UOIT
CHALLENGE INNOVATE CONNECT

Faculty of Science

Course: CSCI 2020u – Software System Development & Integration
Component: Assignment
Weight: 10%
Deadline: March 6, 2019 (due by 11:59pm)

Collaboration Policy

You are permitted to work on this assignment in a team, and submit the results as a team. For this sort of assignment, with an open-ended component, the collaboration between multiple team members can be beneficial. Between groups, however, please limit the discussion to the level of general strategy (not code). Groups of size 2 are recommended. Larger groups will be considered with the proviso that the marker will mark your assignment with higher expectations. In any case, be sure that all members of the team fully understand all code, otherwise they will miss intended learning objectives, which may be a considerable disadvantage at exam time.

How to Submit

You will maintain a **git repository** for this assignment, which is a public repository. To submit the assignment, create a single file 'README.txt' that contains instructions on how to download, compile, and run your codes for each question. A .zip, .7z, or .rar file will not be acceptable. **Also submit this word file (once you complete) into related drop box on Blackboard before deadline.**

Note: *Comments are mandatory. Failure to properly document your program will result in a deduction on the marks you receive for this (and any other) assignment.*

Remember:

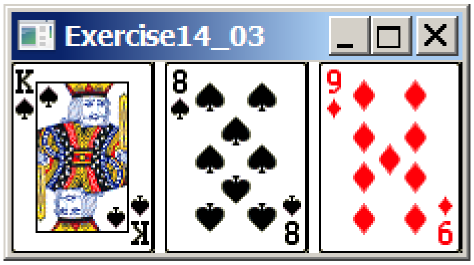
You need to complete this file and submit it in related **drop box on Blackboard**, in addition to uploading your codes in your **git repository**, before deadline.

Question 1: Displaying Three Cards

Problem Description:

Display a frame that contains three labels. Each label displays a card, as shown in the figure below. The card image files are named 1.png, 2.png, ..., 54.png and stored in the image/card directory. All three cards are distinct and selected randomly.

The image icons can be found in the attached card folder.



Your Task:

1. Create three `ImageView` and set their icons using the images.
2. Display three images from 54 image cards randomly.

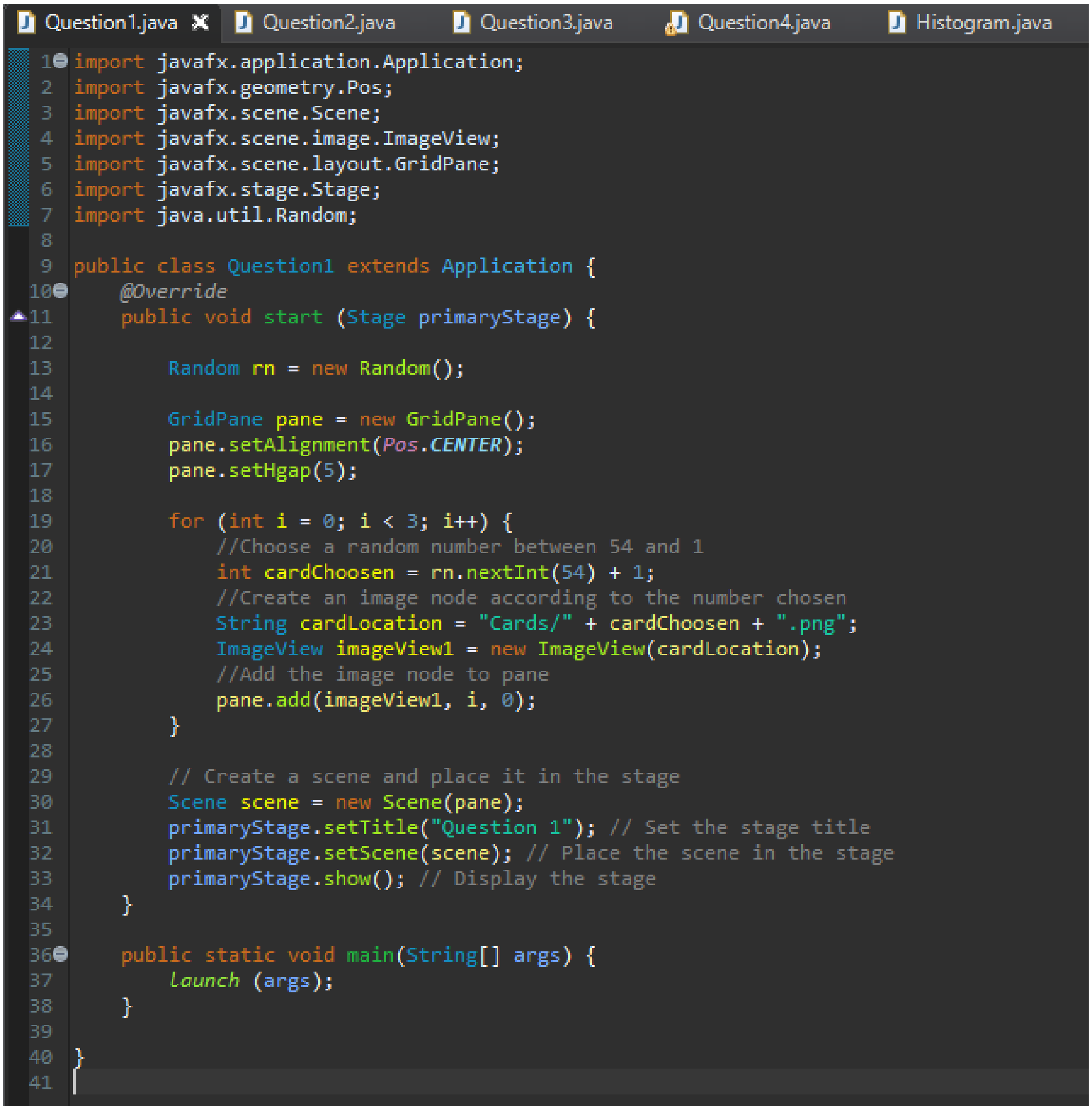
Your Code:

Copy-paste your code here:

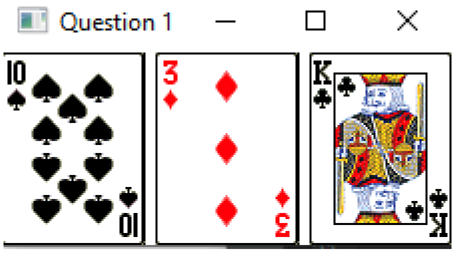
```
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.image.ImageView;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;
import java.util.Random;

public class Question1 extends Application {
        @Override
        public void start (Stage primaryStage) {

                Random rn = new Random();

                GridPane pane = new GridPane();
           pane.setAlignment(Pos.CENTER);
           pane.setHgap(5);

           for (int i = 0; i < 3; i++) {
                //Choose a random number between 54 and 1
                int cardChoosen = rn.nextInt(54) + 1;
                //Create an image node according to the number chosen
                String cardLocation = "Cards/" + cardChoosen + ".png";
                ImageView imageView1 = new ImageView(cardLocation);
                //Add the image node to pane
                pane.add(imageView1, i, 0);
            }

            // Create a scene and place it in the stage
            Scene scene = new Scene(pane);
            primaryStage.setTitle("Question 1"); // Set the stage title
            primaryStage.setScene(scene); // Place the scene in the stage
            primaryStage.show(); // Display the stage
        }

        public static void main(String[] args) {
                launch (args);
        }

}
```

Screen shots:

Include two screen shots here:



```java
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.image.ImageView;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;
import java.util.Random;

public class Question1 extends Application {
    @Override
    public void start (Stage primaryStage) {

        Random rn = new Random();

        GridPane pane = new GridPane();
        pane.setAlignment(Pos.CENTER);
        pane.setHgap(5);

        for (int i = 0; i < 3; i++) {
            //Choose a random number between 54 and 1
            int cardChoosen = rn.nextInt(54) + 1;
            //Create an image node according to the number chosen
            String cardLocation = "Cards/" + cardChoosen + ".png";
            ImageView imageView1 = new ImageView(cardLocation);
            //Add the image node to pane
            pane.add(imageView1, i, 0);
        }

        // Create a scene and place it in the stage
        Scene scene = new Scene(pane);
        primaryStage.setTitle("Question 1"); // Set the stage title
        primaryStage.setScene(scene); // Place the scene in the stage
        primaryStage.show(); // Display the stage
    }

    public static void main(String[] args) {
        launch (args);
    }

}
```

