

ASSIGNMENT # 2

ZARA BAIG
SP24-BSE-128
SECTION A

CODE

```
DataEntryForm.java x UserInfo.java Application.class
1 package com.example.dataentry;
2
3 import javafx.application.Application;
4 import javafx.geometry.Insets;
5 import javafx.geometry.Pos;
6 import javafx.scene.Cursor;
7 import javafx.scene.Scene;
8 import javafx.scene.control.*;
9 import javafx.scene.image.Image;
10 import javafx.scene.image.ImageView;
11 import javafx.scene.layout.*;
12 import javafx.scene.paint.Color;
13 import javafx.scene.text.Font;
14 import javafx.stage.FileChooser;
15 import javafx.stage.Stage;
16
17 import java.io.File;
18 import java.io.IOException;
19 import java.util.ArrayList;
20 import java.util.List;
21
22 public class DataEntryForm extends Application {
23     @Override
24     public void start(Stage stage) throws IOException {
25
26         VBox box = new VBox();
27         List<UserInfo> userInfo = new ArrayList<>();
28
29         Image image = new Image(getClass().getResource("/banner.png").toExternalForm());
30         ImageView imageView = new ImageView(image);
31         box.getChildren().add(imageView);
32
33         GridPane gridPane = new GridPane();
34         gridPane.setPadding(new Insets(20));
35         gridPane.setHgap(50);
36         gridPane.setVgap(10);
37
38         Label nameLabel = new Label("Name");
39         nameLabel.setFont(Font.font("Times New Roman", 18));
40         nameLabel.setAlignment(Pos.BOTTOM_LEFT);
41         gridPane.add(nameLabel, 0, 1);
42         GridPane.setMargin(nameLabel, new Insets(0, 0, 0, 50));
43
44         TextField name = new TextField();
45         name.setFont(Font.font("Times New Roman", 16));
46         gridPane.add(name, 1, 1);
47
48         Label fatherNameLabel = new Label("Father's Name");
49         fatherNameLabel.setFont(Font.font("Times New Roman", 18));
50         fatherNameLabel.setAlignment(Pos.BOTTOM_LEFT);
51         gridPane.add(fatherNameLabel, 0, 2);
52
53         TextField fatherName = new TextField();
54         fatherName.setFont(Font.font("Times New Roman", 16));
55         gridPane.add(fatherName, 1, 2);
56         GridPane.setMargin(fatherNameLabel, new Insets(0, 0, 0, 50));
```

```

57
58 Label cityLabel = new Label(s: "City");
59 cityLabel.setFont(Font.font(s: "Times New Roman", v: 18));
60 cityLabel.setAlignment(Pos.BOTTOM_LEFT);
61 GridPane.setMargin(cityLabel, new Insets(v: 0, v1: 0, v2: 0, v3: 50));
62 gridPane.add(cityLabel, i: 0, i1: 3);
63
64 ComboBox<String> city = new ComboBox<>();
65 city.getItems().addAll(es: "New York", "Florida", "Washington", "Miami");
66 city.setValue("Select a City");
67 gridPane.add(city, i: 1, i1: 3);
68
69 Label addressLabel = new Label(s: "Address");
70 addressLabel.setFont(Font.font(s: "Times New Roman", v: 18));
71 addressLabel.setAlignment(Pos.BOTTOM_LEFT);
72 GridPane.setMargin(addressLabel, new Insets(v: 0, v1: 0, v2: 0, v3: 50));
73 gridPane.add(addressLabel, i: 0, i1: 4);
74
75 TextField address = new TextField();
76 address.setFont(Font.font(s: "Times New Roman", v: 16));
77 gridPane.add(address, i: 1, i1: 4);
78 GridPane.setMargin(fatherNameLabel, new Insets(v: 0, v1: 0, v2: 0, v3: 50));
79
80 Label emailLabel = new Label(s: "Email");
81 emailLabel.setFont(Font.font(s: "Times New Roman", v: 18));
82 emailLabel.setAlignment(Pos.BOTTOM_LEFT);
83 GridPane.setMargin(emailLabel, new Insets(v: 0, v1: 0, v2: 0, v3: 50));
84 gridPane.add(emailLabel, i: 0, i1: 5);

