```
- 0 😣
                                                                                  vim JMenuFrame.java
 3 Jared Dyreson
 4 CWID: 889546529
 5 JMenuFrame.java ->
 6 csrc compile: TRUE
 8 */
10 import javax.swing.*;
11 import java.awt.*;
12 import java.awt.event.*;
13 import java.awt.Color;
14 import java.text.MessageFormat;
15
16 import java.io.*;
17 import java.util.*;
18
19 public class JMenuFrame extends JFrame implements ActionListener{
20
           // there may be some duplicate elements and that's because once an element is applied
21
           // to a container, you cannot use it in another container
22
23
           JMenuBar main bar = new JMenuBar();
24
           JMenu file menu = new JMenu("File");
25
            JMenuItem exit = new JMenuItem("Exit");
26
27
           JMenu select bar = new JMenu("Select");
28
           JMenu font selection = new JMenu("Font");
29
           JMenu color menu = new JMenu("Color");
30
           ¡JMenu subomenu = new JMenu("Submenu");
31
32
           String[] select_options = {"Home", "Congratulations Card", "Birthday Card"};
33
           String[] font options = {"Serif", "Noto Sans CJK JP DemiLight", "Suruma"};
34
           String[] color_options = {"Yellow", "Red", "Green"};
35
           Color[] color_parallel = {Color.YELLOW, Color.RED, Color.GREEN};
36
           Color current frame color = Color.LIGHT GRAY;
37
38
           JLabel application_banner = new JLabel("Card Designer Application");
39
           JLabel instructions = new JLabel("Then Get Started By Making Selections From the Menu");
           JLabel instruction copy = new JLabel("Then Get Started By Making Selections From the Menu");
40
41
42
           JLabel birthday_main_message = new JLabel();
           JLabel congrats main message = new JLabel();
43
44
45
           String name instructions = "Please Enter The Recipient's Name";
           JTextField name_field = new JTextField();
46
47
48
           JTextArea birthday_card_message = new JTextArea();
           JTextArea congrats card message = new JTextArea();
49
50
51
           CardLayout card layout = new CardLayout();
           JPanel birthday panel = new JPanel();
52
"JMenuFrame.java" 201L, 9717C
                                                                                                                                                              1,1
                                                                                                                                                                            Top
53
           JPanel congrats_panel = new JPanel();
54
           JPanel home panel = new JPanel();
55
56
           private final int FRAME HEIGHT = 500, FRAME WIDTH = 500;
57
58
           public JMenuFrame(){
59
                   super("Card Builder");
60
61
                   // defining all features of the JFrame
62
                   this.setDefaultCloseOperation(JFrame.EXIT ON CLOSE);
63
                   this.setSize(FRAME HEIGHT, FRAME WIDTH);
64
                   this.setLayout(card_layout);
65
                   this.setLocationRelativeTo(null);
66
                   this.setJMenuBar(main_bar);
67
68
                   // all components for the home card layout
69
                   home panel.add(application banner);
70
                   home panel.setLayout(new GridLayout(3, 0));
71
                   home panel.setBackground(this.current frame color);
72
                   home panel.add(name field);
73
                   home panel.add(instructions);
74
                   this.add(home panel, "HOME");
75
76
                   // all components for the congratulations card layout
77
                   congrats panel.setLayout(new GridLayout(3, 0));
78
                   congrats_panel.add(congrats_main_message);
79
                    congrats panel.add(congrats card message);
                   congrats panel.add(instructions);
80
81
                   this.add(congrats panel, "CONGRATS");
82
83
                   // all components for the birthday card layout
84
                   birthday panel.setLayout(new GridLayout(3, 0));
85
                   birthday panel.add(birthday main message);
86
                   birthday panel.add(birthday card message);
87
                   birthday panel.add(instruction copy);
88
                   this.add(birthday_panel, "BIRTHDAY");
89
90
                   // since each menu item has three elements (depth 2), i decided to run a loop instead of writing
                   // nine times the amount of code
91
                    for(int i = 0; i < select options.length; ++i){</pre>
93
                            // construct each menu item
94
                            JMenuItem select element = new JMenuItem(select_options[i]);
95
                            JMenuItem font element = new JMenuItem(font options[i]);
96
                            JMenuItem color element = new JMenuItem(color options[i]);
97
98
                            // make it have an action listener
99
                            select element.addActionListener(this);
100
                            font element.addActionListener(this);
101
                            color element.addActionListener(this);
102
103
                            // put it in its respective container
                            select bar.add(select_element);
104
                                                                                                                                                              104,9
                                                                                                                                                                            34%
                            font selection.add(font element);
                            color menu.add(color element);
107
                   // add all the menu's (depth 1)
108
109
                   main bar.add(file menu);
110
