

```

2  /*
3  * To change this template, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  /**
7  *
8  * @author toshiba
9  */
10
11     import java.awt.*;
12     import java.awt.event.*;
13
14     public class RoundButton extends Component{
15
16         ActionListener actLis ;
17         String titre;
18         boolean enfonce = false ;
19         Color filling,border;
20
21         public RoundButton (){
22             this("");
23             enableEvents(AWTEvent.MOUSE_EVENT_MASK);
24         }
25
26
27         public RoundButton(String s){
28             this.titre=s;
29             enableEvents(AWTEvent.MOUSE_EVENT_MASK);
30         }
31
32         public RoundButton(Color border,Color filling){
33             this.border = border;
34             this.filling = filling;
35             this.titre="";
36             enableEvents(AWTEvent.MOUSE_EVENT_MASK);
37         }
38
39         public RoundButton(String s,Color border,Color filling){
40             this.border = border;
41             this.filling = filling;
42             this.titre=s;
43             enableEvents(AWTEvent.MOUSE_EVENT_MASK);
44         }
45
46
47         public String getLabel(){
48             return titre;
49         }
50
51         public void setLabel (String s){
52             this.titre=s;
53             invalidate();
54             repaint();
55         }
56
57         public void paint(Graphics g){
58
59             int n = Math.min(getSize().width-1, getSize().height-1);
60
61             // on dessine le cercle
62             if(enfonce){
63                 g.setColor(getColorFilling());
64                 g.fillOval(0,0,n,n);
65
66             }
67
68             g.setColor(getColorBorder());
69             g.drawOval(0,0,n,n);
70             g.drawLine(0,0,55,55);
71
72             // puis le label
73             Font f = getFont();
74             if(f!=null){
75                 FontMetrics fm = getFontMetrics(getFont());
76                 g.setColor(getForeground());
77                 g.drawString(titre, n/2-fm.stringWidth(titre)/2,n/2+fm.getMaxDescent());
78             }
79         }
80
81         public void setColor(Color b,Color f){
82             border = b;
83             filling = f;
84         }
85
86         public Color getColorBorder(){
87             return border;
88         }
89
90         public Color getColorFilling(){
91             return filling;
92         }
93
94         public Dimension getPreferredSize(){
95             Font f = getFont();
96             if (f!= null){
97                 FontMetrics fm = getFontMetrics(getFont());
98                 int max=Math.max(fm.stringWidth(titre)+40,fm.getHeight()+40);
99                 return new Dimension(max,max);
100             }
101             else return new Dimension (10,10);
102         }
103
104         public Dimension detMinimumSize(){
105             return new Dimension(10,10);
106         }
107
108
109         public void addActionListener(ActionListener lis){
110             // gestion 'manuelle' de l'ajout et suppression d'actionListener sur notre bouton
111             // voir doc de AWTEventMulticaster
112             actLis = AWTEventMulticaster.add(actLis,lis);

```



```

113     enableEvents(AWTEvent.MOUSE_EVENT_MASK);
114 }
115 public void removeActionListener(ActionListener lis){
116     actLis=AWTEventMulticaster.remove(actLis,lis);
117 }
118
119 public boolean contains(int x, int y){
120     int mx = getSize().width/2;
121     int my = getSize().height/2;
122     return ((mx-x)*(mx-x)+(my-y)*(my-y))<= mx*mx;
123 }
124
125 public void processMouseEvent(MouseEvent e){
126     Graphics g;
127     switch(e.getID()){
128     case MouseEvent.MOUSE_PRESSED :
129         enfonce = true;
130         repaint();
131         break;
132
133     case MouseEvent.MOUSE_RELEASED:
134         if(actLis != null)
135             actLis.actionPerformed(new ActionEvent(this,ActionEvent.ACTION_PERFORMED,titre));
136         if(enfonce){
137             enfonce=false;
138             repaint();
139         }
140         break;
141
142     case MouseEvent.MOUSE_ENTERED:
143         break;
144
145     case MouseEvent.MOUSE_EXITED:
146         if(enfonce){
147             enfonce=false;
148             repaint();
149         }
150         break;
151     }
152
153     super.processMouseEvent(e);
154 }
155 }
156
157

```

