

Programmation Orientée Objet

Librairie Java Swing

Timer & Dessin

Mathieu RAYNAL

mathieu.raynal@irit.fr

<http://www.irit.fr/~Mathieu.Raynal>

Utilisation du Timer

Javax.swing.Timer

- Envoie un événement de type `ActionEvent` toutes les N millisecondes
 - Evènements gérés de la même manière que pour les boutons

- Constructeur

```
Timer(int N, ActionListener listener)
```

- Principales méthodes

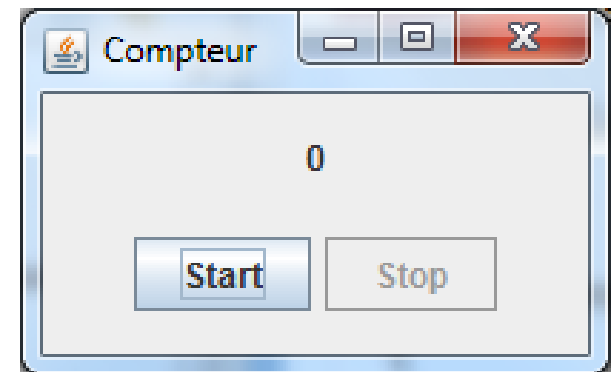
```
void setDelay(int N)
```

```
void start()
```

```
void stop()
```

Exercice : Compteur

- Réalisez le compteur suivant
 - Au démarrage seul le bouton Start est activé et le compteur est initialisé à 0
 - Après un clic sur le bouton Start
 - Le compteur est réinitialisé à 0 si ce n'est pas le cas
 - le bouton Stop est activé
 - le bouton Start est désactivé
 - Jusqu'à l'appui sur Stop, le compteur est incrémenté de 1 toutes les secondes
 - Après un clic sur le bouton Stop
 - Le compteur est arrêté
 - le bouton Stop est désactivé
 - le bouton Start est activé



Dessiner sur un composant

Dessiner en JAVA

- Où ?
 - Sur le composant Canvas en AWT
 - Sur tous les composants SWING héritant de JComponent
- Comment faire en SWING ?
 - Créer sa propre classe qui étend de **JComponent**
 - Avoir un constructeur par défaut qui appelle celui de **JComponent**
 - Pour SWING, redéfinir la méthode

```
void paintComponent(Graphics g)
```

 - Commencer cette méthode en appelant la méthode **paintComponent** de la classe **JComponent**
 - **Rafraichir un composant** `repaint()`

La classe Graphics

- Permet de dessiner sur le composant
- Dessiner les contours

```
void draw3DRect(int x, int y, int width, int height, boolean raised)
void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)
void drawLine(int x1, int y1, int x2, int y2)
void drawOval(int x, int y, int width, int height)
void drawPolygon(int[] xPoints, int[] yPoints, int nPoints)
void drawPolygon(Polygon p)
void drawPolyline(int[] xPoints, int[] yPoints, int nPoints)
void drawRect(int x, int y, int width, int height)
void drawRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight)
```

- Remplir la forme
 - Même méthode avec fill à la place de draw dans le nom de la méthode
- Changer la couleur

```
void setColor(Color couleur)
```

Squelette de classe pour créer un composant « Dessin »

