

```

1  import java.awt.*;
2  import javax.swing.*;
3  import java.awt.event.*;
4  import java.util.*;
5  import java.text.*;
6
7  /*****
8   * GUI for a book store.
9   *
10  * @author Shane Stacy
11  * @version 12/6/2017
12  *****/
13  public class MarketPlaceGUI extends JFrame implements ActionListener{
14
15      MarketPlace store;
16
17      JButton simulate;
18
19      JTextField cashiersTextField, avgArrivalTextField, avgServiceTextField;
20
21      JCheckBox displayCheckBox;
22
23      /** Results text area */
24      JTextArea resultsArea;
25
26      /** menu items */
27      JMenuBar menus;
28      JMenu fileMenu;
29      JMenuItem quitItem;
30      JMenuItem clearItem;
31
32      /*****
33       * Main Method
34       *****/
35      public static void main(String args[]){
36          MarketPlaceGUI gui = new MarketPlaceGUI();
37          gui.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
38          gui.setTitle("MarketPlace");
39          gui.pack();
40          gui.setVisible(true);
41      }
42
43      /*****
44       * constructor installs all of the GUI components
45       *****/
46      public MarketPlaceGUI(){
47
48          store = new MarketPlace();
49
50          // set the layout to GridBag
51          setLayout(new GridBagLayout());
52          GridBagConstraints loc = new GridBagConstraints();
53
54          // create results area to span one column and 10 rows
55          resultsArea = new JTextArea(20,20);
56          JScrollPane scrollPane = new JScrollPane(resultsArea);

```

```

57     loc.gridx = 0;
58     loc.gridy = 1;
59     loc.gridheight = 10;
60     loc.insets.left = 20;
61     loc.insets.right = 20;
62     loc.insets.bottom = 20;
63     add(scrollPane, loc);
64
65     // create Results label
66     loc = new GridBagConstraints();
67     loc.gridx = 0;
68     loc.gridy = 0;
69     loc.insets.bottom = 20;
70     loc.insets.top = 20;
71     add(new JLabel("Results"), loc);
72
73     // create Searches label
74     loc = new GridBagConstraints();
75     loc.gridx = 1;
76     loc.gridy = 0;
77     loc.gridwidth = 2;
78     add(new JLabel("Parameters"), loc);
79
80     loc = new GridBagConstraints();
81     loc.gridx = 1;
82     loc.gridy = 1;
83     //loc.gridwidth = 2;
84     add(new JLabel("Cashiers"), loc);
85
86     loc = new GridBagConstraints();
87     loc.gridx = 1;
88     loc.gridy = 3;
89     //loc.gridwidth = 2;
90     add(new JLabel("Avg Arrival Time"), loc);
91
92     loc = new GridBagConstraints();
93     loc.gridx = 1;
94     loc.gridy = 5;
95     //loc.gridwidth = 2;
96     add(new JLabel("Avg Service Time"), loc);
97
98     loc = new GridBagConstraints();
99     loc.gridx = 1;
100    loc.gridy = 7;
101    //loc.gridwidth = 2;
102    add(new JLabel("Display"), loc);
103
104    cashiersTextField = new JTextField("3", 5);
105    loc.gridx = 2;
106    loc.gridy = 1;
107    loc.gridwidth = 2;
108    add(cashiersTextField, loc);
109
110    avgArrivalTextField = new JTextField("2.5", 5);
111    loc.gridx = 2;
112    loc.gridy = 3;

```

```

113         loc.gridwidth = 2;
114         add(avgArrivalTextField, loc);
115
116         avgServiceTextField = new JTextField("6.6", 5);
117         loc.gridx = 2;
118         loc.gridy = 5;
119         loc.gridwidth = 2;
120         add(avgServiceTextField, loc);
121
122         displayCheckBox = new JCheckBox();
123         loc.gridx = 2;
124         loc.gridy = 7;
125         loc.gridwidth = 2;
126         add(displayCheckBox, loc);
127
128         simulate = new JButton("Simulate");
129         loc.gridx = 2;
130         loc.gridy = 9;
131         loc.gridwidth = 2;
132         add(simulate, loc);
133
134         // hide details of creating menus
135         setupMenus();
136
137     }
138
139     /*****
140     * This method is called when any button is clicked. The proper
141     * internal method is called as needed.
142     *
143     * @param e the event that was fired
144     *****/
145     public void actionPerformed(ActionEvent e){
146
147         // extract the button that was clicked
148         JComponent buttonPressed = (JComponent) e.getSource();
149
150         // Allow user to load baby names from a file
151         int num = 0;
152         double s = 0;
153         double a = 0;
154         boolean ck = true;
155         if (buttonPressed == simulate) {
156             MarketPlace bookstore = new MarketPlace();
157             if (cashiersTextField.getText().length() == 0) {
158                 JOptionPane.showMessageDialog(this, "Provide the number of cashiers."); // check
159             }
160             else if (Integer.parseInt(cashiersTextField.getText()) < 0) {
161                 JOptionPane.showMessageDialog(this, "Negative number of cashiers not allowed.");
162             }
163             else {
164                 num = Integer.parseInt(cashiersTextField.getText());
165             }
166             if (avgServiceTextField.getText().length() == 0) {

```

```

167         JOptionPane.showMessageDialog(this, "Provide the average service time in
minutes."); // check for no answer
168     }
169     else if (Double.parseDouble(avgServiceTextField.getText()) < 0) {
170         JOptionPane.showMessageDialog(this, "Negative number minutes for service time is
not allowed."); // check for negative answer
171     }
172     else {
173         s = Double.parseDouble(avgServiceTextField.getText());
174     }
175     if (avgArrivalTextField.getText().length() == 0) {
176         JOptionPane.showMessageDialog(this, "Provide the average arrival time in
minutes."); // check for no answer
177     }
178     else if (Double.parseDouble(avgArrivalTextField.getText()) < 0) {
179         JOptionPane.showMessageDialog(this, "Negative number minutes for arrival time is
not allowed."); // check for a negative answer
180     }
181     else {
182     }
183     else {
184
185         a = Double.parseDouble(avgArrivalTextField.getText());
186     }
187
188     // if checkbox is checked display the checkoutarea
189     if (displayCheckBox.isSelected()) {
190         ck = true;
191     }
192     else {
193         ck = false;
194     }
195     bookstore.setParameters(num, s, a, ck);
196     bookstore.startSimulation();
197     resultsArea.append(bookstore.results);
198
199 }
200
201 }
202
203 //*****
204 Creates the menu items
205 //*****
206 private void setupMenus(){
207     fileMenu = new JMenu("File");
208     quitItem = new JMenuItem("Quit");
209     clearItem = new JMenuItem("Clear");
210     fileMenu.add(clearItem);
211     fileMenu.add(quitItem);
212     menus = new JMenuBar();
213     setJMenuBar(menus);
214     menus.add(fileMenu);
215
216     quitItem.addActionListener(this);
217     clearItem.addActionListener(this);
218     simulate.addActionListener(this);
219     displayCheckBox.addActionListener(this);

```

219  
220       }  
221  
222 }