```

```

85
86 TextField email = new TextField();
87 email.setFont(Font.font(s: "Times New Roman", v: 16));
88 gridPane.add(email, i: 1, i1: 5);
89 GridPane.setMargin(fatherNameLabel, new Insets(v: 0, v1: 0, v2: 0, v3: 50));
90
91 Label genderLabel = new Label(s: "Gender");
92 genderLabel.setFont(Font.font(s: "Times New Roman", v: 18));
93 genderLabel.setAlignment(Pos.BOTTOM_LEFT);
94 GridPane.setMargin(genderLabel, new Insets(v: 20, v1: 0, v2: 0, v3: 50));
95 gridPane.add(genderLabel, i: 0, i1: 6);
96
97 RadioButton radioButton = new RadioButton(s: "Male");
98 RadioButton radioButton1 = new RadioButton(s: "Female");
99 radioButton.setFont(Font.font(s: "Times New Roman", v: 14));
100 radioButton1.setFont(Font.font(s: "Times New Roman", v: 14));
101
102 VBox genderBox = new VBox();
103
104 ToggleGroup genderToggleGroup = new ToggleGroup();
105 radioButton1.setToggleGroup(genderToggleGroup);
106 radioButton.setToggleGroup(genderToggleGroup);
107 genderBox.getChildren().addAll(radioButton, radioButton1);
108 genderBox.setSpacing(5);
109 GridPane.setMargin(genderBox, new Insets(v: 0, v1: 0, v2: 0, v3: 50));
110 gridPane.add(genderBox, i: 0, i1: 7);
111

```

```

112 VBox imageContainer = new VBox();
113
114 Label imageSelect = new Label( s: "Select an image");
115 imageSelect.setFont(Font.font( s: "Times New Roman", v: 16));
116 imageSelect.setAlignment(Pos.BOTTOM_LEFT);
117 imageSelect.setPadding(new Insets( v: 0, v1: 0, v2: 10, v3: 0));
118
119 Button imageButton = new Button( s: "Choose Image");
120 imageButton.setOnAction( ActionEvent e -> {
121     FileChooser fileChooser = new FileChooser();
122     fileChooser.getExtensionFilters().add(new FileChooser.ExtensionFilter( s: "Image Files", ...strings: "*.png", "*.jpg", "*.jpeg",
123
124     File selectedFile = fileChooser.showOpenDialog(stage);
125
126     if (selectedFile != null) {
127         Image selectedImage = new Image(selectedFile.toURI().toString());
128         ImageView imageView1 = new ImageView(selectedImage);
129         imageView1.setFitWidth(100);
130         imageView1.setFitHeight(100);
131
132         HBox imageBox = new HBox(imageView1);
133         imageBox.setAlignment(Pos.TOP_CENTER);
134         imageBox.setPadding(new Insets( v: 0, v1: 0, v2: 10, v3: 0));
135         imageContainer.getChildren().add(imageBox);
136     }
137 });
138

```

```

139 imageContainer.getChildren().addAll(imageSelect, imageButton);
140 imageContainer.setPrefSize( v: 200, v1: 200);
141 imageContainer.setAlignment(Pos.CENTER);
142 imageContainer.setBorder(new Border(new BorderStroke(
143     Color.BLACK,
144     BorderStrokeStyle.SOLID,
145     new CornerRadii( v: 20),
146     BorderStroke.DEFAULT_WIDTHS
147 ));
148
149 GridPane.setMargin(imageContainer, new Insets( v: 10, v1: 0, v2: 0, v3: 50));
150 gridPane.add(imageContainer, i: 1, i1: 7);
151
152
153 Button submit = new Button( s: "SUBMIT");
154 submit.setBackground(new Background(new BackgroundFill(Color.CADETBLUE, new CornerRadii( v: 10), Insets.EMPTY)));
155 submit.setFont(Font.font( v: 16));
156 submit.setTextFill(Color.WHITE);
157 submit.setPadding(new Insets( v: 5));
158 submit.setPrefWidth(120);
159 submit.setPrefHeight(8);
160 submit.setCursor(Cursor.HAND);
161 gridPane.add(submit, i: 0, i1: 7);
162 GridPane.setMargin(submit, new Insets( v: 0, v1: 0, v2: 10, v3: 50));
163

```

```

164         submit.setOnAction( ActionEvent e -> {
165             String userName = name.getText().trim();
166             String userFatherName = fatherName.getText().trim();
167             String userCity = city.getValue();
168             String userAddress = address.getText().trim();
169             String userEmail = email.getText().trim();
170             RadioButton selectedGender = (RadioButton) genderToggleGroup.getSelectedToggle();
171             String userGender = selectedGender.getText();
172
173             UserInfo newUser = new UserInfo(userName, userFatherName, userCity, userAddress, userEmail, userGender);
174             userInfo.add(newUser);
175             System.out.println(newUser);
176
177             openNewScene(stage, newUser);
178         });
179
180         Scene scene = new Scene(box, 500, 650);
181         box.setBackground(new Background(new BackgroundFill(Color.ANTIQUEWHITE, CornerRadii.EMPTY, Insets.EMPTY)));
182         box.getChildren().add(gridPane);
183         stage.setTitle("Data Entry Form");
184         stage.setScene(scene);
185         stage.setResizable(false);
186         stage.show();
187     }

