

```
1 package pv_unidade_04_ex06_mouseevents;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.event.MouseEvent;
6 import java.awt.event.MouseListener;
7 import java.awt.event.MouseMotionListener;
8 import javax.swing.JFrame;
9 import javax.swing.JLabel;
10 import javax.swing.JPanel;
11
12 /**
13  * @author Edwar Saliba Júnior - http://www.esj.eti.br
14  */
15
16 public class MouseTrackerFrame extends JFrame {
17
18     private JPanel mousePanel;
19     private JLabel statusBar;
20
21     public MouseTrackerFrame() {
22         super("Demonstrando Eventos de Mouse");
23
24         mousePanel = new JPanel();
25         mousePanel.setBackground(Color.WHITE);
26         add(mousePanel, BorderLayout.CENTER);
27
28         statusBar = new JLabel("Mouse fora do JPanel.");
29         add(statusBar, BorderLayout.SOUTH);
30
31         // Cria e registra listener para mouse e eventos de movimento do mouse.
32         MouseHandler handler = new MouseHandler();
33         mousePanel.addMouseListener(handler);
34         mousePanel.addMouseMotionListener(handler);
35     }
36
37     private class MouseHandler implements MouseListener, MouseMotionListener {
38
39         @Override
40         public void mouseClicked(MouseEvent event) {
41             statusBar.setText(String.format("Clicado na coordenada [%d, %d]", event.getX(), event.getY()));
42         }
43
44         // Handler de eventos de botão do mouse pressionado.
45         @Override
46         public void mousePressed(MouseEvent event) {
47             statusBar.setText(String.format("Pressionado na coordenada [%d, %d]", event.getX(), event.getY()));
48         }
49
50         // Handler de eventos do mouse quando botão é liberado depois de pegar um objeto.
51         @Override
52         public void mouseReleased(MouseEvent event) {
53             statusBar.setText(String.format("Liberado na coordenada [%d, %d]", event.getX(), event.getY()));
54         }
55
56         // Handler de eventos quando mouse entra em determinada área (passa de um objeto para outro).
57         @Override
58         public void mouseEntered(MouseEvent event) {
59             statusBar.setText(String.format("Mouse mudou de objeto na coord. [%d, %d]", event.getX(), event.getY()));
60             mousePanel.setBackground(Color.GREEN);
61         }
62
63         // Handler de eventos quando o mouse sai de uma determinada área (sai de um objeto).
64         @Override
65         public void mouseExited(MouseEvent event) {
66             statusBar.setText("Mouse saiu do JPanel.");
67             mousePanel.setBackground(Color.WHITE);
68         }
69
70         // Handler de eventos quando o usuário arrasta o mouse com um botão pressionado.
71         @Override
72         public void mouseDragged(MouseEvent event) {
73             statusBar.setText(String.format("Arrastou até a coordenada [%d, %d]", event.getX(), event.getY()));
74         }
75
76         // Handler de eventos quando o usuário move o mouse.
77         @Override
78         public void mouseMoved(MouseEvent event) {
```

```
        statusBar.setText(String.format("Moveu até a coordenada [%d, %d]", event.getX(), event.getY()));  
    }  
}
```