors toute rouge)

⇒ Solution : agrandir la fenêtre et vous verrez le cercle

↳ **Fichier : `AppletCercle.java` :**

```
import javax.swing.*;

public class AppletCercle extends JApplet
  {
   PanelPrincipal monPanel;

  public void init()
     {
     //Instanciation du panel principal
     monPanel = new PanelPrincipal(50);
     getContentPane().add(monPanel);
     }
}
```

codebase précise l'adresse relative ou absolue du répertoire contenant le fichier du code compilé de l'applet. Si ce paramètre n'est pas indiqué, le code de l'applet est recherché dans le répertoire de la page "html".

↳ **Fichier de lancement de l'applet: `TestAppletCercle.html` :**

```
<HTML>
<HEAD>
<TITLE> Dessin paramétré dans une applet </TITLE>
</HEAD>
<BODY>
<APPLET
   code  = "AppletCercle.class"
   width = "300"
   height   = "200"
   >
</APPLET>
</BODY>
</HTML>
```

2. `diametre` passé en paramètre du fichier HTML :

Quels fichiers doivent –être modifiée ? Seulement les 2 fichiers suivants :

- AppletCercle.java
- TestAppletCercle.html

PanelPrincipal reste inchangé …

↳ **Modification fichier de lancement de l'applet: `TestAppletCercle.html` :**

Dans l'objet de type **`AppletCercle`,** on récupère les valeurs des paramètres figurant dans le fichier HTML. Rappelons que ces paramètres sont identifiés par un attribut **`<PARAM …>`** à l'intérieur des balises **`<APPLET>`**……**`</APPLET>`**

- un nom : chaîne dans laquelle la casse n'est pas significative (paramètre : NAME)
- une valeur : également une chaîne (paramètre : VALUE)

```
<APPLET
   code   = "AppletCercle.class"
   width  = "300"
   height = "200"
   >
<PARAM NAME="diametre" VALUE="100">
```

```
</APPLET>
```

↳ **Modification du fichier : `AppletCercle.java` :**

On récupère la valeur du paramètre à l'aide de la méthode `getParameter` dans la méthode `init` à laquelle on fournit en argument le nom du paramètre voulue.
Attention ! ! !

Si les valeurs ne sont pas présentes dans le fichier HTML cad s 'il n'y a pas dans le fichier HTML d'attibut **<PARAM .....>** avec le nom **NAME="diametre"**
ou si elles ne sont pas convertibles en entier, nous attribuons au cercle un diametre par défaut (50 par exemple)
Remarque : en cas de besoin les dimensions de l'applet width et height peuvent également être récupérées. (pas besoin ici)

```
import javax.swing.*;

public class AppletCercle extends JApplet
  {
   PanelPrincipal monPanel;

  public void init()
     {
     // Récupération des paramètres
     // Avant car diametre sera passé en paramètre au panel Principal
     // Attention !!! getParameter renvoie un String !!!
     int diametre; //déclarer ici car utilisé ensuite hors catch !!!
     try
          {
        String chDiametre = getParameter("diametre");
        diametre = Integer.parseInt(chDiametre);
         }
   catch (NumberFormatException e) {diametre=50;}
// Exception levée si parametre diametre n'existe pas dans fichier HTML
   // ou si valeur de diametre non compatible avec un format d'entier

     // Instanciation du panel principal
     monPanel = new PanelPrincipal(diametre);
     getContentPane().add(monPanel);
     }

 }
```

Attention ! ! !

↳ Si les **valeurs ne sont pas présentes dans le fichier HTML** cad s 'il n'y a pas dans le fichier HTML d'attibut **<PARAM .....>** avec le nom **NAME="diametre"**
`chDiametre vaut null` après l'instruction : `String chDiametre = getParameter("diametre");`
⇒ alors l'instruction `int diametre = Integer.parseInt(chDiametre)` va déclencher un **`NumberFormatException`**
↳ de même si **les valeurs ne sont pas convertibles en format entier (présence de lettres par exemple)**, cad **`VALUE`** ne contient pas une chaîne « compatible » entier alors un **`NumberFormatException`** sera déclenché

*⇒ un **`try…catch(un NumberFormatException e)`***
est nécessaire pour attriber au cercle un diametre par défaut (25 par exemple)

Remarque : en cas de besoin les dimensions de l'applet width et height peuvent également être récupérées. (pas besoin ici)

3. Ajout d'un JSlider

Cette fois-ci, seul le fichier **PanelPrincipal** subira des modifications …
(ajout du JSlider pour un petit rappel sur les événements …)
⇒ Modification uniquement dans le constructeur et ajout d'une classe interne pour gérer l'événment !!!

↳ JSlider : glissière

Le **Jslider** est créé en indiquant une orientation : HORIZONTAL ou VERTICAL. On peut également préciser les valeurs minimum et maximum ainsi qu'une valeur intiale :

```
final JSlider slider = new JSlider (JSlider.HORIZONTAL,0,50,0);
```

Le JSLider déclenche des **ChangeEvent** lorsque la valeur change.

```
import javax.swing.*;
import java.awt.*; //pour les couleurs

import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;
// ou plus généralemennt : import javax.swing.event.*

import java.awt.*; //pour les couleurs

public class PanelPrincipal extends JPanel
{
int diametre;
JLabel infoDiametre;
JSlider diametreSlider;

    // Constructeur permettant de donner une valeur au diametre
    // et d'organiser l'IHM
    public PanelPrincipal (int diametre)
    {
    // valeur du diametre
    this.diametre=diametre;

    // JLabel
    setLayout(new BorderLayout());
    infoDiametre = new JLabel("Diametre : " + diametre );
    add(infoDiametre,BorderLayout.NORTH);


    // réglage des dimensions
    diametreSlider = new JSlider (JSlider.HORIZONTAL,0,100,diametre);
    add(diametreSlider,BorderLayout.SOUTH);

    // ecouteur d'événement de diamtreSlider
    diametreSlider.addChangeListener(new ChangeListener()
        {
        public void stateChanged(ChangeEvent e)
            {
            setDiametre(diametreSlider.getValue());
            infoDiametre.setText("Diametre : " + getDiametre() ); // diametre est private !
            repaint();
            }
        });


    }// fin Constructeur
    ..........
    ..........
}// fin classe PanelPrincipal
```