                   main bar.add(select bar);
                   main bar.add(color menu);
111
112
                   main_bar.add(font_selection);
113
114
                   file menu.add(exit);
115
116
                   exit.addActionListener(this);
117
118
                   name_field.setText(this.name_instructions);
119
           public static void main(String[] args){
120
121
                   // Auto generated with caffine and getty@.service
122
                   JMenuFrame frame = new JMenuFrame();
123
                   frame.setVisible(true);
124
125
           public int find pos(String[] arr, String value){
126
                   // find where the string is in an array of strings
127
                   for(int i = 0; i < arr.length; ++i){
                            if(arr[i] == value){ return i; }
128
129
130
                   // if not found, return -1
131
                   return -1;
132
133
           @Override
134
                   public void actionPerformed(ActionEvent event){
135
                            // this is where the code gets....creative
136
                            // since we don't have declared objects for the MenuItems
137
                            // we still need a way to access and call them
139
                            // this object handles which drop down is called
140
                            JMenuItem text source = (JMenuItem)event.getSource();
141
                            // this object only exists to allow us to exit
142
                            Object source = event.getSource();
143
144
                            // we attempt to find them in the inside each array
145
                            // 2/3 of these values will be -1 because only one drop down can be called
146
                            int selection_position = find_pos(select_options, text_source.getText());
147
                            int font_position = find_pos(font_options, text_source.getText());
148
                            int color position = find pos(color options, text source.getText());
149
150
                            // here we retrieve the name given in the textfield
151
                            // if someone did not put anything in we will give them
152
153
                            // a default name of Timmy and save it to the textfield
154
                            String name = name_field.getText();
155
                            String font name = "";
                                                                                                                                                              156,0-1
                                                                                                                                                                            69%
                            // find the font and if it cannot find the font give a default of Ubuntu
158
                            if(font_position >= 0){ font_name = font_options[font_position]; }
159
                            else{ font name = "Ubuntu"; }
160
161
                            // if color in array
162
                            if(color position >= 0){
163
                                    // if the current color of the JPanel is not the same as the one looked up
164
                                    if(this.current frame color != color parallel[color position]){
165
                                            // update the color
166
                                            this.current_frame_color = color_parallel[color_position];
167
168
169
                            // since we don't know what frame will be displayed, each frame gets the same color
170
                            home panel.setBackground(this.current frame color);
171
                            congrats panel.setBackground(this.current frame color);
172
                            birthday panel.setBackground(this.current frame color);
173
174
                            if(selection_position >= 0){
175
                                    // here we treat the card layout like an array
176
                                    // 0 -> home
177
                                    // 1 -> congratulations
178
                                    // 2 -> birthday
179
                                    switch(selection position){
180
                                            case 0:
181
                                                    card layout.show(getContentPane(), "HOME");
                                                    break;
183
                                            case 1:
184
                                                    card layout.show(getContentPane(), "CONGRATS");
185
186
187
                                                    congrats main message.setText(MessageFormat.format("Congratulations {0}", name));
                                                    break;
                                            case 2:
188
                                                    card_layout.show(getContentPane(), "BIRTHDAY");
189
                                                    birthday main message.setText(MessageFormat.format("Happy Birthday {0}", name));
190
                                                    break;
191
192
                           }
else{
193
194
                                    // since we don't know what frame will be displayed, each frame gets the same font
195
196
                                    congrats card message.setFont(new Font(font name, Font.ITALIC, 24));
                                    birthday card message.setFont(new Font(font name, Font.ITALIC, 24));
197
198
199
                            if(source == exit){ System.exit(0); }
201 }
                                                                                                                                                              201,1
