**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Computer Engineering Department**

Program: MCA, Semester I

**Course: Java Programming**

**Experiment No.10**

PART A

**Aim:** To implement concept of GUI using JavaFX

**Prerequisite: Concept of Java Basic and JavaFx**

**Outcome: After successful completion of this practical students will be able to**

1. Understand and implement concepts of JavaFx architecture
2. Understand and develop GUI using JavaFX classes and Interfaces.

**Theory:**

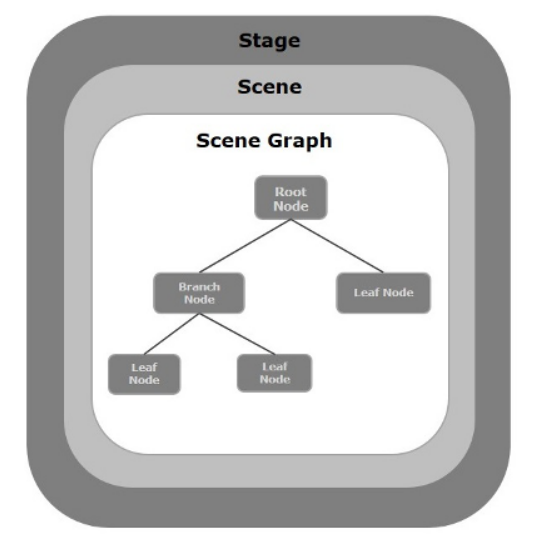
* + It is a Java library and a GUI tool kit designed to develop
    - Web applications
    - Desktop applications
  + Applications created using JavaFX can run on multiple operating systems like
    - Windows, Linux, iOS, Android
    - Several platforms like, Desktops, Web, Mobile Phones, TVS and Tablets etc..
  + It was introduced to supersede the Java Swing GUI Framework
  + It provides more enhanced functionalities
  + It has numerous built-in elements that are inter-connected with each other.
  + It is a collection of
  + API’s, classes and interfaces

**The JavaFX architecture:**

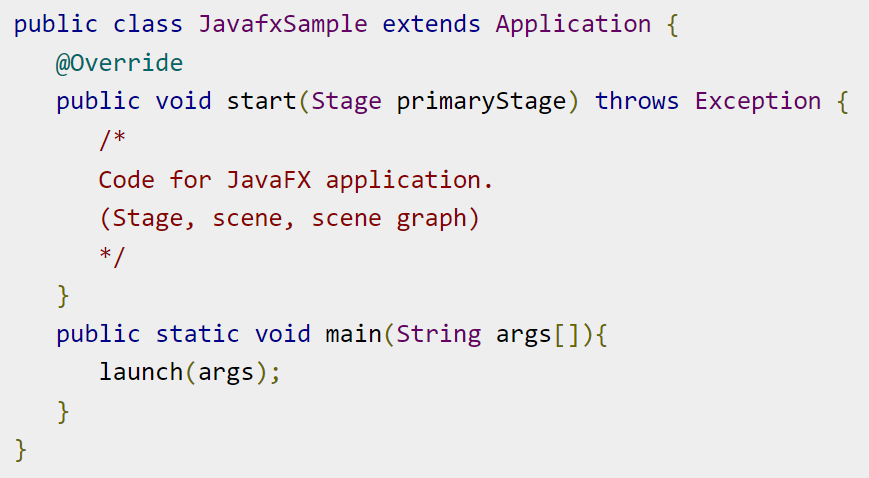


**JavaFX Application:**

* It has three major components namely Stage, Scene and Nodes.



**Creating a JavaFX application:**



**Programs:**

Task 1: Develop JavaFX application that verifies user’s username and password. The appropriate messages are to be displayed.

Task 2: Develop JavaFX application for Calculator that has below functionalities.

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus
6. Power (x raise to y)
7. Square root
8. Square (x raise to 2)
9. Clear – that clears the text fields

**Part B**

|  |  |
| --- | --- |
| **Roll no.: A073** | **Name: Aryan Srivastava** |
| **Class: MCA** | **Batch: 3** |
| **Date of Experiment: 17-10-2023** | **Date of Submission:** |
| **Grade:** |  |

1. **Program scenario and Program code: (**Paste you program code )

**Task 1:**

package javafxapplication1;

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

public class JavaFXApplication1 extends Application {

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage primaryStage) {

primaryStage.setTitle("User Login");

TextField usernameField = new TextField();

PasswordField passwordField = new PasswordField();

Button loginButton = new Button("Login");

Label messageLabel = new Label();

loginButton.setOnAction(e -> {

String username = usernameField.getText();

String password = passwordField.getText();

if (isValidUser(username, password)) {

messageLabel.setText("Login successful");

} else {

messageLabel.setText("Invalid username or password");

}

});

VBox vbox = new VBox(10);

vbox.getChildren().addAll(

new Label("Username:"),

usernameField,

new Label("Password:"),

passwordField,

loginButton,

messageLabel

);

Scene scene = new Scene(vbox, 300, 200);

primaryStage.setScene(scene);

primaryStage.show();

}

private boolean isValidUser(String username, String password) {

return username.equals("admin") && password.equals("password");