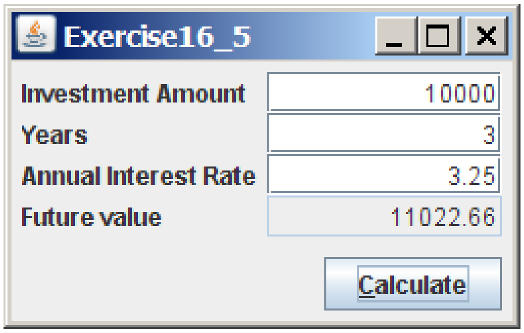


Question 2: Investment-Value calculator

Problem Description:

Write a program that calculates the future value of an investment at a given interest rate for a specified number of years. The formula for the calculation is as follows:

$$\mathtt{futureValue = investmentAmount * (1 + monthlyInterestRate)^{years*12}}$$



Investment Amount	10000
Years	3
Annual Interest Rate	3.25
Future value	11022.66

Your Task:

Use text fields for interest rate, investment amount, and years. Display the future amount in a text field when the user clicks the Calculate button, as shown in the figure.

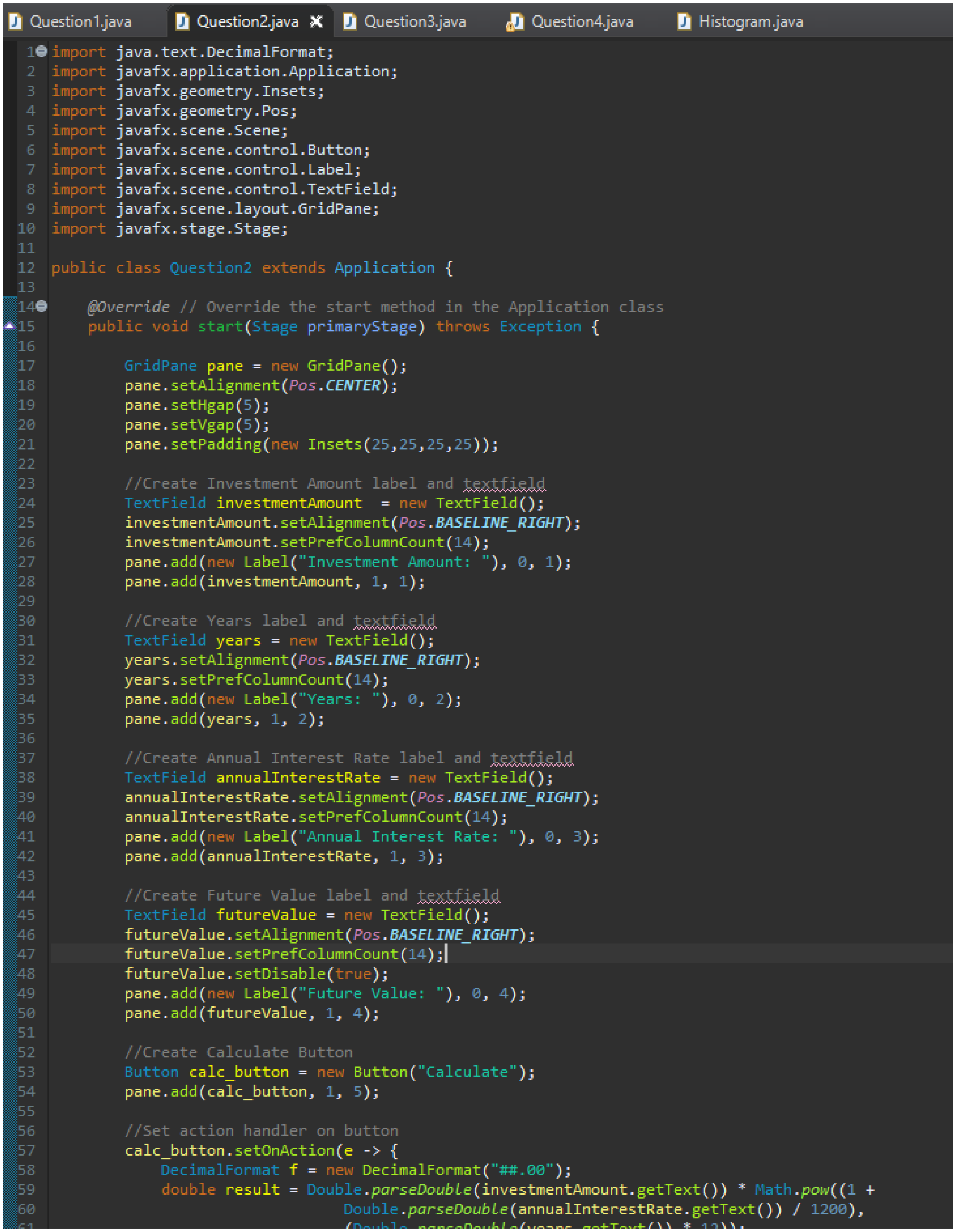
Your Code:

Copy-paste your code here.

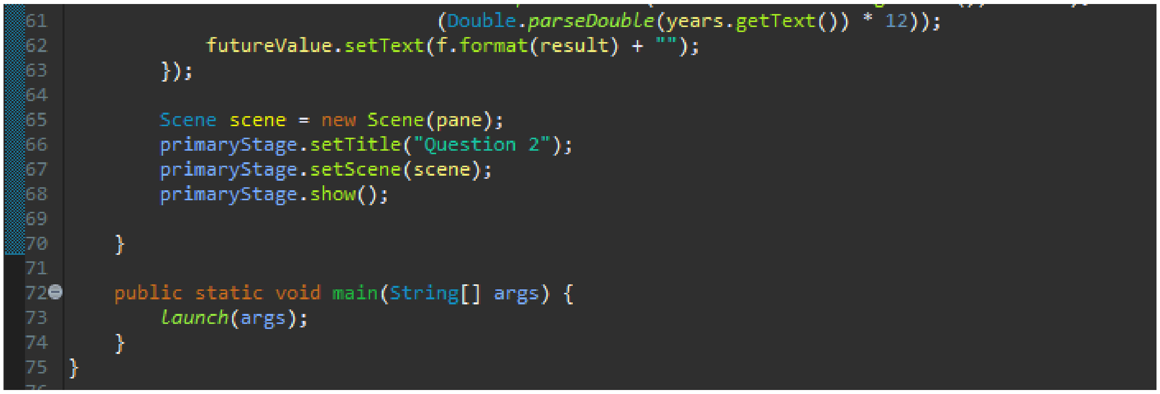
```
import java.text.DecimalFormat;
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

public class Question2 extends Application {

        @Override // Override the start method in the Application class
        public void start(Stage primaryStage) throws Exception {

                GridPane pane = new GridPane();
                pane.setAlignment(Pos.CENTER);
                pane.setHgap(5);
                pane.setVgap(5);
                pane.setPadding(new Insets(25,25,25,25));

                //Create Investment Amount label and textfield
                TextField investmentAmount  = new TextField();
                investmentAmount.setAlignment(Pos.BASELINE_RIGHT);
                investmentAmount.setPrefColumnCount(14);
                pane.add(new Label("Investment Amount: "), 0, 1);
                pane.add(investmentAmount, 1, 1);

                //Create Years label and textfield
                TextField years = new TextField();
                years.setAlignment(Pos.BASELINE_RIGHT);
                years.setPrefColumnCount(14);
                pane.add(new Label("Years: "), 0, 2);
                pane.add(years, 1, 2);

                //Create Annual Interest Rate label and textfield
                TextField annualInterestRate = new TextField();
                annualInterestRate.setAlignment(Pos.BASELINE_RIGHT);
                annualInterestRate.setPrefColumnCount(14);
                pane.add(new Label("Annual Interest Rate: "), 0, 3);
                pane.add(annualInterestRate, 1, 3);

                //Create Future Value label and textfield
                TextField futureValue = new TextField();
                futureValue.setAlignment(Pos.BASELINE_RIGHT);
                futureValue.setPrefColumnCount(14);
                futureValue.setDisable(true);
                pane.add(new Label("Future Value: "), 0, 4);
                pane.add(futureValue, 1, 4);

                //Create Calculate Button
                Button calc_button = new Button("Calculate");
                pane.add(calc_button, 1, 5);

                //Set action handler on button
                calc_button.setOnAction(e -> {
                        DecimalFormat f = new DecimalFormat("##.00");
                        double result = Double.parseDouble(investmentAmount.getText())
* Math.pow((1 +

        Double.parseDouble(annualInterestRate.getText()) / 1200),

        (Double.parseDouble(years.getText()) * 12));
                        futureValue.setText(f.format(result) + "");
                });

                Scene scene = new Scene(pane);
                primaryStage.setTitle("Question 2");
                primaryStage.setScene(scene);
                primaryStage.show();

        }

        public static void main(String[] args) {
                launch(args);
        }
}
```

