

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <form xmlns="http://www.intellij.com/uidesigner/form/" version="1" bind-to-class="JComboBoxExampleGUI">
3   <grid id="27dc6" binding="PanelMain" layout-manager="GridLayoutManager" row-count="7" column-count="4" same-
   size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">
4     <margin top="20" left="20" bottom="20" right="20"/>
5     <constraints>
6       <xy x="20" y="20" width="500" height="400"/>
7     </constraints>
8     <properties>
9       <background color="-1"/>
10    </properties>
11    <border type="none"/>
12    <children>
13      <vspacer id="e56b1">
14        <constraints>
15          <grid row="6" column="0" row-span="1" col-span="4" vsize-policy="6" hsize-policy="1" anchor="0" fill
            ="2" indent="0" use-parent-layout="false"/>
16        </constraints>
17      </vspacer>
18      <hspacer id="35127">
19        <constraints>
20          <grid row="4" column="0" row-span="1" col-span="4" vsize-policy="1" hsize-policy="6" anchor="0" fill
            ="1" indent="0" use-parent-layout="false"/>
21        </constraints>
22      </hspacer>
23      <component id="d266d" class="javax.swing.JLabel" binding="LabelCarType">
24        <constraints>
25          <grid row="2" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="0" anchor="8" fill
            ="0" indent="5" use-parent-layout="false"/>
26        </constraints>
27        <properties>
28          <text value="Car Type"/>
29        </properties>
30      </component>
31      <component id="134" class="javax.swing.JLabel" binding="LabelCarBrand">
32        <constraints>

```

```

33         <grid row="2" column="1" row-span="1" col-span="1" vsize-policy="0" hsize-policy="0" anchor="8" fill
    ="0" indent="6" use-parent-layout="false"/>
34     </constraints>
35     <properties>
36         <text value="Brand"/>
37     </properties>
38 </component>
39 <component id="ea3d8" class="javax.swing.JLabel" binding="LabelCarModel">
40     <constraints>
41         <grid row="2" column="2" row-span="1" col-span="2" vsize-policy="0" hsize-policy="0" anchor="8" fill
    ="0" indent="5" use-parent-layout="false"/>
42     </constraints>
43     <properties>
44         <text value="Model"/>
45     </properties>
46 </component>
47 <component id="743a3" class="javax.swing.JComboBox" binding="ComboBoxCarModel">
48     <constraints>
49         <grid row="3" column="2" row-span="1" col-span="2" vsize-policy="0" hsize-policy="2" anchor="8" fill
    ="1" indent="0" use-parent-layout="false"/>
50     </constraints>
51     <properties>
52         <enabled value="false"/>
53     </properties>
54 </component>
55 <component id="41dc2" class="javax.swing.JComboBox" binding="ComboBoxBrand">
56     <constraints>
57         <grid row="3" column="1" row-span="1" col-span="1" vsize-policy="0" hsize-policy="2" anchor="8" fill
    ="1" indent="0" use-parent-layout="false"/>
58     </constraints>
59     <properties>
60         <enabled value="false"/>
61     </properties>
62 </component>
63 <component id="7fb05" class="javax.swing.JComboBox" binding="ComboBoxCarType">
64     <constraints>

```

```

65         <grid row="3" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="2" anchor="8" fill
    ="1" indent="0" use-parent-layout="false"/>
66     </constraints>
67     <properties/>
68 </component>
69 <vspacer id="b60f9">
70     <constraints>
71         <grid row="1" column="1" row-span="1" col-span="1" vsize-policy="6" hsize-policy="1" anchor="0" fill
    ="2" indent="0" use-parent-layout="false"/>
72     </constraints>
73 </vspacer>
74 <component id="4af57" class="javax.swing.JButton" binding="ButtonCalculateCar">
75     <constraints>
76         <grid row="5" column="0" row-span="1" col-span="4" vsize-policy="0" hsize-policy="3" anchor="0" fill
    ="1" indent="0" use-parent-layout="false"/>
77     </constraints>
78     <properties>
79         <text value="Calculate Car"/>
80     </properties>
81 </component>
82 <grid id="a575a" class="javax.swing.JToggleButton" binding="ToggleButtonDarkLightMode" layout-manager="
    GridLayoutManager" row-count="1" column-count="1" same-size-horizontally="false" same-size-vertically="false"
    hgap="-1" vgap="-1">
83     <margin top="0" left="0" bottom="0" right="0"/>
84     <constraints>
85         <grid row="0" column="3" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" anchor="0" fill
    ="0" indent="0" use-parent-layout="false"/>
86     </constraints>
87     <properties>
88         <icon value="Light.png"/>
89         <selected value="false"/>
90         <toolTipText value="Sets Light / Dark Mode"/>
91     </properties>
92     <border type="none"/>
93     <children/>
94 </grid>

```