```

```

188
189 @ private void openNewScene(Stage stage, UserInfo user) { 1 usage
190     VBox v = new VBox(10);
191     v.setPadding(new Insets(10, 20));
192     v.setAlignment(Pos.TOP_CENTER);
193
194     Label welcomeLabel = new Label("WELCOME!");
195     Label loginLabel = new Label("Login Info");
196     Label nameLabel = new Label("Name: " + user.getName());
197     Label fatherNameLabel = new Label("Father's Name: " + user.getFatherName());
198     Label cityLabel = new Label("City: " + user.getCity());
199     Label addressLabel = new Label("Address: " + user.getAddress());
200     Label emailLabel = new Label("Email: " + user.getEmail());
201     Label genderLabel = new Label("Gender: " + user.getGender());
202
203     welcomeLabel.setPadding(new Insets(50, 0, 0, 0));
204
205     welcomeLabel.setFont(Font.font("Times New Roman", 20));
206     loginLabel.setFont(Font.font("Times New Roman", 20));
207     nameLabel.setFont(Font.font("Times New Roman", 18));
208     fatherNameLabel.setFont(Font.font("Times New Roman", 18));
209     cityLabel.setFont(Font.font("Times New Roman", 18));
210     addressLabel.setFont(Font.font("Times New Roman", 18));
211     emailLabel.setFont(Font.font("Times New Roman", 18));
212     genderLabel.setFont(Font.font("Times New Roman", 18));

```

```

214     v.getChildren().addAll(welcomeLabel, loginLabel, nameLabel, fatherNameLabel, cityLabel, addressLabel, emailLabel, genderLabel);
215     v.setSpacing(10);
216     Scene newScene = new Scene(v, 500, 650);
217     v.setBackground(new Background(new BackgroundFill(Color.LAVENDER, CornerRadii.EMPTY, Insets.EMPTY)));
218
219
220     stage.setScene(newScene);
221 }

```

```

224
225 ▶ public static void main(String[] args) {
226     launch();
227 }
228 }
229

```

```

DataEntryForm.java  UserInfo.java  Application.class
1  package com.example.dataentry;
2
3  public class UserInfo {
4
5      private String name; 4 usages
6      private String fatherName; 4 usages
7      private String city; 4 usages
8      private String address; 4 usages
9      private String email; 4 usages
10     private String gender; 4 usages
11
12     public UserInfo(String name, String fatherName, String city, String address, String email, String gender) { 1 usage
13         this.name = name;
14         this.fatherName = fatherName;
15         this.city = city;
16         this.address = address;
17         this.email = email;
18         this.gender = gender;
19     }
20
21     public String getName() { 1 usage
22         return name;
23     }
24
25     public void setName(String name) { no usages
26         this.name = name;
27     }
28
29     public String getFatherName() { 1 usage
30         return fatherName;
31     }
32

```

```

33     public void setFatherName(String fatherName) { no usages
34         this.fatherName = fatherName;
35     }
36
37     public String getCity() { 1 usage
38         return city;
39     }
40
41     public void setCity(String city) { no usages
42         this.city = city;
43     }
44
45     public String getAddress() { 1 usage
46         return address;
47     }
48
49     public void setAddress(String address) { no usages
50         this.address = address;
51     }
52
53     public String getEmail() { 1 usage
54         return email;
55     }
56
57     public void setEmail(String email) { no usages
58         this.email = email;
59     }
60
61     public String getGender() { 1 usage
62         return gender;
63     }