Screen shots:

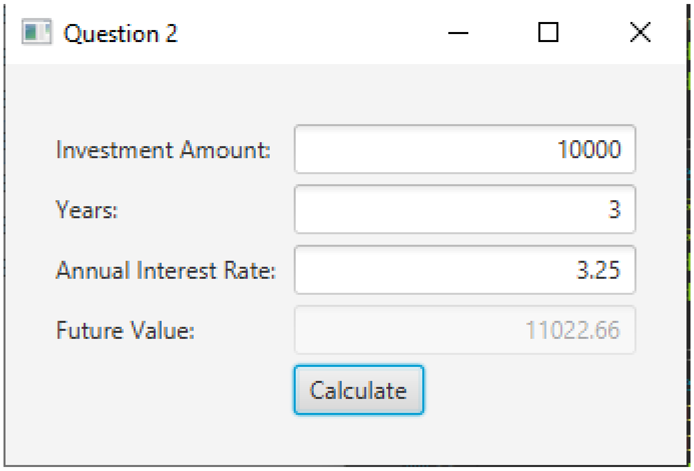
Include two screen shots here:



```java
import java.text.DecimalFormat;
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

public class Question2 extends Application {

    @Override // Override the start method in the Application class
    public void start(Stage primaryStage) throws Exception {

        GridPane pane = new GridPane();
        pane.setAlignment(Pos.CENTER);
        pane.setHgap(5);
        pane.setVgap(5);
        pane.setPadding(new Insets(25,25,25,25));

        //Create Investment Amount label and textfield
        TextField investmentAmount  = new TextField();
        investmentAmount.setAlignment(Pos.BASELINE_RIGHT);
        investmentAmount.setPrefColumnCount(14);
        pane.add(new Label("Investment Amount: "), 0, 1);
        pane.add(investmentAmount, 1, 1);

        //Create Years label and textfield
        TextField years = new TextField();
        years.setAlignment(Pos.BASELINE_RIGHT);
        years.setPrefColumnCount(14);
        pane.add(new Label("Years: "), 0, 2);
        pane.add(years, 1, 2);

        //Create Annual Interest Rate label and textfield
        TextField annualInterestRate = new TextField();
        annualInterestRate.setAlignment(Pos.BASELINE_RIGHT);
        annualInterestRate.setPrefColumnCount(14);
        pane.add(new Label("Annual Interest Rate: "), 0, 3);
        pane.add(annualInterestRate, 1, 3);

        //Create Future Value label and textfield
        TextField futureValue = new TextField();
        futureValue.setAlignment(Pos.BASELINE_RIGHT);
        futureValue.setPrefColumnCount(14);
        futureValue.setDisable(true);
        pane.add(new Label("Future Value: "), 0, 4);
        pane.add(futureValue, 1, 4);

        //Create Calculate Button
        Button calc_button = new Button("Calculate");
        pane.add(calc_button, 1, 5);

        //Set action handler on button
        calc_button.setOnAction(e -> {
            DecimalFormat f = new DecimalFormat("##.00");
            double result = Double.parseDouble(investmentAmount.getText()) * Math.pow((1 +
                                Double.parseDouble(annualInterestRate.getText()) / 1200),
                                (Double.parseDouble(years.getText()) * 12));
```



```
                                (Double.parseDouble(years.getText()) * 12));
            futureValue.setText(f.format(result) + "");
        });

        Scene scene = new Scene(pane);
        primaryStage.setTitle("Question 2");
        primaryStage.setScene(scene);
        primaryStage.show();

    }

    public static void main(String[] args) {
        launch(args);
    }
}
```



Investment Amount: 10000

Years: 3

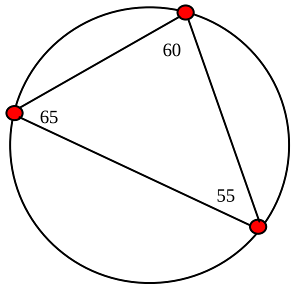
Annual Interest Rate: 3.25

Future Value: 11022.66

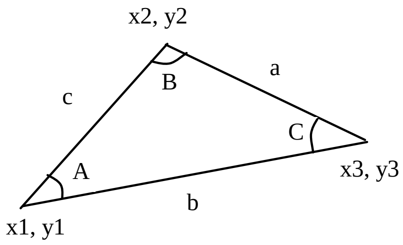
Question 3: Dragging Points on a Circle

Problem Description:

Draw a circle with three random points on the circle. Connect the points to form a triangle. Display the angles in the triangle. Use the mouse to drag a point along the perimeter of the circle. As you drag it, the triangle and angles are redisplayed dynamically.



Here is the formula to compute angles:



```
A = acos((a * a - b * b - c * c) / (-2 * b * c))
B = acos((b * b - a * a - c * c) / (-2 * a * c))
C = acos((c * c - b * b - a * a) / (-2 * a * b))
```

Your Code:

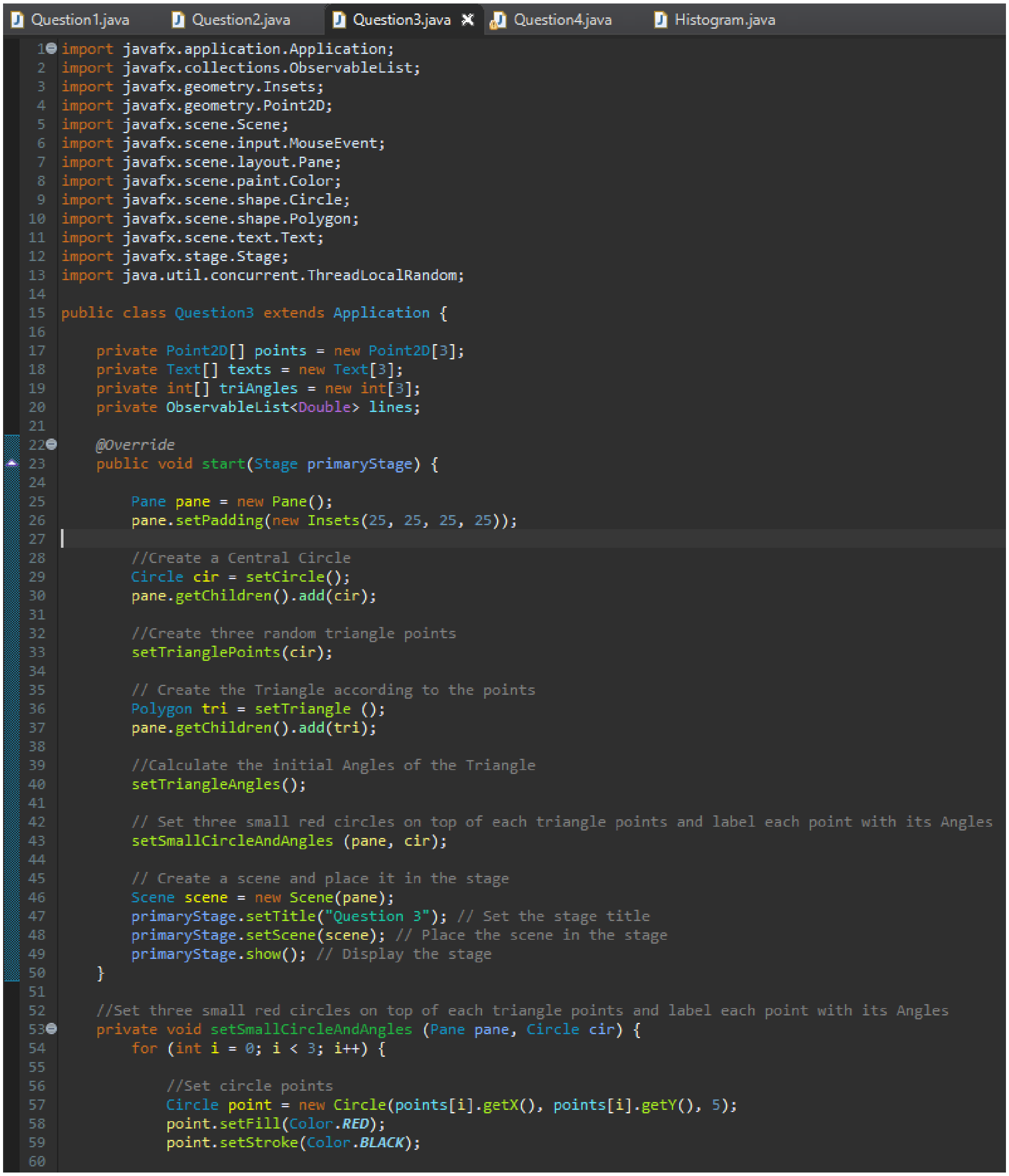
Copy-paste your code here:

```
import javafx.application.Application;
import javafx.collections.ObservableList;
import javafx.geometry.Insets;
import javafx.geometry.Point2D;
import javafx.scene.Scene;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Polygon;
import javafx.scene.text.Text;
import javafx.stage.Stage;
import java.util.concurrent.ThreadLocalRandom;
```