                                                                                                                                                                            Bot
```



```
1: /*
 2:
 3: Jared Dyreson
 4: CWID: 889546529
 5: JMenuFrame.java ->
 6: csrc_compile: TRUE
 8: */
 9:
10: import javax.swing.*;
11: import java.awt.*;
12: import java.awt.event.*;
13: import java.awt.Color;
14: import java.text.MessageFormat;
15:
16: import java.io.*;
17: import java.util.*;
18:
19: public class JMenuFrame extends JFrame implements ActionListener{
20:
            // there may be some duplicate elements and that's because once an element is applied
21:
            // to a container, you cannot use it in another container
22:
23:
            JMenuBar main_bar = new JMenuBar();
24:
            JMenu file_menu = new JMenu("File");
25:
            JMenuItem exit = new JMenuItem("Exit");
26:
27:
            JMenu select bar = new JMenu("Select");
28:
            JMenu font selection = new JMenu("Font");
29:
            JMenu color_menu = new JMenu("Color");
30:
            JMenu sub_menu = new JMenu("Submenu");
31:
32:
            String[] select_options = {"Home", "Congratulations Card", "Birthday Card"};
33:
            String[] font_options = {"Serif", "Noto Sans CJK JP DemiLight", "Suruma"};
            String[] color_options = {"Yellow", "Red", "Green"};
34:
35:
            Color[] color_parallel = {Color.YELLOW, Color.RED, Color.GREEN};
36:
            Color current_frame_color = Color.LIGHT_GRAY;
37:
            JLabel application_banner = new JLabel("Card Designer Application");
38:
39:
            JLabel instructions = new JLabel ("Then Get Started By Making Selections From the Menu");
40:
            JLabel instruction_copy = new JLabel("Then Get Started By Making Selections From the Menu");
41:
42:
            JLabel birthday_main_message = new JLabel();
43:
            JLabel congrats_main_message = new JLabel();
44:
45:
            String name_instructions = "Please Enter The Recipient's Name";
46:
            JTextField name_field = new JTextField();
47:
48:
            JTextArea birthday card message = new JTextArea();
```

JMenuFrame.java

```
49:
            JTextArea congrats_card_message = new JTextArea();
50:
51:
            CardLayout card_layout = new CardLayout();
52:
            JPanel birthday panel = new JPanel();
53:
            JPanel congrats_panel = new JPanel();
54:
            JPanel home panel = new JPanel();
55:
56:
            private final int FRAME_HEIGHT = 500, FRAME_WIDTH = 500;
57:
58:
            public JMenuFrame() {
59:
                    super("Card Builder");
60:
61:
                    // defining all features of the JFrame
62:
                    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
63:
                    this.setSize(FRAME_HEIGHT, FRAME_WIDTH);
64:
                    this.setLayout(card layout);
65:
                    this.setLocationRelativeTo(null);
66:
                    this.setJMenuBar(main_bar);
67:
68:
                    // all components for the home card layout
69:
                    home_panel.add(application_banner);
70:
                    home panel.setLayout(new GridLayout(3, 0));
71:
                    home_panel.setBackground(this.current_frame_color);
72:
                    home_panel.add(name_field);
73:
                    home panel.add(instructions);
74:
                    this.add(home_panel, "HOME");
75:
76:
                    // all components for the congratulations card layout
                    congrats_panel.setLayout(new GridLayout(3, 0));
77:
78:
                    congrats_panel.add(congrats_main_message);
79:
                    congrats_panel.add(congrats_card_message);
80:
                    congrats_panel.add(instructions);
81:
                    this.add(congrats_panel, "CONGRATS");
82:
83:
                    // all components for the birthday card layout
84:
                    birthday_panel.setLayout(new GridLayout(3, 0));
85:
                    birthday panel.add(birthday main message);
86:
                    birthday_panel.add(birthday_card_message);
87:
                    birthday_panel.add(instruction_copy);
                    this.add(birthday_panel, "BIRTHDAY");
88:
89:
90:
                    // since each menu item has three elements (depth 2), i decided to run a loop instead of writin
                    // nine times the amount of code
91:
92:
                    for(int i = 0; i < select_options.length; ++i){</pre>
93:
                            // construct each menu item
94:
                            JMenuItem select element = new JMenuItem(select options[i]);
95:
                            JMenuItem font_element = new JMenuItem(font_options[i]);
96:
                            JMenuItem color element = new JMenuItem(color options[i]);
```

JMenuFrame.java

```
97:
 98:
                              // make it have an action listener
                              select_element.addActionListener(this);
 99:
100:
                              font element.addActionListener(this);
101:
                              color_element.addActionListener(this);
102:
103:
                              // put it in its respective container
104:
                              select_bar.add(select_element);
105:
                              font_selection.add(font_element);
106:
                              color_menu.add(color_element);
107:
108:
                     // add all the menu's (depth 1)
109:
                     main_bar.add(file_menu);
110:
                     main_bar.add(select_bar);
111:
                     main_bar.add(color_menu);
112:
                     main_bar.add(font_selection);
113:
114:
                     file_menu.add(exit);
115:
116:
                     exit.addActionListener(this);
117:
118:
                     name_field.setText(this.name_instructions);
119:
120:
             public static void main(String[] args) {
