JavaFX

A brief reminder

- Label
- Button
- RadioButton
- ToggleButton
- CheckBox
- textField
- PasswordField

Critical Thinking

1. What kind of role does a CSS file play in a JavaFX app?

   a) Describes the style
   b) Configures the compilation process
   c) Lays out components into their places
   d) Expresses the application logic

Answer: a

---

2. What is the main functional difference between ToggleButton and RadioButton?

   a) RadioButton cannot be a member of a ToggleGroup while ToggleButton can.
   b) When grouped with a ToggleGroup you can select multiple ToggleButtons while you cannot do the same thing with RadioButtons.
   c) RadioButtons cannot be deselected while ToggleButtons can.
   d) They are identical functionally, only their appearance differ.

Answer: c

---

3. What is the main functional difference between ChoiceBox and ComboBox?

   a) ChoiceBox allows single selection while ComboBox allows multiple selections.
   b) The order of options is adjustable in ChoiceBox but not in ComboBox.
   c) ComboBox takes a generic type argument while ChoiceBox does not.
   d) ComboBox has an option which allows user to enter a custom value besides the predefined options.

Answer: d

---

4. What is wrong with the following code? Assume that a ChoiceBox and a ComboBox is properly defined in the associated .fxml file and their fx:id property is set to "choice" and "combo", respectively.

```
public class SampleController implements Initializable {
    @FXML
    ChoiceBox<String> choice;
    @FXML
    ComboBox<String> combo;
    @Override
    public void initialize(URL location, ResourceBundle resources)
    {
        choice = new ChoiceBox<>();
        combo = new ComboBox<>();
        choice.getItems().addAll("hello", "world");
        combo.getItems().add("option");
    }
}
```

```
a) Multiple usage of @FXML annotation causes error.
b) The method name getItems is wrongly spelled, it should be getOptions.
c) Assigning new objects to choice and combo breaks the connection with .fxml document.
d) The method addAll cannot take multiple arguments.
```

Answer: c

---

5. I defined a Button in the corresponding .fxml file and set its fx:id property to "button". However the following controller code still doesn't work. What could be the problem?

```
public class SampleController  {
    public void buttonAction() {
```

```
        System.out.println("hey");
    }
}
```

a) The handler method buttonAction must take an argument of type ActionEvent.
b) The onAction property of Button in .fxml file may not be set appropriately.
c) Event handlers must return a value.
d) Using a @FXML annotation a Button reference must be put in SampleController class.

Answer: b

Practice

1. Write a JavaFX app which displays random images when user clicks a button (gets images from internet).

Answer:

```
public void start(Stage stage) {

    BorderPane root = new BorderPane();

    String imgUrl = "https://picsum.photos/200/300";
    Button button = new Button("change");

    ImageView img = new ImageView(imgUrl);
    root.setCenter(img);
    root.setBottom(button);
    BorderPane.setAlignment(button, Pos.CENTER);

    button.setOnAction(e -> {
        img.setImage(new Image(imgUrl));
    });

    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.show();
}
```

---

2. Write a JavaFX app which displays three radio buttons, more than one of which cannot be selected at the same time.

Answer:

```
public void start(Stage stage) {
    RadioButton b1 = new RadioButton("button1");
    RadioButton b2 = new RadioButton("button2");
    RadioButton b3 = new RadioButton("button3");
    // adding buttons to the same toggle group
    ToggleGroup g = new ToggleGroup();
    g.getToggles().addAll(b1, b2, b3);
    HBox box = new HBox(b1, b2, b3);
    box.setSpacing(20);
    box.setPadding(new Insets(50));
    Scene scene = new Scene(box);
    stage.setScene(scene);
    stage.show();
}
```

---

3. write a javaFX app which displays a scene containing a username and a password field, and a button. When the button is clicked, the app should print the username and password to the console.

NOTE: username and passwords are stored in a `HashMap<String, String>` object.

Answer:

```
public void start(Stage stage) {

    Label uname = new Label("Username: ");
    Label pass = new Label("Password: ");
    Label response = new Label();

    Button signIn = new Button("Sign in");
```

```
    TextField unameField = new TextField();
    PasswordField passField = new PasswordField();

    GridPane root = new GridPane();
    root.setAlignment(Pos.CENTER);
    root.add(uname, 0, 0);
    root.add(pass, 0, 1);
    root.add(unameField, 1, 0);
    root.add(passField, 1, 1);
    root.add(signIn, 1, 2);
    root.add(response, 1, 3);

    HashMap<String, String> userList = new HashMap<>();
    userList.put("admin", "12345");

    signIn.setOnAction( e -> {
        String unameEntered = unameField.getText();
        String passEntered = passField.getText();
        if(userList.containsKey(unameEntered) &&
                userList.get(unameEntered).equals(passEntered))
            response.setText("Login successful!");
        else
            response.setText("Access denied!");
    });

    Scene scene = new Scene(root, 100, 100);
    stage.setScene(scene);
    stage.setResizable(false);
    stage.show();
}
```

Project

1. write a todo app using JavaFX consists of a text field where the user can enter a task and an “Add” button to add it to the list. Each task is displayed as a row in a VBox container, containing a label for the task description and a button to mark it as “done”. When the “done” button is clicked, the corresponding row is removed from the list. Clicking the “done” button removes the item from the list and updates the labels for the remaining items.

Answer:

```
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Node;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

public class Main extends Application {
    public static void main(String[] args) {
        launch();
    }
    @Override
    public void start(Stage primaryStage) throws Exception {
        TodoApp app = new TodoApp();
        app.start(primaryStage);
    }
}
class TodoApp extends Application {
    final double width = 400;
    final double height = 300;
    int numItems = 0;
    BorderPane root;
    VBox content;
    public void start(Stage stage) {
        Button addButton = new Button("Add");
        TextField field = new TextField();
        root = new BorderPane();
        content = new VBox();
        HBox bottom = new HBox(field, addButton);
        bottom.setAlignment(Pos.CENTER);
        root.setBottom(bottom);
        root.setCenter(content);
        addButton.setOnAction(e -> {
            String text = field.getText();
            if(text.length() > 0) {
                addRow(text);
            }
        });
        cosmetics();
        finalize(stage);
    }
    public void addRow(String text, TextField field) {
        Button delButton = new Button("done");
        Label id = new Label(numItems+1+"");
        Label item = new Label(text);
```

```
        HBox row = new HBox(30, id, item, delButton);
        row.setAlignment(Pos.CENTER);
        content.getChildren().add(row);
        numItems++;
        field.setText("");
        delButton.setOnAction(e -> {
            updateNums(row);
            content.getChildren().remove(row);
            numItems--;
        });
    }
    public void updateNums(Node row) {
        int rowIndex = content.getChildren().indexOf(row);
        for(int i=rowIndex+1; i<content.getChildren().size(); i++) {
            HBox box = (HBox) content.getChildren().get(i);
            Label l = (Label) box.getChildren().get(0);
            l.setText(i+""); // decrement by one
        }
    }
    public void cosmetics() {
        content.setAlignment(Pos.CENTER);
        BorderPane.setAlignment(content, Pos.CENTER);
        BorderPane.setMargin(content, new Insets(20));
    }
    public void finalize(Stage stage) {
        Scene scene = new Scene(root, width, height);
        stage.setScene(scene);
        stage.setTitle("ToDo App");
        stage.show();
    }
}
```