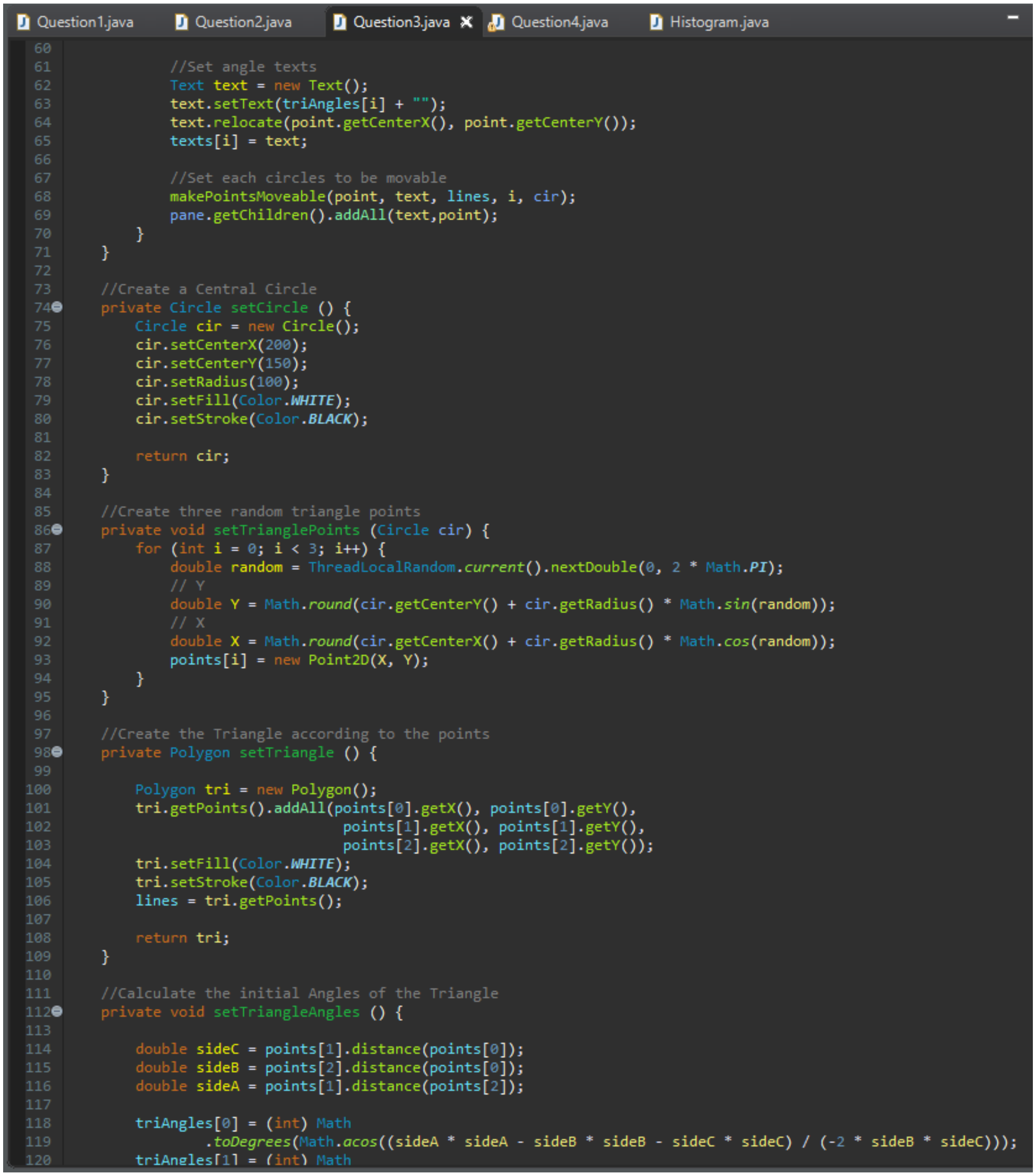
```
public class Question3 extends Application {

    private Point2D[] points = new Point2D[3];
    private Text[] texts = new Text[3];
    private int[] triAngles = new int[3];
    private ObservableList<Double> lines;

    @Override
    public void start(Stage primaryStage) {

        Pane pane = new Pane();
        pane.setPadding(new Insets(25, 25, 25, 25));

        //Create a Central Circle
        Circle cir = setCircle();
        pane.getChildren().add(cir);

        //Create three random triangle points
        setTrianglePoints(cir);

        // Create the Triangle according to the points
        Polygon tri = setTriangle ();
        pane.getChildren().add(tri);

        //Calculate the initial Angles of the Triangle
        setTriangleAngles();

        // Set three small red circles on top of each triangle points and label each
point with its Angles
        setSmallCircleAndAngles (pane, cir);

        // Create a scene and place it in the stage
        Scene scene = new Scene(pane);
        primaryStage.setTitle("Question 3"); // Set the stage title
        primaryStage.setScene(scene); // Place the scene in the stage
        primaryStage.show(); // Display the stage
    }

    //Set three small red circles on top of each triangle points and label each point
with its Angles
    private void setSmallCircleAndAngles (Pane pane, Circle cir) {
        for (int i = 0; i < 3; i++) {

            //Set circle points
            Circle point = new Circle(points[i].getX(), points[i].getY(), 5);
            point.setFill(Color.RED);
            point.setStroke(Color.BLACK);

            //Set angle texts
            Text text = new Text();
            text.setText(triAngles[i] + "");
            text.relocate(point.getCenterX(), point.getCenterY());
            texts[i] = text;

            //Set each circles to be movable
            makePointsMoveable(point, text, lines, i, cir);
            pane.getChildren().addAll(text,point);
        }
    }

    //Create a Central Circle
    private Circle setCircle () {
        Circle cir = new Circle();
        cir.setCenterX(200);
        cir.setCenterY(150);
        cir.setRadius(100);
        cir.setFill(Color.WHITE);
        cir.setStroke(Color.BLACK);

        return cir;
    }

    //Create three random triangle points
    private void setTrianglePoints (Circle cir) {
        for (int i = 0; i < 3; i++) {
            double random = ThreadLocalRandom.current().nextDouble(0, 2 *
Math.PI);
            // Y
            double Y = Math.round(cir.getCenterY() + cir.getRadius() *
Math.sin(random));
            // X
            double X = Math.round(cir.getCenterX() + cir.getRadius() *
Math.cos(random));
            points[i] = new Point2D(X, Y);
        }
    }

    //Create the Triangle according to the points
    private Polygon setTriangle () {

        Polygon tri = new Polygon();
        tri.getPoints().addAll(points[0].getX(), points[0].getY(),
                                                                points[1].getX(),
points[1].getY(),
                                                                points[2].getX(),
points[2].getY());
                tri.setFill(Color.WHITE);
                tri.setStroke(Color.BLACK);
                lines = tri.getPoints();

                return tri;
        }

        //Calculate the initial Angles of the Triangle
        private void setTriangleAngles () {

                double sideC = points[1].distance(points[0]);
                double sideB = points[2].distance(points[0]);
                double sideA = points[1].distance(points[2]);

                triAngles[0] = (int) Math
                                .toDegrees(Math.acos((sideA * sideA - sideB * sideB -
sideC * sideC) / (-2 * sideB * sideC)));
                triAngles[1] = (int) Math
                                .toDegrees(Math.acos((sideB * sideB - sideA * sideA -
sideC * sideC) / (-2 * sideA * sideC)));
                triAngles[2] = (int) Math
                                .toDegrees(Math.acos((sideC * sideC - sideB * sideB -
sideA * sideA) / (-2 * sideA * sideB)));
        }

        //Update each angles when a point is move
        private void updateAngles () {

                double sideC = points[1].distance(points[0]);
                double sideB = points[2].distance(points[0]);
                double sideA = points[1].distance(points[2]);

                triAngles[0] = (int) Math
                                .toDegrees(Math.acos((sideA * sideA - sideB * sideB -
sideC * sideC) / (-2 * sideB * sideC)));
                triAngles[1] = (int) Math
                                .toDegrees(Math.acos((sideB * sideB - sideA * sideA -
sideC * sideC) / (-2 * sideA * sideC)));
                triAngles[2] = (int) Math
                                .toDegrees(Math.acos((sideC * sideC - sideB * sideB -
sideA * sideA) / (-2 * sideA * sideB)));

                texts[0].setText(triAngles[0] + "");
        texts[1].setText(triAngles[1] + "");
        texts[2].setText(triAngles[2] + "");

    }

    //Make each points movable
    private void makePointsMoveable(Circle point, Text text,
ObservableList<Double> line, int index, Circle cir) {
        point.setOnMouseDragged((MouseEvent me) -> {

            Point2D center = new Point2D(cir.getCenterX(), cir.getCenterY());
            Point2D mouse = new Point2D(me.getX(), me.getY());
            Point2D centerToMouse = mouse.subtract(center);
            Point2D centerToNewPoint =
centerToMouse.normalize().multiply(cir.getRadius());
            Point2D newPoint = centerToNewPoint.add(center);
            Point2D newAnglePoint =
centerToNewPoint.add(center).subtract(20, 20);

            points[index] = newPoint;
            point.setCenterX(newPoint.getX());
            point.setCenterY(newPoint.getY());

            line.set(index * 2, newPoint.getX());
            line.set(index * 2 + 1, newPoint.getY());

            updateAngles();

            text.relocate(newAnglePoint.getX(), newAnglePoint.getY());
        });
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```

Screen shots:

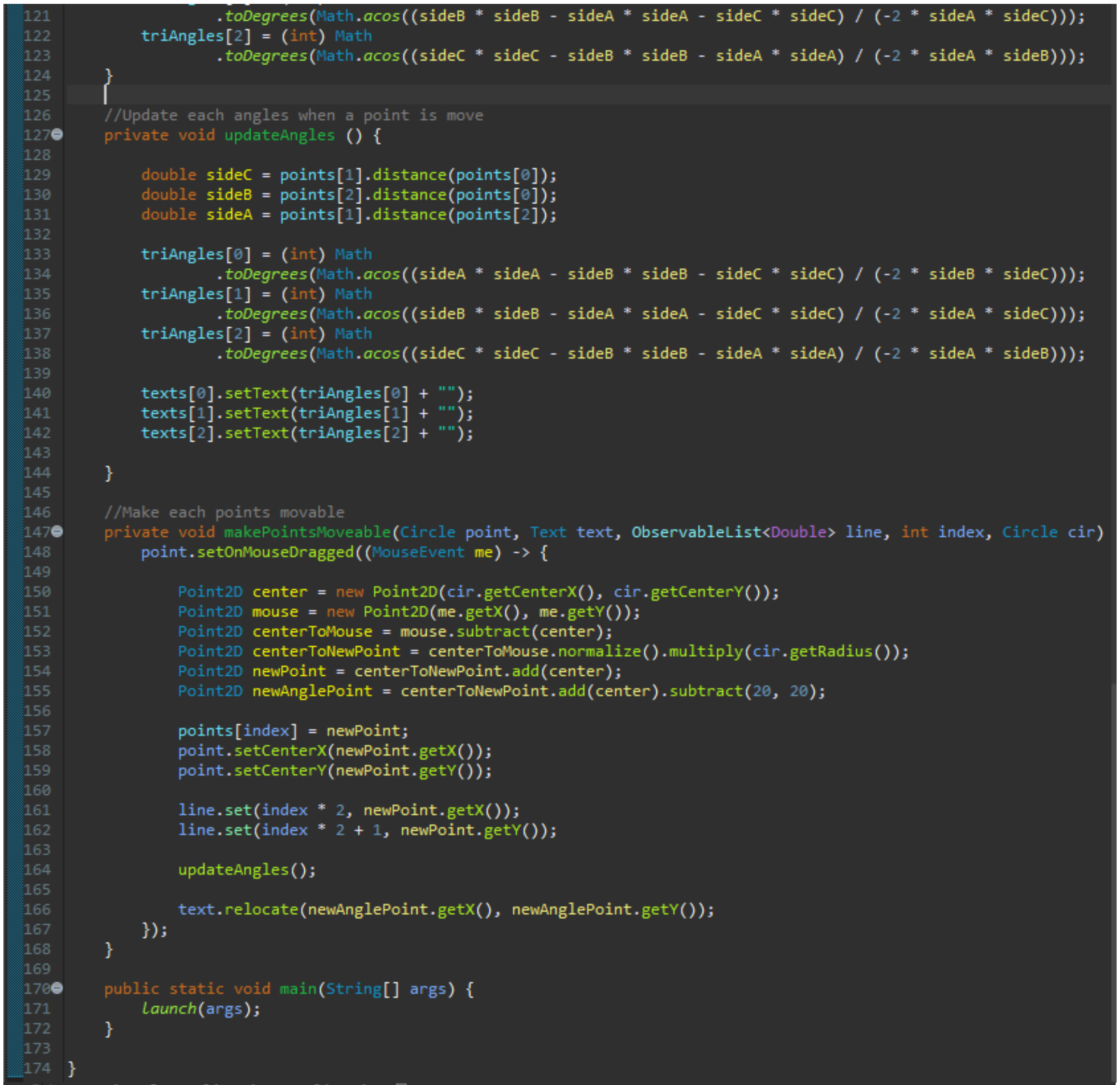
Include two screen shots here:



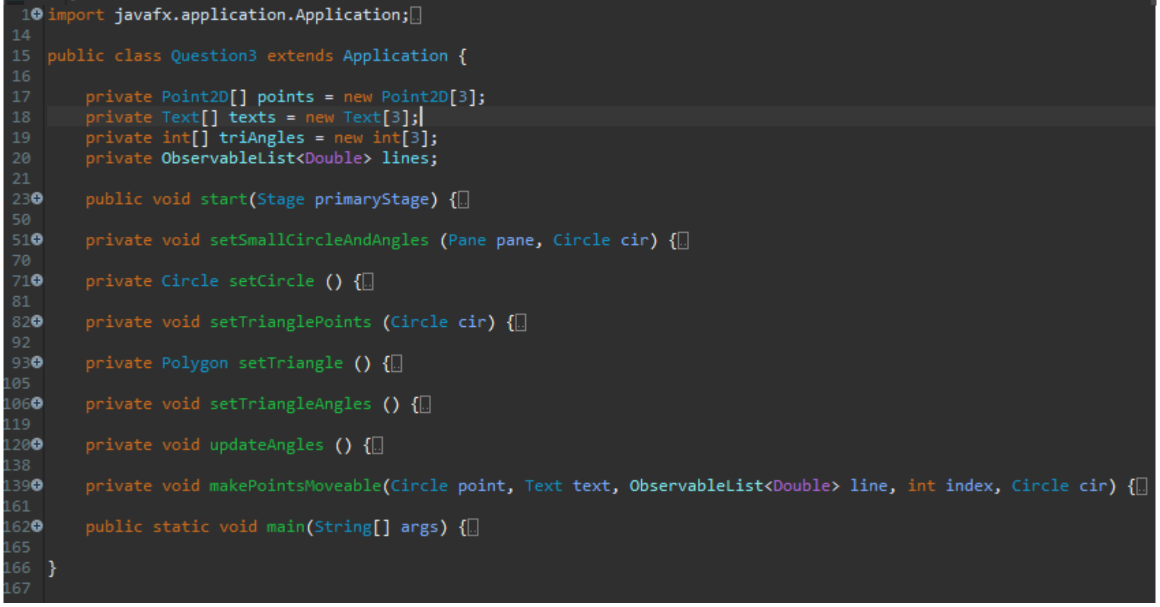
```java
import javafx.application.Application;
import javafx.collections.ObservableList;
import javafx.geometry.Insets;
import javafx.geometry.Point2D;
import javafx.scene.Scene;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Polygon;
import javafx.scene.text.Text;
import javafx.stage.Stage;
import java.util.concurrent.ThreadLocalRandom;

