

```

1 //Robert Swartz
2 //Move Counter
3 //Copyright 2020
4 //Spacecraft Design
5
6
7 import java.awt.*;
8 import java.awt.event.*;
9
10
11 public class MoveCounter
12 {
13     public static void main(String[] args)
14     {
15         AppletImage cubs = new AppletImage();
16         int w = 600, h = 500, h2 = 75;
17         cubs.powArray[0] = 1;
18         for (int i = 1; i < 63; i++)
19             cubs.powArray[i] = cubs.powArray[i-1]*2+1;
20         cubs.pF.setLocation(125, 125);
21         cubs.pF.setVisible(true);
22         cubs.pF.setSize(w, h);
23         cubs.pF.setResizable(false);
24         cubs.mwa = new MyWindowAdapter(cubs);
25         cubs.pF.addWindowListener(cubs.mwa);
26         cubs.p0.setSize(w, h);
27         cubs.pF.add(cubs.p0);
28         cubs.start = new Button("Start");
29         cubs.stop = new Button("Stop");
30         cubs.mka1 = new MyKeyAdapter1(cubs);
31         cubs.mma1 = new MyMouseAdapter1(cubs);
32         cubs.mka2 = new MyKeyAdapter2(cubs);
33         cubs.mma2 = new MyMouseAdapter2(cubs);
34         cubs.start.setBackground(Color.green);
35         cubs.start.setForeground(new Color(0, 0, 208));
36         cubs.start.setFont(new Font("SansSerif", Font.BOLD, 13));
37         cubs.start.addKeyListener(cubs.mka1);
38         cubs.start.addMouseListener(cubs.mma1);
39         cubs.stop.setBackground(Color.red);
40         cubs.stop.setForeground(new Color(0, 0, 208));
41         cubs.stop.setFont(new Font("SansSerif", Font.BOLD, 13));
42         cubs.stop.addKeyListener(cubs.mka2);
43         cubs.stop.addMouseListener(cubs.mma2);
44         cubs.p1.setSize(w, h2);
45         cubs.p1.setBackground(Color.lightGray);
46         cubs.p1.add(new Label());
47         cubs.p1.add(new Label());
48         String[] instructions = {"Min # of Discs", "Max # of Discs", "# of Pegs"},
49             defaults = {"1", "2000", "4"};
50         for (int i = 0; i < 3; i++)
51         {
52             cubs.lbls[i] = new Label(instructions[i], Label.RIGHT);
53             cubs.lbls[i].setFont(new Font("SansSerif", Font.PLAIN, 13));
54             cubs.entries[i] = new TextField(defaults[i]);
55             cubs.entries[i].setFont(new Font("SansSerif", Font.PLAIN, 13));
56             cubs.p1.add(cubs.lbls[i]);
57             cubs.p1.add(cubs.entries[i]);
58             for (int j = 0; j < 4; j++)
59                 cubs.p1.add(new Label());
60         }

```

```

61     cubs.p1.add(cubs.start);
62     cubs.p1.add(cubs.stop);
63     cubs.lbls[3] = new Label("", Label.CENTER);
64     cubs.lbls[3].setForeground(new Color(0, 0, 208));
65     cubs.p1.add(cubs.lbls[3]);
66     cubs.p1.add(new Label());
67     cubs.out.setEditable(false);
68     cubs.out.setSize(w, h-h2);
69     cubs.out.setFont(new Font("SansSerif", Font.PLAIN, 15));
70     cubs.out.setBackground(new Color(112, 224, 192));
71     cubs.p0.add(cubs.out, "Center");
72     cubs.p0.add(cubs.p1, "South");
73     cubs.pF.paintAll(cubs.pF.getGraphics());
74     cubs.entries[0].requestFocus();
75     System.gc();
76 }
77 }
78
79
80 class AppletImage
81 {
82     public MoveCounter2 cubs2;
83     public Frame pF = new Frame("Move Counter");
84     public Panel p0 = new Panel(new BorderLayout()), p1 = new Panel(new
85         GridLayout(4, 6));
86     public TextArea out = new TextArea("Robert Swartz\nwww.mathapplets.net\n"+
87         "Move Counter\nCopyright 2020", 10, 10, TextArea.SCROLLBARS_VERTICAL_ONLY)
88         ;
89     public Button start, stop;
90     public Label lbls[] = new Label[4];
91     public TextField[] entries = new TextField[3];
92     public MyKeyAdapter1 mka1;
93     public MyMouseAdapter1 mma1;
94     public MyKeyAdapter2 mka2;
95     public MyMouseAdapter2 mma2;
96     public MyWindowAdapter mwa;
97     public boolean running;
98     public long[] powArray = new long[63];
99     public boolean[][] doneYet = new boolean[2000][1000];
100    public long[][] moves = new long[2000][1000];
101 }
102
103
104 class MoveCounter2 extends Thread
105 {
106     public AppletImage cubs;
107     public int minDiscs, maxDiscs, pegs;
108     public MyException myE = new MyException();
109
110     public MoveCounter2(AppletImage cubs0, int min, int max, int pegs0)
111     {
112         cubs = cubs0;
113         minDiscs = min;
114         maxDiscs = max;
115         pegs = pegs0;
116     }
117
118     public void run()
119     {
120         cubs.running = true;

