```
1  //*****************************************************
2  //  Kathleen Kunsman
3  //  April 4, 2011
4  //  Object Oriented Programming
5  //  CHAPTER 6
6  //  RPSRevistedPanel.java
7  //*****************************************************
8
9  import javax.swing.*;
10 import java.awt.*;
11 import java.awt.event.*;
12 import java.util.Random;
13
14
15 public class RPSRevisitedPanel extends JPanel
16
17 {
18         private JPanel intro, chooseOne, output, output2;
19         private JLabel chosen, choseRock, chosePaper, choseScissors,
 … choseQuit, compChosen, playOnLabel, totalGames;
20         private JRadioButton rock, paper, scissors, quit;
21         private String choiceRock, choicePaper, choiceScissors, choiceQuit,
 … rules, playOn, compChoiceRock, userText, games, compChoicePaper,
 … compChoiceScissors, compOutput, rbs, pbr, sbp, iw, yw, wbc, tie;
22         private ImageIcon rockps, rockbs, paperbr, scissorsbp, rockImage,
 … scissorsImage, paperImage, currentImage;
23         private final int IMAGE_SIZE=100;
24         private int x,y;
25
26         enum Choice{quit, rock, paper, scissors};
27     //-----------------------------------------------------------------
28     // Set up radio buttons, panels with BoxLayouts and BorderLayouts
29     //-----------------------------------------------------------------
30     public RPSRevisitedPanel()
31     {
32
33     Choice input0, input1, input2, input3;
34     input1 = Choice.rock;
35     input2 = Choice.paper;
36     input3 = Choice.scissors;
37
38
39     rules = "Choose Rock, Paper or Scissors";
40     choiceRock="You chose Rock.";
41     choicePaper="You chose Paper.";
42     choiceScissors="You chose Scissors.";
43     choiceQuit="You chose to Quit.";
44     compOutput = "I will choose also.";
45     compChoiceRock="I chose Rock.";
46     compChoicePaper="I chose Paper.";
47     compChoiceScissors="I chose Scissors.";
48     playOn="May the best guesser win.";
49     rbs="Rock beats Scissors.";
50     pbr="Paper beats Rock.";
51     sbp="Scissors beat Paper.";
52     iw="I win!";
```

```
53      yw="You win!";
54      wbc="We both chose";
55      tie="It's a tie!";
56      userText="";
57      games="";
58
59      chosen=new JLabel(rules);
60      chosen.setFont (new Font ("Helvetica", Font.PLAIN, 14));
61      compChosen=new JLabel(compOutput);
62      compChosen.setFont (new Font ("Helvetica", Font.PLAIN, 14));
63      playOnLabel=new JLabel(playOn);
64      playOnLabel.setFont (new Font ("Helvetica", Font.PLAIN, 14));
65      totalGames=new JLabel(games);
66      totalGames.setFont (new Font ("Helvetica", Font.PLAIN, 14));
67      totalGames.setText("Let's play Rock, Paper, Scissors!");
68      totalGames.setPreferredSize (new Dimension(350,50));
69
70
71      rock = new JRadioButton ("Rock", true);
72      rock.setBackground (Color.pink);
73      paper = new JRadioButton ("Paper", true);
74      paper.setBackground (Color.pink);
75      scissors = new JRadioButton ("Scissors", true);
76      scissors.setBackground (Color.pink);
77      quit = new JRadioButton ("Quit", true);
78      quit.setBackground (Color.pink);
79
80      ButtonGroup group = new ButtonGroup();
81      group.add (rock);
82      group.add (paper);
83      group.add (scissors);
84      group.add (quit);
85
86      ChosenListener listener = new ChosenListener();
87      rock.addActionListener (listener);
88      paper.addActionListener (listener);
89      scissors.addActionListener (listener);
90      quit.addActionListener (listener);
91
92      chooseOne=new JPanel();
93      BoxLayout layout = new BoxLayout(chooseOne, BoxLayout.X_AXIS);
94      chooseOne.setLayout (layout);
95      chooseOne.setBackground (Color.pink);
96      chooseOne.setPreferredSize (new Dimension(350,75));
97      chooseOne.add (rock);
98      chooseOne.add (paper);
99      chooseOne.add (scissors);
100     chooseOne.add (quit);
101
102     intro=new JPanel();
103     BoxLayout layout1 = new BoxLayout(intro, BoxLayout.Y_AXIS);
104     intro.setLayout (layout1);
105     intro.add (Box.createRigidArea(new Dimension(0,10)));
106     intro.add (chosen);
107     intro.add (Box.createRigidArea(new Dimension(0,10)));
108     intro.add (compChosen);
```

```
109     intro.add (Box.createRigidArea(new Dimension(0,10)));
110     intro.add (playOnLabel);
111     intro.add (Box.createRigidArea(new Dimension(0,10)));
112     intro.setBackground (Color.white);
113     intro.setPreferredSize (new Dimension(220,100));
114
115     output=new JPanel();
116     BoxLayout layout2 = new BoxLayout(output, BoxLayout.Y_AXIS);
117     output.setLayout (layout2);
118     output.setBackground (Color.white);
119     output.setPreferredSize (new Dimension(20,100));
120