public class Question3 extends Application {

    private Point2D[] points = new Point2D[3];
    private Text[] texts = new Text[3];
    private int[] triAngles = new int[3];
    private ObservableList<Double> lines;

    @Override
    public void start(Stage primaryStage) {

        Pane pane = new Pane();
        pane.setPadding(new Insets(25, 25, 25, 25));

        //Create a Central Circle
        Circle cir = setCircle();
        pane.getChildren().add(cir);

        //Create three random triangle points
        setTrianglePoints(cir);

        // Create the Triangle according to the points
        Polygon tri = setTriangle ();
        pane.getChildren().add(tri);

        //Calculate the initial Angles of the Triangle
        setTriangleAngles();

        // Set three small red circles on top of each triangle points and label each point with its Angles
        setSmallCircleAndAngles (pane, cir);

        // Create a scene and place it in the stage
        Scene scene = new Scene(pane);
        primaryStage.setTitle("Question 3"); // Set the stage title
        primaryStage.setScene(scene); // Place the scene in the stage
        primaryStage.show(); // Display the stage
    }

    //Set three small red circles on top of each triangle points and label each point with its Angles
    private void setSmallCircleAndAngles (Pane pane, Circle cir) {
        for (int i = 0; i < 3; i++) {

            //Set circle points
            Circle point = new Circle(points[i].getX(), points[i].getY(), 5);
            point.setFill(Color.RED);
            point.setStroke(Color.BLACK);

```



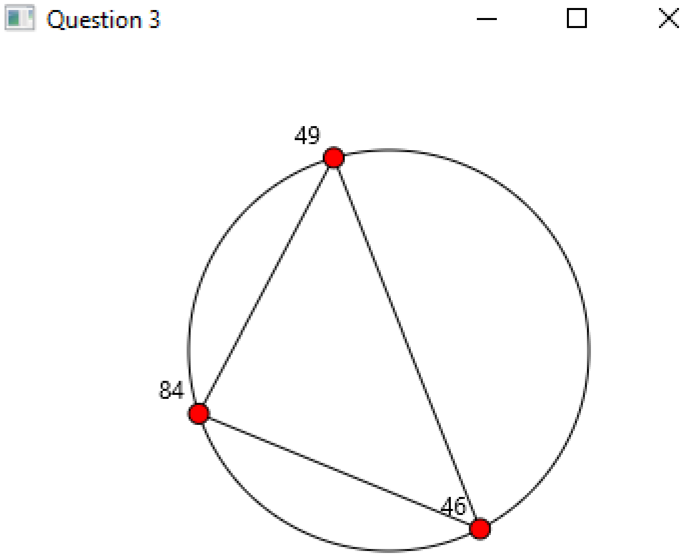
```java

            //Set angle texts
            Text text = new Text();
            text.setText(triAngles[i] + "");
            text.relocate(point.getCenterX(), point.getCenterY());
            texts[i] = text;

            //Set each circles to be movable
            makePointsMoveable(point, text, lines, i, cir);
            pane.getChildren().addAll(text,point);
        }
    }

    //Create a Central Circle
    private Circle setCircle () {
        Circle cir = new Circle();
        cir.setCenterX(200);
        cir.setCenterY(150);
        cir.setRadius(100);
        cir.setFill(Color.WHITE);
        cir.setStroke(Color.BLACK);

        return cir;
    }

    //Create three random triangle points
    private void setTrianglePoints (Circle cir) {
        for (int i = 0; i < 3; i++) {
            double random = ThreadLocalRandom.current().nextDouble(0, 2 * Math.PI);
            // Y
            double Y = Math.round(cir.getCenterY() + cir.getRadius() * Math.sin(random));
            // X
            double X = Math.round(cir.getCenterX() + cir.getRadius() * Math.cos(random));
            points[i] = new Point2D(X, Y);
        }
    }

    //Create the Triangle according to the points
    private Polygon setTriangle () {

        Polygon tri = new Polygon();
        tri.getPoints().addAll(points[0].getX(), points[0].getY(),
                            points[1].getX(), points[1].getY(),
                            points[2].getX(), points[2].getY());
        tri.setFill(Color.WHITE);
        tri.setStroke(Color.BLACK);
        lines = tri.getPoints();

        return tri;
    }

    //Calculate the initial Angles of the Triangle
    private void setTriangleAngles () {

        double sideC = points[1].distance(points[0]);
        double sideB = points[2].distance(points[0]);
        double sideA = points[1].distance(points[2]);

        triAngles[0] = (int) Math
                .toDegrees(Math.acos((sideA * sideA - sideB * sideB - sideC * sideC) / (-2 * sideB * sideC)));
        triAngles[1] = (int) Math
```



```
                .toDegrees(Math.acos((sideB * sideB - sideA * sideA - sideC * sideC) / (-2 * sideA * sideC)));
        triAngles[2] = (int) Math
                .toDegrees(Math.acos((sideC * sideC - sideB * sideB - sideA * sideA) / (-2 * sideA * sideB)));
    }

    //Update each angles when a point is move
    private void updateAngles () {

        double sideC = points[1].distance(points[0]);
        double sideB = points[2].distance(points[0]);
        double sideA = points[1].distance(points[2]);

        triAngles[0] = (int) Math
                .toDegrees(Math.acos((sideA * sideA - sideB * sideB - sideC * sideC) / (-2 * sideB * sideC)));
        triAngles[1] = (int) Math
                .toDegrees(Math.acos((sideB * sideB - sideA * sideA - sideC * sideC) / (-2 * sideA * sideC)));
        triAngles[2] = (int) Math
                .toDegrees(Math.acos((sideC * sideC - sideB * sideB - sideA * sideA) / (-2 * sideA * sideB)));

        texts[0].setText(triAngles[0] + "");
        texts[1].setText(triAngles[1] + "");
        texts[2].setText(triAngles[2] + "");