121:
                     // Auto generated with caffine and getty@.service
122:
                     JMenuFrame frame = new JMenuFrame();
123:
                     frame.setVisible(true);
124:
125:
             public int find_pos(String[] arr, String value) {
126:
                     // find where the string is in an array of strings
127:
                     for(int i = 0; i < arr.length; ++i){</pre>
128:
                              if(arr[i] == value) { return i; }
129:
130:
                     // if not found, return -1
131:
                     return -1;
132:
133:
             @Override
134:
                     public void actionPerformed(ActionEvent event) {
135:
                              // this is where the code gets....creative
136:
                              // since we don't have declared objects for the MenuItems
137:
                              // we still need a way to access and call them
138:
139:
                              // this object handles which drop down is called
140:
                              JMenuItem text_source = (JMenuItem)event.getSource();
141:
                              // this object only exists to allow us to exit
142:
                              Object source = event.getSource();
143:
144:
                              // we attempt to find them in the inside each array
```

```
145:
                             // 2/3 of these values will be -1 because only one drop down can be called
                             int selection_position = find_pos(select_options, text_source.getText());
146:
                             int font_position = find_pos(font_options, text_source.getText());
147:
148:
                             int color_position = find_pos(color_options, text_source.getText());
149:
150:
                             // here we retrieve the name given in the textfield
151:
                             // if someone did not put anything in we will give them
152:
                             // a default name of Timmy and save it to the textfield
153:
154:
                             String name = name_field.getText();
155:
                             String font_name = "";
156:
157:
                             // find the font and if it cannot find the font give a default of Ubuntu
158:
                             if(font_position >= 0) { font_name = font_options[font_position]; }
159:
                             else{ font name = "Ubuntu"; }
160:
161:
                             // if color in array
162:
                             if(color_position >= 0){
163:
                                     // if the current color of the JPanel is not the same as the one looked up
164:
                                     if(this.current_frame_color != color_parallel[color_position]) {
165:
                                             // update the color
166:
                                             this.current_frame_color = color_parallel[color_position];
167:
168:
169:
                             // since we don't know what frame will be displayed, each frame gets the same color
170:
                             home_panel.setBackground(this.current_frame_color);
171:
                             congrats panel.setBackground(this.current frame color);
172:
                             birthday panel.setBackground(this.current frame color);
173:
174:
                             if(selection_position >= 0){
175:
                                     // here we treat the card layout like an array
176:
                                     // 0 -> home
177:
                                     // 1 -> congratulations
178:
                                     // 2 -> birthday
179:
                                     switch (selection_position) {
180:
                                              case 0:
181:
                                                      card_layout.show(getContentPane(), "HOME");
182:
                                                      break;
183:
                                              case 1:
184:
                                                      card_layout.show(getContentPane(), "CONGRATS");
185:
                                                      congrats_main_message.setText(MessageFormat.format("Congratulat
186:
                                                      break:
187:
                                              case 2:
188:
                                                      card_layout.show(getContentPane(), "BIRTHDAY");
189:
                                                      birthday main message.setText(MessageFormat.format("Happy Birth
190:
                                                      break:
191:
                                     }
192:
```

```
193:

194:

195:

196:

197:

198:

199:

199:

199:

199:

199:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:

190:
```

```
1: #!/usr/bin/env zsh
 2:
 3: # support for Java, CPP, C, zsh/sh/bash, Python
 4: # more will be added when I use more file types
 5:
 6: function find_extension() {
 7:
            header="$ (head -n1 "$1")"
            [[ "$(grep "env" "$1")" ]] && file_extension="$(echo "$header" | awk '{print $2}')"
 8:
 9:
            [[ "$(grep "\#\!\/bin" "$1")" ]] && file_extension="$(basename "$1")"
10:
            [[ "$file_extension" == "zsh" ]] && file_extension="bash"
11: }
12:
13: [[ ! -d csrc_exported ]] && mkdir csrc_exported
14:
15: for file in "$@"; do
16:
            [[ ! -f "$file" ]] && break
17:
            filename="$file:t:r"
18:
            file_extension="$file:t:e"
19:
            [[ -z "$file extension" ]] && find extension "$file"
            [[ "$file_extension" == "py" ]] && file_extension="python"
20:
21:
22:
            enscript -rG --word-wrap --line-numbers -p - --highlight="$file_extension" --color=1 -c "$file" \
23:
24:
            ps2pdf - csrc_exported/"$filename".pdf
25: done
26:
```