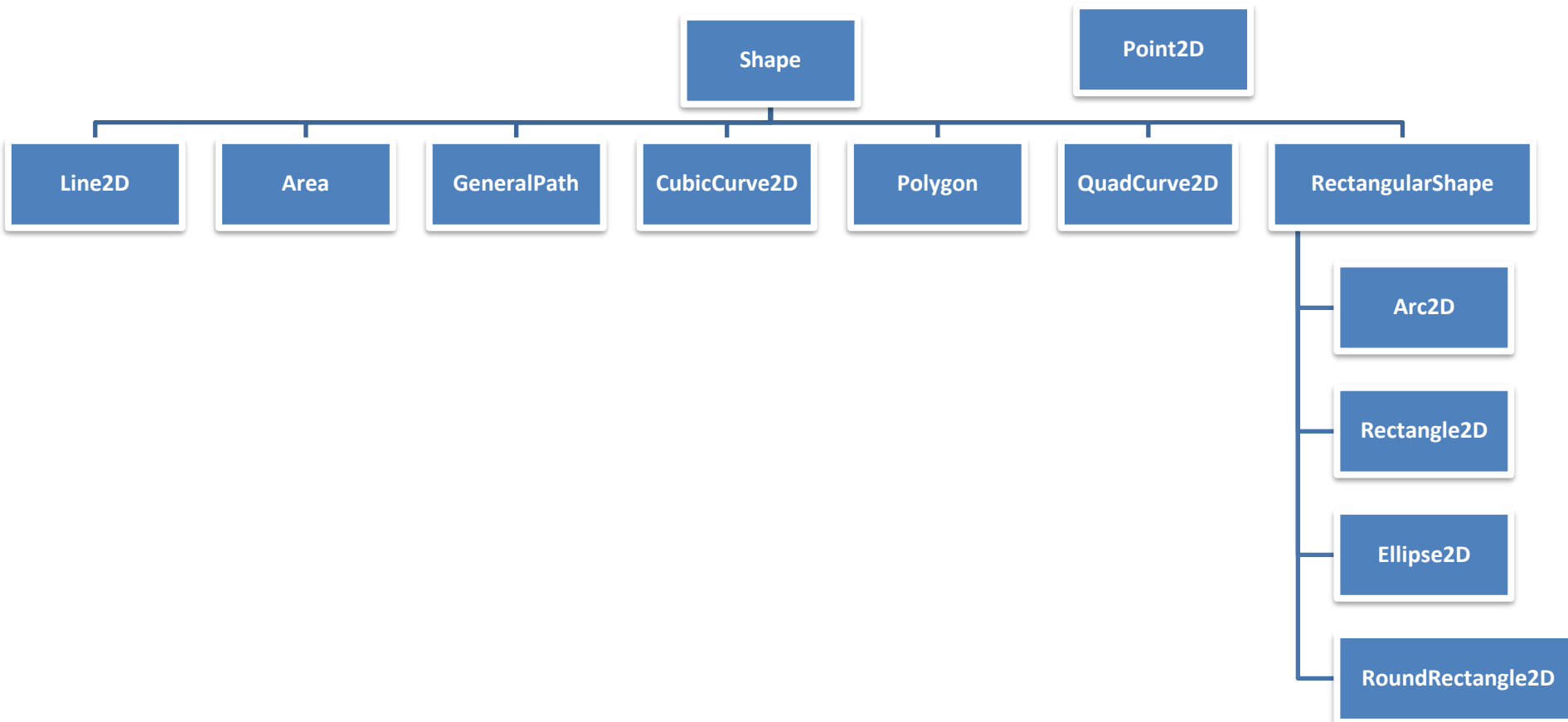
```
public class PanneauDessin extends JComponent{
    private static final int LONGUEUR = 800;
    private static final int HAUTEUR = 600;

    public PanneauDessin() {
        super();
        ...
        setPreferredSize(new Dimension(LONGUEUR,HAUTEUR));
    }

    public void paintComponent(Graphics g){
        super.paintComponent(g);
        ...
    }
}
```


La classe Graphics2D

- Affichage de Shape
 - void draw(Shape s)
 - void fill(Shape s)



Méthodes de Graphics2D

- Fonctionnalités supplémentaires avec Graphics2D

- Propriété d'affichage : RenderingHints

```
void setRenderingHint(RenderingHints.Key hintKey, Object hintValue)
```

- Style de trait : Stroke

```
void setStroke(Stroke s)
```

- Outil de remplissage : Paint

```
void setPaint(Paint paint)
```

- Transformation géométrique : AffineTransform

```
void setTransform(AffineTransform Tx)
```

Exemple

```
17 public class PanneauDessin extends JComponent{
18     int diametre;
19     Ellipse2D.Double rond;
20     RoundRectangle2D.Double rect;
21
22     public PanneauDessin()
23     {
24         super();
25         rect = new RoundRectangle2D.Double(80, 120, 100, 50, 5, 5);
26         diametre = 100;
27         rond = new Ellipse2D.Double(200-diametre/2, 200-diametre/2, diametre, diametre);
28         setPreferredSize(new Dimension(400, 400));
29     }
30
31     public void paintComponent(Graphics g)
32     {
33         super.paintComponent(g);
34         Graphics2D g2 = (Graphics2D)g;
35         g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
36             RenderingHints.VALUE_ANTIALIAS_ON);
37         g2.setColor(Color.GREEN);
38         g2.fill(rect);
39         g2.setColor(Color.BLUE);
40         g2.setStroke(new BasicStroke(2.0f));
41         g2.draw(rond);
42     }
43 }
```