    }

    //Make each points movable
    private void makePointsMoveable(Circle point, Text text, ObservableList<Double> line, int index, Circle cir)
        point.setOnMouseDragged((MouseEvent me) -> {

            Point2D center = new Point2D(cir.getCenterX(), cir.getCenterY());
            Point2D mouse = new Point2D(me.getX(), me.getY());
            Point2D centerToMouse = mouse.subtract(center);
            Point2D centerToNewPoint = centerToMouse.normalize().multiply(cir.getRadius());
            Point2D newPoint = centerToNewPoint.add(center);
            Point2D newAnglePoint = centerToNewPoint.add(center).subtract(20, 20);

            points[index] = newPoint;
            point.setCenterX(newPoint.getX());
            point.setCenterY(newPoint.getY());

            line.set(index * 2, newPoint.getX());
            line.set(index * 2 + 1, newPoint.getY());

            updateAngles();

            text.relocate(newAnglePoint.getX(), newAnglePoint.getY());
        });
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```



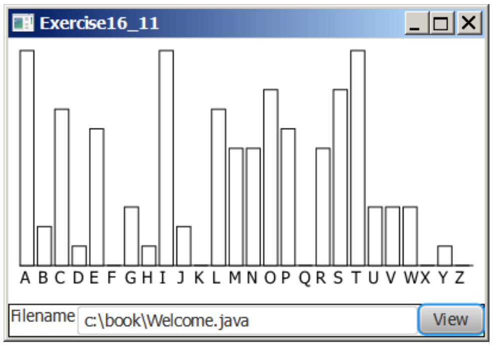
```
import javafx.application.Application;□

public class Question3 extends Application {

    private Point2D[] points = new Point2D[3];
    private Text[] texts = new Text[3];
    private int[] triAngles = new int[3];
    private ObservableList<Double> lines;

    public void start(Stage primaryStage) {□

    private void setSmallCircleAndAngles (Pane pane, Circle cir) {□

    private Circle setCircle () {□

    private void setTrianglePoints (Circle cir) {□

    private Polygon setTriangle () {□

    private void setTriangleAngles () {□

    private void updateAngles () {□

    private void makePointsMoveable(Circle point, Text text, ObservableList<Double> line, int index, Circle cir) {□

    public static void main(String[] args) {□

}
```


Question 3
49
84
46

Question 4: Histogram

Problem Description:

Develop a program that displays a histogram to show the occurrences of each letter in a text area. The histogram should show the occurrences of each letter in a text file, as shown in the following figure. Assume that the letters are not case sensitive.



Filename c:\book\Welcome.java

Your Task:

- Place a pane that will display the histogram in the center of the frame.
- Place a label and a text field in a panel, and put the panel in the south side of the frame. The text file will be entered from this text field.
- Pressing the Enter key on the text field causes the program to count the occurrences of each letter and display the count in a histogram.

Your Code:

Copy-paste your code here:

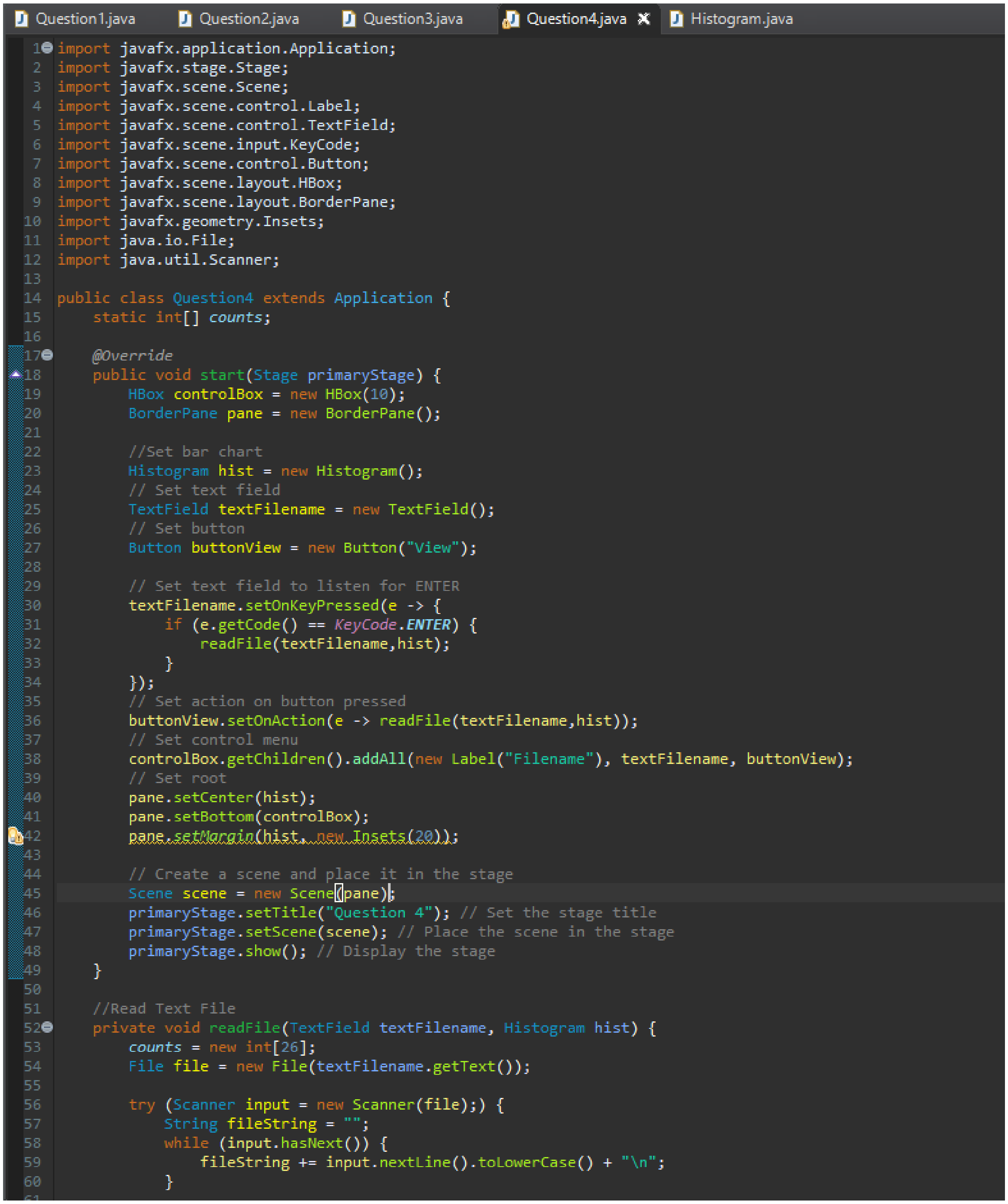
```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyCode;
import javafx.scene.control.Button;
import javafx.scene.layout.HBox;
import javafx.scene.layout.BorderPane;
import javafx.geometry.Insets;
import java.io.File;
import java.util.Scanner;

public class Question4 extends Application {
        static int[] counts;
```

```
    @Override
    public void start(Stage primaryStage) {
        HBox controlBox = new HBox(10);
        BorderPane pane = new BorderPane();

        //Set bar chart
        Histogram hist = new Histogram();
        // Set text field
        TextField textFilename = new TextField();
        // Set button
        Button buttonView = new Button("View");

        // Set text field to listen for ENTER
        textFilename.setOnKeyPressed(e -> {
            if (e.getCode() == KeyCode.ENTER) {
                readFile(textFilename,hist);
            }
        });
        // Set action on button pressed
        buttonView.setOnAction(e -> readFile(textFilename,hist));
        // Set control menu
        controlBox.getChildren().addAll(new Label("Filename"), textFilename,
buttonView);
        // Set root
        pane.setCenter(hist);
        pane.setBottom(controlBox);
        pane.setMargin(hist, new Insets(20));

        // Create a scene and place it in the stage
        Scene scene = new Scene(pane);
        primaryStage.setTitle("Question 4"); // Set the stage title
        primaryStage.setScene(scene); // Place the scene in the stage
        primaryStage.show(); // Display the stage
    }

    //Read Text File
    private void readFile(TextField textFilename, Histogram hist) {
        counts = new int[26];
        File file = new File(textFilename.getText());

        try (Scanner input = new Scanner(file);) {
            String fileString = "";
            while (input.hasNext()) {
                fileString += input.nextLine().toLowerCase() + "\n";
            }
            countLetters(fileString);
            hist.setCounts(counts);
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

    //Count the letters in text file
    public static void countLetters(String s) {
        for (int i = 0; i < s.length(); i++) {
            char character = s.charAt(i);

            if (Character.isLetter(character)) {
                counts[character - 97]++;
            }
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

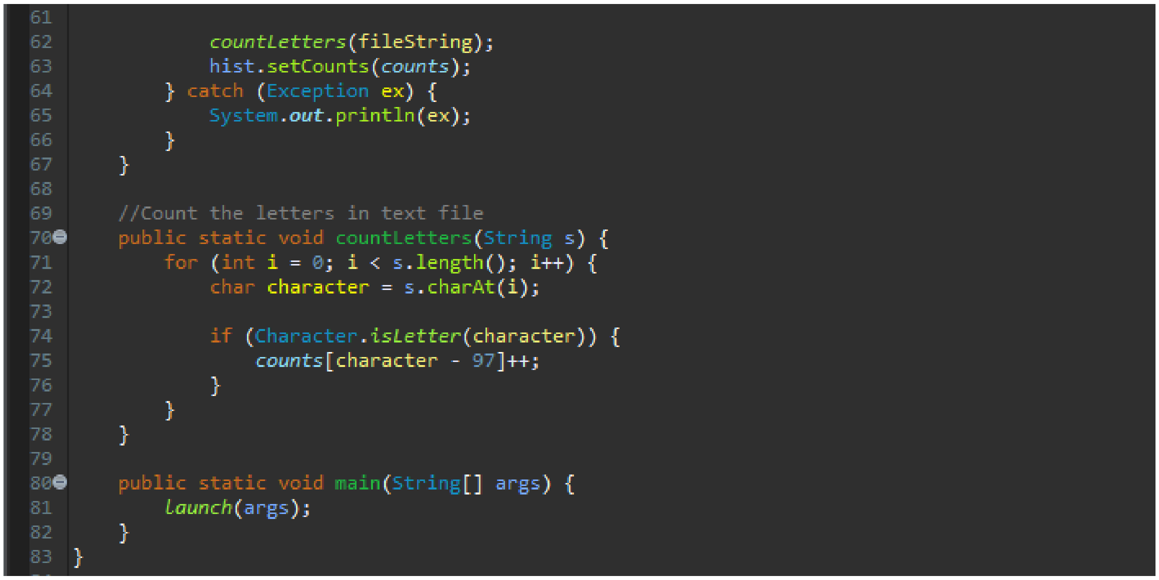
```
import javafx.scene.layout.Pane;
import javafx.scene.shape.Rectangle;
import javafx.scene.text.Text;
import javafx.scene.paint.Color;