```
95     </children>
96   </grid>
97 </form>
98
```

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.util.ArrayList;
4 import java.util.Dictionary;
5 import java.util.Hashtable;
6 import java.util.List;
7
8 public class JComboBoxExampleGUI extends JFrame{
9     public JPanel PanelMain;
10    public JComboBox ComboBoxCarModel;
11    public JComboBox ComboBoxBrand;
12    public JComboBox ComboBoxCarType;
13    private JButton ButtonCalculateCar;
14    private JToggleButton ToggleButtonDarkLightMode;
15    private JLabel LabelCarType;
16    private JLabel LabelCarBrand;
17    private JLabel LabelCarModel;
18
19    public JComboBoxExampleGUI(){
20        this.add(PanelMain);
21        this.setSize(500,200);
22        this.setTitle("Rent A Car JComboBox Example");
23        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
24
25        Color whiteColour = new Color(255,255,255);
26        Color blackColour = new Color(0,0,0);
27
28        List<String> carTypes = new ArrayList<>();
29        carTypes.add("SUV");
30        carTypes.add("SEDAN");
31        carTypes.add("SPORTS CAR");
32
33        ComboBoxModel comboBoxCarTypeModel = new DefaultComboBoxModel(carTypes.toArray());
34        ComboBoxCarType.setModel(comboBoxCarTypeModel);
35
36
```

```
37     ComboBoxCarType.addActionListener(e->{
38         ComboBoxCarModel.setModel(new DefaultComboBoxModel());
39         ComboBoxBrand.setModel(SetComboBoxBrand(((String) ComboBoxCarType.getSelectedItem()));
40         ComboBoxBrand.setEnabled(true);
41     });
42
43     ComboBoxBrand.addActionListener(e -> {
44         ComboBoxCarModel.setModel(SetComboCarModel(((String) ComboBoxBrand.getSelectedItem() , (String)
45     ComboBoxCarType.getSelectedItem()));
46         ComboBoxCarModel.setEnabled(true);
47     });
48
49     ToggleButtonDarkLightMode.addActionListener(e -> {
50         if (ToggleButtonDarkLightMode.isSelected()){
51             PanelMain.setBackground(blackColour);
52             LabelCarBrand.setForeground(whiteColour);
53             LabelCarModel.setForeground(whiteColour);
54             LabelCarType.setForeground(whiteColour);
55         }
56         else{
57             PanelMain.setBackground(whiteColour);
58             LabelCarBrand.setForeground(blackColour);
59             LabelCarModel.setForeground(blackColour);
60             LabelCarType.setForeground(blackColour);
61         }
62     });
63
64     ButtonCalculateCar.addActionListener(e -> {
65         try {
66             CalculateCar(((String) ComboBoxCarModel.getSelectedItem()));
67         } catch (Exception exception){
68             JOptionPane.showMessageDialog(this,exception.getMessage());
69         }
70     }
71
```

```
72     });
73 }
74
75 private ComboBoxModel SetComboCarModel(String carBrand,String carType) {
76     List<String> carModels = new ArrayList<>();
77
78     switch (carType) {
79         case "SUV":
80             switch (carBrand){
81                 case "Mazda":
82                     carModels.add("Mazda CX-5");
83                     break;
84                 case "Hyundai":
85                     carModels.add("Hyundai 1.4 MPI Style");
86             }
87             break;
88         case "SEDAN":
89             switch (carBrand){
90                 case "Volkswagen":
91                     carModels.add("Volkswagen Passat 1.6 TDI BMT ComfortLine");
92                     break;
93                 case "Volvo":
94                     carModels.add("Volvo 2.0 D D5 Inscription");
95             }
96             break;
97         case "SPORTS CAR":
98             switch (carBrand){
99                 case "BMW":
100                 carModels.add("BMW 3 Serisi 5.20i");
101                 break;
102                 case "Audi":
103                 carModels.add("Audio A4 Sedan 2.0");
104             }
105             break;
106     }
107 }
```

```
108
109     ComboBoxModel convertTypeModel = new DefaultComboBoxModel(carModels.toArray());
110     return convertTypeModel;
111 }
112
113 public void CalculateCar(String carModel){
114
115     if (carModel == null)
116         throw new NullPointerException("Araba modeli seçiniz");
117
118     Dictionary<String,Integer> modelPriceDictionary = new Hashtable<>();
119     modelPriceDictionary.put("BMW 3 Serisi 5.20i",400000);
120     modelPriceDictionary.put("Audio A4 Sedan 2.0",400000);
121
122     modelPriceDictionary.put("Mazda CX-5",400000);
123     modelPriceDictionary.put("Hyundai 1.4 MPI Style",400000);
124
125     modelPriceDictionary.put("Volkswagen Passat 1.6 TDI BMT ComfortLine",400000);
126     modelPriceDictionary.put("Volvo 2.0 D D5 Inscription",400000);
127
128     String carType = (String) ComboBoxCarType.getSelectedItem();
129     String carBrand = (String) ComboBoxBrand.getSelectedItem();
130
131     String message = "You choosed " + carType +
132         " type " + carBrand +
133         " brand " + carModel +
134         " model \n\n" + "Price is " +
135         modelPriceDictionary.get(carModel);
136
137     JOptionPane.showMessageDialog(this,message);
138 }
139
140 public static ComboBoxModel SetComboBoxBrand(String carType){
141
142     List<String> brands = new ArrayList<>();
143
```



```
144     switch (carType){
145         case "SUV" :
146             brands.add("Mazda");
147             brands.add("Hyundai");
148         break;
149         case "SEDAN":
150             brands.add("Volkswagen");
151             brands.add("Volvo");
152         break;
153         case "SPORTS CAR":
154             brands.add("BMW");
155             brands.add("Audi");
156         break;
157     }
158
159     ComboBoxModel convertTypeModel = new DefaultComboBoxModel(brands.toArray());
160     return convertTypeModel;
161 }
162
163 public static void main(String[] args){
164     try {
165         UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
166     } catch (ClassNotFoundException e) {
167         e.printStackTrace();
168     } catch (InstantiationException e) {
169         e.printStackTrace();
170     } catch (IllegalAccessException e) {
171         e.printStackTrace();
172     } catch (UnsupportedLookAndFeelException e) {
173         e.printStackTrace();
174     }
175
176     //Set UI Thread to another thread
177     SwingUtilities.invokeLater(() -> {
178         JComboBoxExampleGUI gui = new JComboBoxExampleGUI();
179         gui.setVisible(true);
```

```
180     });  
181     }  
182 }  
183
```



Car Type

Brand

Model

SUV



Calculate Car



Car Type

Brand

Model

SUV



Calculate Car