}

}

**Task 2:**

import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.TextField;

import javafx.scene.layout.GridPane;

import javafx.stage.Stage;

public class JavaFXApplication2 extends Application {

private TextField inputField;

private double num1 = 0;

private String operator = "";

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage primaryStage) {

primaryStage.setTitle("Calculator");

inputField = new TextField();

inputField.setPrefColumnCount(10);

inputField.setEditable(false);

GridPane grid = new GridPane();

grid.setHgap(10);

grid.setVgap(10);

grid.setPadding(new Insets(10, 10, 10, 10));

grid.add(inputField, 0, 0, 4, 1);

String[] buttonLabels = {

"7", "8", "9", "+",

"4", "5", "6", "-",

"1", "2", "3", "\*",

"0", "C", "=", "/",

"sqrt", "^2", "%", "^"

};

int row = 1;

int col = 0;

for (String label : buttonLabels) {

Button button = createButton(label);

grid.add(button, col, row);

col++;

if (col > 3) {

col = 0;

row++;

}

}

Scene scene = new Scene(grid);

primaryStage.setScene(scene);

primaryStage.show();

}

private Button createButton(String label) {

Button button = new Button(label);

button.setMinSize(50, 50);

button.setOnAction(e -> handleButtonClick(label));

return button;

}

private void handleButtonClick(String label) {

switch (label) {

case "C":

inputField.clear();

num1 = 0;

operator = "";

break;

case "=":

double result = performCalculation();

inputField.setText(Double.toString(result));

num1 = result;

operator = "";

break;

case "sqrt":

num1 = Double.parseDouble(inputField.getText());

double sqrtResult = Math.sqrt(num1);

inputField.setText(Double.toString(sqrtResult));

break;

case "^2":

num1 = Double.parseDouble(inputField.getText());

double squareResult = num1 \* num1;

inputField.setText(Double.toString(squareResult));

break;

default:

if (label.matches("[0-9]+")) {

inputField.appendText(label);

} else if (label.matches("[+\\-\*/%^]")) {

num1 = Double.parseDouble(inputField.getText());

operator = label;

inputField.clear();

}

}

}

private double performCalculation() {

double num2 = Double.parseDouble(inputField.getText());

switch (operator) {

case "+":

return num1 + num2;

case "-":

return num1 - num2;

case "\*":

return num1 \* num2;

case "/":

if (num2 == 0) {

return 0; // You can handle division by zero differently

}

return num1 / num2;

case "%":

return num1 % num2;

case "^":

return Math.pow(num1, num2);

default:

return 0;

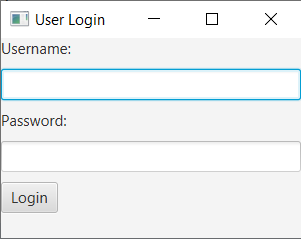
}

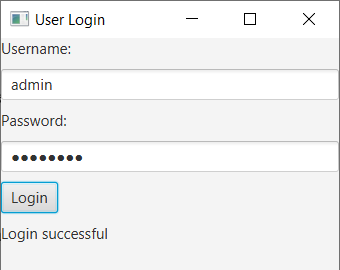
}

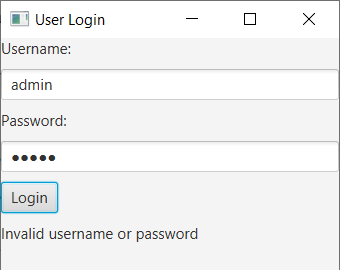
}

1. **Output: (**Paste your program input and output for the program)

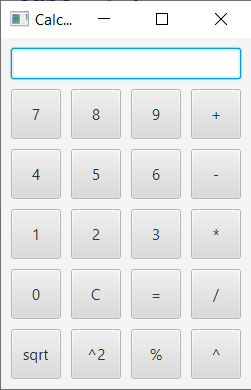
**Task 1:**

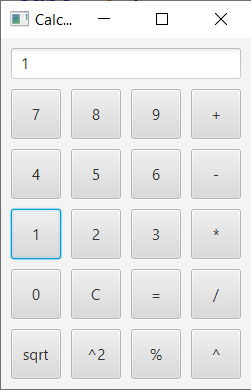
****

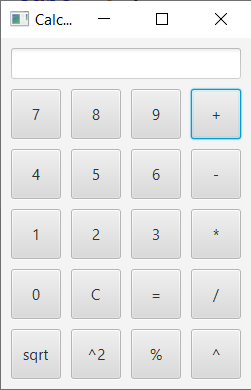
****

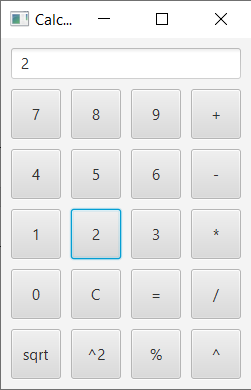
****

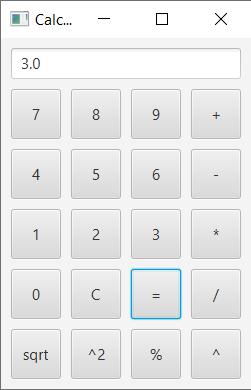
**Task 2:**

****

****

****

****

****

1. **Observation learning and conclusion:** successfully coded JAVA FX