121     output2=new JPanel();
122     BoxLayout layout3 = new BoxLayout(output2, BoxLayout.Y_AXIS);
123     output2.setLayout (layout3);
124     output2.setBackground (Color.white);
125     output2.setPreferredSize (new Dimension(110,100));
126     setLayout(new BorderLayout());
127     setBackground(Color.pink);
128
129
130     add(totalGames,BorderLayout.NORTH);
131     add(intro, BorderLayout.CENTER);
132     add(output, BorderLayout.WEST);
133     add(output2,BorderLayout.EAST);
134     add(chooseOne,BorderLayout.SOUTH);
135
136
137
138     }
139
140
141     //-----------------------------------------------------------------
142     // Represents the listener for radio buttons
143     //-----------------------------------------------------------------
144     private class ChosenListener implements ActionListener
145     {
146
147         int userChoice, ties, userWins, compWins;
148
149         //-----------------------------------------------------------------
150         // Displays text representing user radio picks and computer random
  … picks
151         //-----------------------------------------------------------------
152         public void actionPerformed (ActionEvent event)
153         {
154             Object source = event.getSource();
155             Random pick = new Random();
156
157             int compChoice = pick.nextInt(3)+1;
158
159             if (source==rock){
160                 userChoice = 1;
161                 userText=Choice.rock.name();
162                 chosen.setText(choiceRock);}
163                 else
```

```
164                 if (source == paper){
165                 userChoice = 2;
166                 userText=Choice.paper.name();
167                 chosen.setText(choicePaper);}
168                 else
169                 if (source == scissors){
170                 userChoice = 3;
171                 userText=Choice.scissors.name();
172                 chosen.setText(choiceScissors);}
173                 else{
174                 chosen.setText(choiceQuit);
175                 userChoice = 4;}
176
177             if (compChoice==1)
178                 compChosen.setText(compChoiceRock);
179                 else
180                 if (compChoice==2)
181                 compChosen.setText(compChoicePaper);
182                 else
183                 if (compChoice==3)
184                 compChosen.setText(compChoiceScissors);
185
186
187             if (userChoice == compChoice){
188                 ties++;
189                 playOnLabel.setText(wbc+" "+userText+ ". "+tie);
190                 totalGames.setText("");
191
192                 }else
193 {
194             switch(userChoice){
195
196             case 1:
197             {
198             if (compChoice ==2){
199             playOnLabel.setText(pbr +" "+ iw);
200             compWins++;
201             totalGames.setText("");
202             }else{
203             playOnLabel.setText(rbs +" "+ yw);
204             userWins++;
205             totalGames.setText("");
206
207             }
208             break;
209             }
210
211             case 2:
212             {
213             if (compChoice ==1){
214             playOnLabel.setText(pbr +" "+ yw);
215             userWins++;
216             totalGames.setText("");
217
218             }else{
219             playOnLabel.setText(sbp +" "+ iw);
```

```
220             compWins++;
221             totalGames.setText("");
222 
223             }
224             break;
225             }
226 
227             case 3:
228             {
229             if (compChoice ==1){
230             playOnLabel.setText(rbs +" "+ iw);
231             compWins++;
232             totalGames.setText("");
233 
234             }else{
235             playOnLabel.setText(sbp +" "+ yw);
236             userWins++;
237             totalGames.setText("");
238 
239             }
240             break;
241             }
242 
243             default:{
244         }
245 
246         if (source==quit){
247         int gamesPlayed = userWins + compWins + ties;
248         chosen.setText("Player wins: " + userWins);
249         compChosen.setText("Computer wins: "+ compWins);
250         playOnLabel.setText("Ties: "+ties);
251         totalGames.setText("Games played: " + gamesPlayed + "  Let's play
  … again!");
252                     }
253                     //---------------------------------------
254                     // Resets counts for new game
255                     //---------------------------------------
256 
257                     userWins=0;
258                     compWins=0;
259                     ties=0;
260 
261                             }
262 
263                             }}}}
264 
265 
266 
267 
```