```

```

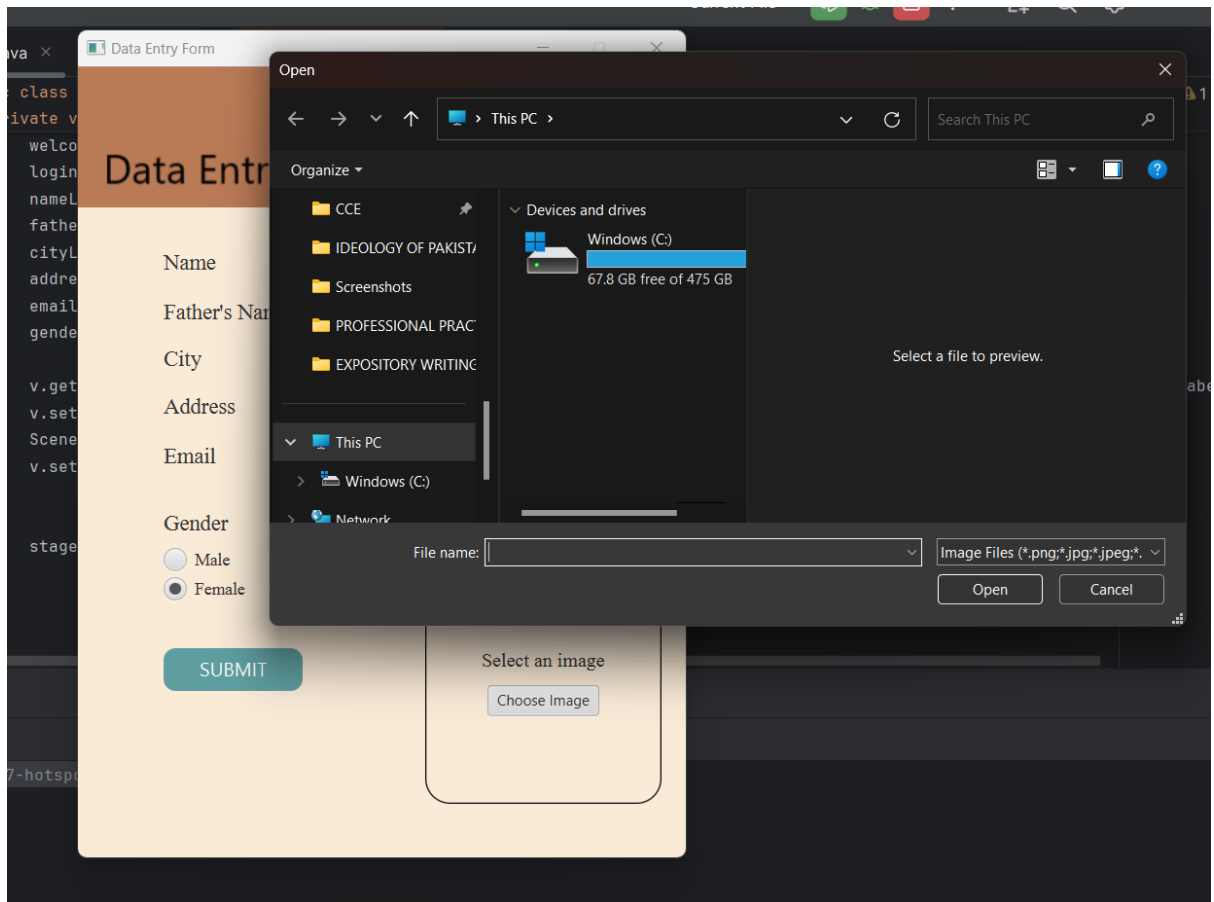
public void setGender(String gender) { no usages
    this.gender = gender;
}

@Override
public String toString() {
    return "User\n" +
        "Name: " + name + " | " +
        "Father Name: " + fatherName + " | " +
        "City: " + city + " | " +
        "Address: " + address + " | " +
        "Email: " + email + " | " +
        "Gender: " + gender;
}
}

```

OUTPUT

The screenshot shows a Java Swing window titled "Data Entry Form". The window has a standard macOS-style title bar with a green icon, a minus sign, a maximize button, and a close button. The main content area has a light orange background. At the top, there is a brown header bar with the text "Data Entry Form" in white. Below the header, the form is organized into several sections. The first section contains five input fields: "Name", "Father's Name", "City" (a dropdown menu with "Select a City" as the selected option), "Address", and "Email". The "City" dropdown is a light gray button with a downward arrow. The second section contains the "Gender" label and two radio buttons: "Male" and "Female". At the bottom left, there is a green "SUBMIT" button. On the right side, there is a rounded rectangle containing the text "Select an image" and a "Choose Image" button. The background of the entire application window is dark gray, and some code from the previous image is visible in the background.



Data Entry Form

Data Entry Form

Name

Zara Baig

Father's Name

M Naeem

City

New York

Address

25th street

Email

abc@gmail.com

Gender


☐ Male

☒ Female

SUBMIT

Select an image

Choose Image



Data Entry Form

WELCOME!

Login Info

Name: Zara Baig

Father's Name: M Naeem

City: New York

Address: 25th street

Email: abc@gmail.com

Gender: Female

```
"C:\Program Files\Eclipse Adoptium\jdk-21.0.4-hotspot\bin\java.exe" ..
```

```
User
```

```
Name: Zara Baig | Father Name: M Naeem | City: New York | Address: 25th street | Email: abc@gmail.com | Gender: Female
```