```



```

181         if (count1 < 1)
182             break $towers;
183         cubs.moves[n-1][p-1] = count1;
184         cubs.doneYet[n-1][p-1] = true;
185     }
186 }
187 }
188
189
190 class MyException extends Exception{}
191
192
193 class MyKeyAdapter1 extends KeyAdapter
194 {
195     public AppletImage cubs;
196
197     public MyKeyAdapter1(AppletImage cubs0)
198     {
199         cubs = cubs0;
200     }
201
202     public void keyReleased(KeyEvent e)
203     {
204         int minDiscs = 0, maxDiscs = 0, pegs = 0;
205         boolean error = false;
206         if ((e.getKeyCode() == KeyEvent.VK_ENTER || e.getKeyCode() ==
207             KeyEvent.VK_SPACE) && !cubs.running)
208         {
209             try
210             {
211                 minDiscs = Integer.parseInt(cubs.entries[0].getText());
212                 maxDiscs = Integer.parseInt(cubs.entries[1].getText());
213                 pegs = Integer.parseInt(cubs.entries[2].getText());
214             }
215             catch (NumberFormatException e2)
216             {
217                 error = true;
218             }
219             if (error || minDiscs < 1 || maxDiscs < 1 || maxDiscs < minDiscs ||
220                 pegs < 4 || minDiscs > 2000 || maxDiscs > 2000 || pegs > 1000)
221             {
222                 cubs.lbls[3].setFont(new Font("Times", Font.BOLD, 12));
223                 cubs.lbls[3].setText("INVALID ENTRY!!!");
224             }
225             else
226             {
227                 cubs.cubs2 = new MoveCounter2(cubs, minDiscs, maxDiscs, pegs);
228                 cubs.cubs2.start();
229             }
230         }
231     }
232 }
233
234
235 class MyMouseAdapter1 extends MouseAdapter
236 {
237     public AppletImage cubs;
238
239     public MyMouseAdapter1(AppletImage cubs0)
240     {

```

```

241     cubs = cubs0;
242 }
243
244 public void mousePressed(MouseEvent e)
245 {
246     int minDiscs = 0, maxDiscs = 0, pegs = 0;
247     boolean error = false;
248     if (e.getButton() == MouseEvent.BUTTON1 && !cubs.running)
249     {
250         try
251         {
252             minDiscs = Integer.parseInt(cubs.entries[0].getText());
253             maxDiscs = Integer.parseInt(cubs.entries[1].getText());
254             pegs = Integer.parseInt(cubs.entries[2].getText());
255         }
256         catch (NumberFormatException e2)
257         {
258             error = true;
259         }
260         if (error || minDiscs < 1 || maxDiscs < 1 || maxDiscs < minDiscs ||
261             pegs < 4 || minDiscs > 2000 || maxDiscs > 2000 || pegs > 1000)
262         {
263             cubs.lbls[3].setFont(new Font("Times", Font.BOLD, 12));
264             cubs.lbls[3].setText("INVALID ENTRY!!!");
265         }
266         else
267         {
268             cubs.cubs2 = new MoveCounter2(cubs, minDiscs, maxDiscs, pegs);
269             cubs.cubs2.start();
270         }
271     }
272 }
273 }
274
275
276 class MyKeyAdapter2 extends KeyAdapter
277 {
278     public AppletImage cubs;
279
280     public MyKeyAdapter2(AppletImage cubs0)
281     {
282         cubs = cubs0;
283     }
284
285     public void keyPressed(KeyEvent e)
286     {
287         if (e.getKeyCode() == KeyEvent.VK_ENTER || e.getKeyCode() ==
288             KeyEvent.VK_SPACE)
289         {
290             cubs.running = false;
291             cubs.lbls[3].setText("");
292         }
293     }
294 }
295
296
297 class MyMouseAdapter2 extends MouseAdapter
298 {
299     public AppletImage cubs;
300

```

```
301     public MyMouseAdapter2(AppletImage cubs0)
302     {
303         cubs = cubs0;
304     }
305
306     public void mousePressed(MouseEvent e)
307     {
308         if (e.getButton() == MouseEvent.BUTTON1)
309         {
310             cubs.running = false;
311             cubs.lbls[3].setText("");
312         }
313     }
314 }
315
316
317 class MyWindowAdapter extends WindowAdapter
318 {
319     public AppletImage cubs;
320
321     public MyWindowAdapter(AppletImage cubs0)
322     {
323         cubs = cubs0;
324     }
325
326     public void windowClosing(WindowEvent e)
327     {
328         cubs.running = false;
329         cubs.pF.dispose();
330     }
331 }
```