

```

1 package Project4;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7 import java.util.GregorianCalendar;
8
9 /*****
10  *
11  * Maintains the GUI1024 for the red box rental store
12  *
13  *****/
14 public class GUICarDealer extends JFrame implements ActionListener {
15     /**
16      * Holds menu bar
17      */
18     private JMenuBar menus;
19
20     /**
21      * menus in the menu bar
22      */
23     private JMenu fileMenu;
24     private JMenu actionMenu;
25
26     /**
27      * menu items in each of the menus
28      */
29     private JMenuItem openSerItem;
30     private JMenuItem exitItem;
31     private JMenuItem saveSerItem;
32     private JMenuItem boughtCarItem;
33     private JMenuItem boughtTruckItem;
34     private JMenuItem Undo;
35
36     private JMenuItem soldItem;
37
38     /**
39      * Holds the list engine
40      */
41     private ListEngine DList;
42     private JPanel panel;
43
44     /**
45      * Holds JListArea
46      */
47     private JList jListArea;
48
49     /** Scroll pane */
50     //private JScrollPane scrollList;
51
52     /*****
53      *
54      * A constructor that starts a new GUI1024 for the rental store
55      *
56      *****/

```

```

57     public GUICarDealer() {
58         //adding menu bar and menu items
59         menus = new JMenuBar();
60         fileMenu = new JMenu("File");
61         actionMenu = new JMenu("Action");
62         openSerItem = new JMenuItem("Open File");
63         exitItem = new JMenuItem("Exit");
64         saveSerItem = new JMenuItem("Save File");
65         boughtCarItem = new JMenuItem("Bought a Car");
66         boughtTruckItem = new JMenuItem("Bought a Truck");
67         Undo = new JMenuItem("Undo");
68
69         soldItem = new JMenuItem("Sold Car or Truck");
70
71         //adding items to bar
72         fileMenu.add(openSerItem);
73         fileMenu.add(saveSerItem);
74         fileMenu.add(exitItem);
75         actionMenu.add(boughtCarItem);
76         actionMenu.add(boughtTruckItem);
77
78         actionMenu.add(soldItem);
79         actionMenu.add(Undo);
80
81         menus.add(fileMenu);
82         menus.add(actionMenu);
83
84         //adding ActionListener
85         openSerItem.addActionListener(this);
86         saveSerItem.addActionListener(this);
87         exitItem.addActionListener(this);
88         boughtCarItem.addActionListener(this);
89         boughtTruckItem.addActionListener(this);
90         soldItem.addActionListener(this);
91         Undo.addActionListener(this);
92
93         setJMenuBar(menus);
94         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
95
96         panel = new JPanel();
97         DList = new ListEngine();
98         jListArea = new JList(DList);
99         JScrollPane scrollList = new JScrollPane(jListArea);
100        scrollList.setPreferredSize(new Dimension(800, 300));
101        panel.add(scrollList);
102
103        add(panel, BorderLayout.CENTER);
104
105        setVisible(true);
106        setSize(950, 450);
107    }
108
109    /*****
110     *
111     * This method handles event-handling code for the GUI1024
112     *

```

```

113     * @param e - Holds the action event parameter
114     *****/
115     public void actionPerformed(ActionEvent e) {
116
117         Object comp = e.getSource();
118
119         if (openSerItem == comp) {
120             JFileChooser chooser = new JFileChooser();
121             int status = chooser.showOpenDialog(null);
122             if (status == JFileChooser.APPROVE_OPTION) {
123                 String filename = chooser.getSelectedFile().getAbsolutePath();
124                 if (openSerItem == comp)
125                     DList.loadDatabase(filename);
126             }
127         }
128
129         if (saveSerItem == comp) {
130             JFileChooser chooser = new JFileChooser();
131             int status = chooser.showSaveDialog(null);
132             if (status == JFileChooser.APPROVE_OPTION) {
133                 String filename = chooser.getSelectedFile().getAbsolutePath();
134                 if (saveSerItem == e.getSource())
135                     DList.saveDatabase(filename);
136             }
137         }
138
139         if (e.getSource() == Undo){
140             DList.undo();
141         }
142         //MenuBar options
143         if (e.getSource() == exitItem) {
144             System.exit(1);
145         }
146
147         if (e.getSource() == boughtCarItem) {
148             Car car = new Car();
149             BoughtCarDialog dialog = new BoughtCarDialog(this, car);
150             if (dialog.getCloseStatus() == BoughtCarDialog.OK) {
151                 DList.add(car);
152                 DList.addTo2ndLinked(car, 1);
153             }
154         }
155
156         if (e.getSource() == boughtTruckItem) {
157             Truck truck = new Truck();
158             BoughtTruckDialog dialog = new BoughtTruckDialog(this, truck);
159             if (dialog.getCloseStatus() == BoughtTruckDialog.OK) {
160                 DList.add(truck);
161                 DList.addTo2ndLinked(truck, 1);
162             }
163         }
164
165         if (soldItem == e.getSource()) {
166             int index = jListArea.getSelectedIndex();
167             if (index != -1) {
168                 GregorianCalendar dat = new GregorianCalendar();

```

```
169
170         Auto unit = DList.get(index);
171         SoldOnDialog dialog = new SoldOnDialog(this, unit);
172         if (dialog.getCloseStatus() == SoldOnDialog.OK) {
173             unit = DList.remove(index);
174             DList.addTo2ndLinked(unit, 0);
175
176             JOptionPane.showMessageDialog(null, "For the sales person, " +
177                 "\n Sold the " + unit.getAutoName() +
178                 ", the price difference was: " +
179                 unit.getBoughtSoldCost(null, unit.getSoldPrice()) +
180                 " dollars");
181         }
182     }
183 }
184
185
186 public static void main(String[] args) {
187     new GUICarDealer();
188 }
189 }
190
```