

## src\zuoye\TicketApp.java

```
1 package zuoye;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7
8 public class TicketApp {
9
10     private JFrame frame;
11     private JTextArea textArea;
12     private JButton runButton, stopButton, resetButton;
13     private Ticket ticketSystem; // 票务系统
14     private Thread soldWindow1, soldWindow2; // 售票窗口线程
15     private SoldWindow soldWindowTask1, soldWindowTask2; // 售票线程任务
16
17     public TicketApp() {
18         ticketSystem = new Ticket(100); // 初始化票务系统, 100张票
19         initUI();
20     }
21
22     private void initUI() {
23         frame = new JFrame("售票系统");
24         frame.setSize(500, 400);
25         frame.setLayout(new BorderLayout());
26
27         textArea = new JTextArea(20, 30);
28         textArea.setEditable(false); // 禁止编辑
29         JScrollPane scrollPane = new JScrollPane(textArea);
30
31         runButton = new JButton("开始买票");
32         stopButton = new JButton("停止买票");
33         resetButton = new JButton("重新买票");
34
35         // 添加按钮监听器
36         runButton.addActionListener(new ButtonActionListener());
37         stopButton.addActionListener(new ButtonActionListener());
38         resetButton.addActionListener(new ButtonActionListener());
39
40         // 按钮面板
41         JPanel buttonPanel = new JPanel();
42         buttonPanel.add(runButton);
43         buttonPanel.add(stopButton);
44         buttonPanel.add(resetButton);
45
46         // 添加组件到窗口
47         frame.add(scrollPane, BorderLayout.CENTER);
48         frame.add(buttonPanel, BorderLayout.SOUTH);
49
50         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
51         frame.setVisible(true);
```

```
52     }
53
54     private class ButtonActionListener implements ActionListener {
55         @Override
56         public void actionPerformed(ActionEvent e) {
57             if (e.getSource() == runButton) {
58                 startThreads();
59             } else if (e.getSource() == stopButton) {
60                 stopThreads();
61             } else if (e.getSource() == resetButton) {
62                 resetSystem();
63             }
64         }
65     }
66
67     private void startThreads() {
68
69         soldWindowTask1 = new SoldWindow(ticketSystem, textArea, "窗口1");
70         soldWindowTask2 = new SoldWindow(ticketSystem, textArea, "窗口2");
71         soldWindow1 = new Thread(soldWindowTask1);
72         soldWindow2 = new Thread(soldWindowTask2);
73         soldWindow1.start();
74         soldWindow2.start();
75         textArea.append("售票开始! \n");
76
77     }
78
79     private void stopThreads() {
80
81         soldWindowTask1.stop();
82
83         soldWindowTask2.stop();
84         textArea.append("售票已停止。 \n");
85     }
86
87     private void resetSystem() {
88         stopThreads(); // 停止线程
89         ticketSystem.resetTickets(100); // 重置票数
90         textArea.setText("系统已重置, 票数为 100 张。 \n");
91     }
92
93     public static void main(String[] args) {
94         new TicketApp();
95     }
96 }
97
98 class Ticket {
99     private int ticketNum; // 剩余票数
100
101     public Ticket(int initialTickets) {
102         this.ticketNum = initialTickets;
103     }
104
105     public synchronized boolean sellTicket() {
```

```
106         if (ticketNum > 0) {
107             ticketNum--;
108             return true;
109         }
110         return false;
111     }
112
113     public synchronized int getRemainingTickets() {
114         return ticketNum;
115     }
116
117     public synchronized void resetTickets(int initialTickets) {
118         this.ticketNum = initialTickets;
119     }
120 }
121
122 class SoldWindow implements Runnable {
123     private final Ticket ticketSystem;
124     private final JTextArea textArea;
125     private final String windowName;
126     private volatile boolean running = true;
127
128     public SoldWindow(Ticket ticketSystem, JTextArea textArea, String windowName) {
129         this.ticketSystem = ticketSystem;
130         this.textArea = textArea;
131         this.windowName = windowName;
132     }
133
134     public void stop() {
135         running = false;
136     }
137
138     @Override
139     public void run() {
140         while (running) {
141             synchronized (ticketSystem) {
142                 if (ticketSystem.sellTicket()) {
143                     textArea.append(windowName + " 卖出一张票， 剩余票数: " +
ticketSystem.getRemainingTickets() + "\n");
144                 } else {
145                     textArea.append("票已售罄! \n");
146                     break;
147                 }
148             }
149             try {
150                 Thread.sleep(500); // 模拟售票间隔
151             } catch (InterruptedException e) {
152                 Thread.currentThread().interrupt();
153                 break;
154             }
155         }
156     }
157 }
158
```