

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <form xmlns="http://www.intelliij.com/uidesigner/form/" version="1" bind-to-class="JCheckBoxGUI">
3   <grid id="27dc6" binding="PanelMain" layout-manager="GridLayoutManager" row-count="5" column-count="5" same-
size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">
4     <margin top="0" left="0" bottom="0" right="0"/>
5     <constraints>
6       <xy x="20" y="20" width="500" height="400"/>
7     </constraints>
8     <properties/>
9     <border type="none"/>
10    <children>
11      <vspacer id="7b1aa">
12        <constraints>
13          <grid row="4" column="0" row-span="1" col-span="5" vsize-policy="6" hsize-policy="1" anchor="0" fill
="2" indent="0" use-parent-layout="false"/>
14        </constraints>
15      </vspacer>
16      <component id="7002e" class="javax.swing.JTextField" binding="TextFieldSample">
17        <constraints>
18          <grid row="1" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="6" anchor="8" fill
="1" indent="0" use-parent-layout="false">
19            <preferred-size width="150" height="-1"/>
20          </grid>
21        </constraints>
22        <properties/>
23      </component>
24      <component id="6b0d4" class="javax.swing.JButton" binding="ButtonShowResult">
25        <constraints>
26          <grid row="1" column="3" row-span="1" col-span="2" vsize-policy="0" hsize-policy="3" anchor="0" fill
="1" indent="0" use-parent-layout="false"/>
27        </constraints>
28        <properties>
29          <font size="16"/>
30          <text value="Show Result"/>
31        </properties>
32      </component>

```

```

33 <component id="6b308" class="javax.swing.JCheckBox" binding="CheckBoxBold">
34 <constraints>
35 <grid row="2" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="3" anchor="8" fill
   ="0" indent="0" use-parent-layout="false"/>
36 </constraints>
37 <properties>
38 <font size="16"/>
39 <text value="Bold"/>
40 </properties>
41 </component>
42 <component id="2b558" class="javax.swing.JCheckBox" binding="CheckBoxUnderline">
43 <constraints>
44 <grid row="2" column="1" row-span="1" col-span="1" vsize-policy="0" hsize-policy="3" anchor="8" fill
   ="0" indent="0" use-parent-layout="false"/>
45 </constraints>
46 <properties>
47 <font size="16"/>
48 <text value="Underline"/>
49 </properties>
50 </component>
51 <component id="712d1" class="javax.swing.JCheckBox" binding="CheckBoxItalic">
52 <constraints>
53 <grid row="2" column="2" row-span="1" col-span="1" vsize-policy="0" hsize-policy="3" anchor="8" fill
   ="0" indent="0" use-parent-layout="false"/>
54 </constraints>
55 <properties>
56 <enabled value="true"/>
57 <font size="16"/>
58 <text value="Italic"/>
59 </properties>
60 </component>
61 <component id="904d0" class="javax.swing.JCheckBox" binding="CheckBoxStrikeThrough">
62 <constraints>
63 <grid row="2" column="3" row-span="1" col-span="2" vsize-policy="0" hsize-policy="3" anchor="8" fill
   ="0" indent="0" use-parent-layout="false"/>
64 </constraints>

```

```

65 <properties>
66 <font size="16"/>
67 <text value="Strike Through"/>
68 </properties>
69 </component>
70 <vspacer id="d39ed">
71 <constraints>
72 <grid row="0" column="1" row-span="1" col-span="1" vsize-policy="6" hsize-policy="1" anchor="0" fill
   ="2" indent="0" use-parent-layout="false"/>
73 </constraints>
74 </vspacer>
75 <component id="47b34" class="javax.swing.JLabel">
76 <constraints>
77 <grid row="3" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="8" fill
   ="0" indent="0" use-parent-layout="false"/>
78 </constraints>
79 <properties>
80 <font size="16"/>
81 <text value="Result : "/>
82 </properties>
83 </component>
84 <component id="f0f51" class="javax.swing.JLabel" binding="LabelResult">
85 <constraints>
86 <grid row="3" column="1" row-span="1" col-span="3" vsize-policy="0" hsize-policy="8" fill
   ="0" indent="0" use-parent-layout="false"/>
87 </constraints>
88 <properties>
89 <font size="16" style="0"/>
90 <text value="">
91 <visible value="true"/>
92 </properties>
93 </component>
94 </children>
95 </grid>
96 </form>
97

```

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.font.TextAttribute;
4 import java.util.Map;
5
6 public class JCheckBoxGUI extends JFrame{
7     private JTextField TextFieldSample;
8     private JButton ButtonShowResult;
9     private JCheckBox CheckBoxBold;
10    private JCheckBox CheckBoxUnderline;
11    private JCheckBox CheckBoxItalic;
12    private JCheckBox CheckBoxStrikeThrough;
13    private JLabel LabelResult;
14    private JPanel PanelMain;
15
16    public JCheckBoxGUI(){
17        this.add(PanelMain);
18        this.setSize(500,200);
19        this.setTitle("Text Decoration Changer");
20        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
21
22        ButtonShowResult.addActionListener(e->{
23            try{
24                ChangeTextDecoration();
25            }
26            catch (Exception exception){
27                JOptionPane.showMessageDialog(this,exception.getMessage());
28            }
29        });
30    }
31
32    public void ChangeTextDecoration(){
33        if (TextFieldSample.getText().equals(""))
34            throw new NullPointerException("You didn't enter any text");
35
36        String text = TextFieldSample.getText();
```

```

37 LabelResult.setText(text);
38
39 Font f = LabelResult.getFont();
40 Map attributes = f.getAttributes();
41
42 LabelFontChangeBold(attributes, f);
43 LabelFontChangeItalic(attributes, f);
44 LabelFontChangeUnderline(attributes, f);
45 LabelFontChangeStrikeThrough(attributes, f);
46
47 LabelResult.setText(text);
48
49 }
50
51 private void LabelFontChangeStrikeThrough(Map attributes, Font f) {
52     if (CheckBoxStrikeThrough.isSelected()){
53         attributes.put(TextAttribute.STRIKETHROUGH, TextAttribute.STRIKETHROUGH_ON);
54     }
55     else{
56         attributes.put(TextAttribute.STRIKETHROUGH, -1);
57     }
58
59 LabelResult.setFont(f.deriveFont(attributes));
60
61
62 }
63
64 private void LabelFontChangeUnderline(Map attributes, Font f) {
65     if (CheckBoxUnderline.isSelected()){
66         attributes.put(TextAttribute.UNDERLINE, TextAttribute.UNDERLINE_ON);
67     }
68     else{
69         attributes.put(TextAttribute.UNDERLINE, -1);
70     }
71 LabelResult.setFont(f.deriveFont(attributes));
72 }

```

```

73
74 private void LabelFontChangeItalic(Map attributes,Font f) {
75     if (CheckBoxItalic.isSelected()){
76         attributes.put(TextAttribute.POSTURE, TextAttribute.POSTURE_OBLIQUE);
77     }
78     else{
79         attributes.put(TextAttribute.POSTURE, TextAttribute.POSTURE_REGULAR);
80     }
81
82     LabelResult.setFont(f.deriveFont(attributes));
83 }
84
85 public void LabelFontChangeBold(Map attributes,Font f){
86     if (CheckBoxBold.isSelected()){
87         attributes.put(TextAttribute.WEIGHT, TextAttribute.WEIGHT_BOLD);
88     }
89     else{
90         attributes.put(TextAttribute.WEIGHT, TextAttribute.WEIGHT_REGULAR);
91     }
92
93     LabelResult.setFont(f.deriveFont(attributes));
94 }
95
96 public static void main(String[] args){
97     try {
98         UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
99     } catch (ClassNotFoundException e) {
100         e.printStackTrace();
101     } catch (InstantiationException e) {
102         e.printStackTrace();
103     } catch (IllegalAccessException e) {
104         e.printStackTrace();
105     } catch (UnsupportedLookAndFeelException e) {
106         e.printStackTrace();
107     }
108

```

```
109 //Set UI Thread to another thread
110 SwingUtilities.invokeLater(() -> {
111     JCheckBoxGUI gui = new JCheckBoxGUI();
112     gui.setVisible(true);
113 });
114 }
115
116 }
117
```