public class Histogram extends Pane {
    private int[] counts;
    private int maxValue;
    private int paneHeight;

    public Histogram() {
        this.counts = new int[26];
        maxValue = 0;
        paneHeight = 400;
        drawHistogram();
    }

    //Create the bar chart
    public void drawHistogram() {
        for (int i = 0, x = 0; i < counts.length; i++, x += 25) {
            double rHeight = (double) counts[i] / maxValue * paneHeight;

            Rectangle r = new Rectangle(20, rHeight);
            r.setX(x);
            r.setY(paneHeight - rHeight);
            r.setFill(Color.WHITE);
            r.setStroke(Color.BLACK);

            Text t = new Text((char) (i + 65) + "");
            t.setX(x + 5);
            t.setY(paneHeight + 15);
            getChildren().addAll(r, t);
        }
    }

    //Set the count of letters
    public void setCounts(int[] counts) {
        this.counts = counts;
        maxValue = getMaxValue();
        drawHistogram();
    }

    //Get the highest number in count
    public int getMaxValue() {
        int m = counts[0];
        for (int i = 1; i < counts.length; i++) {
            if (counts[i] > m) {
                m = counts[i];
            }
        }
        return m;
    }
}
```

Screen shots:

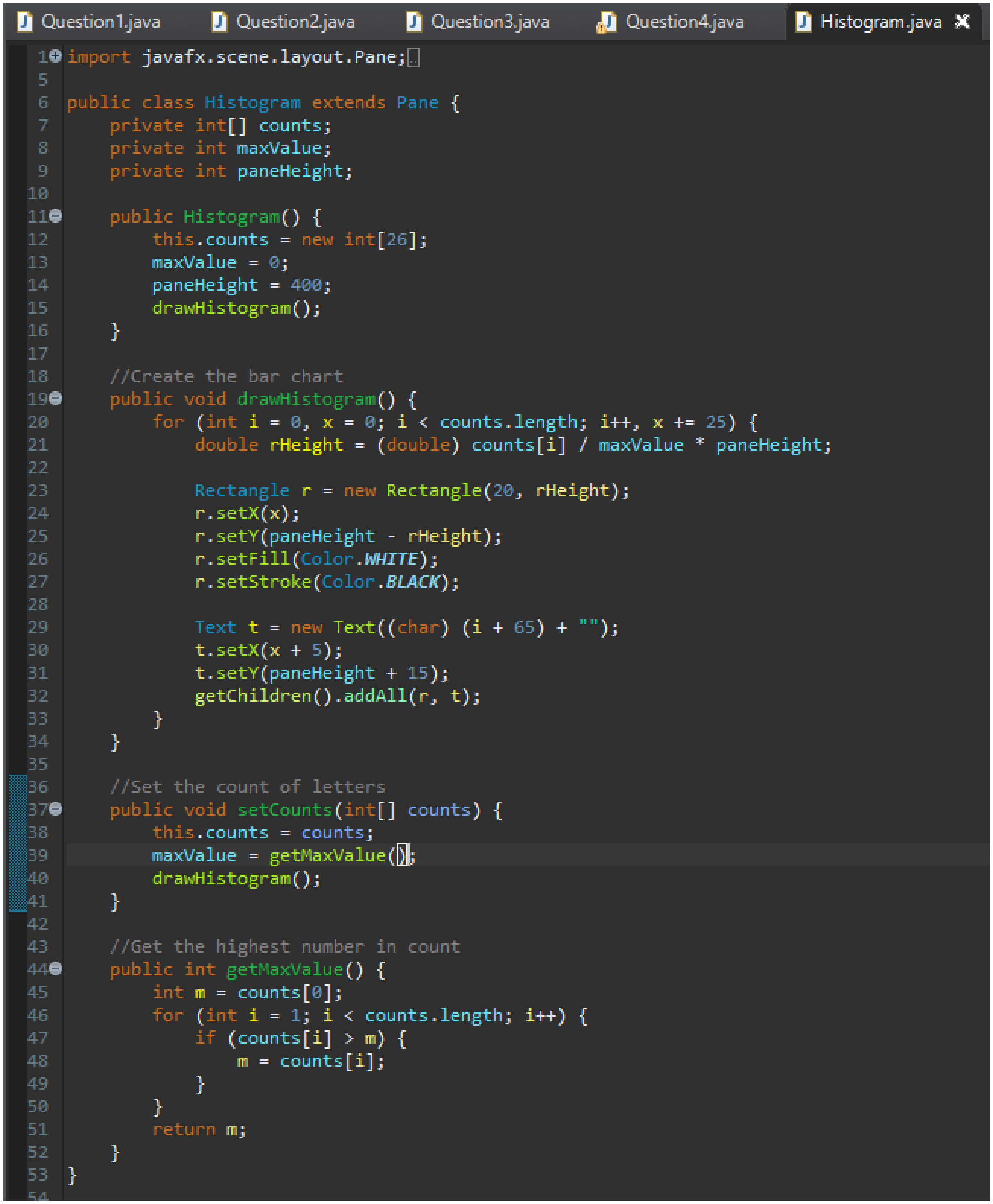
Include two screen shots here:



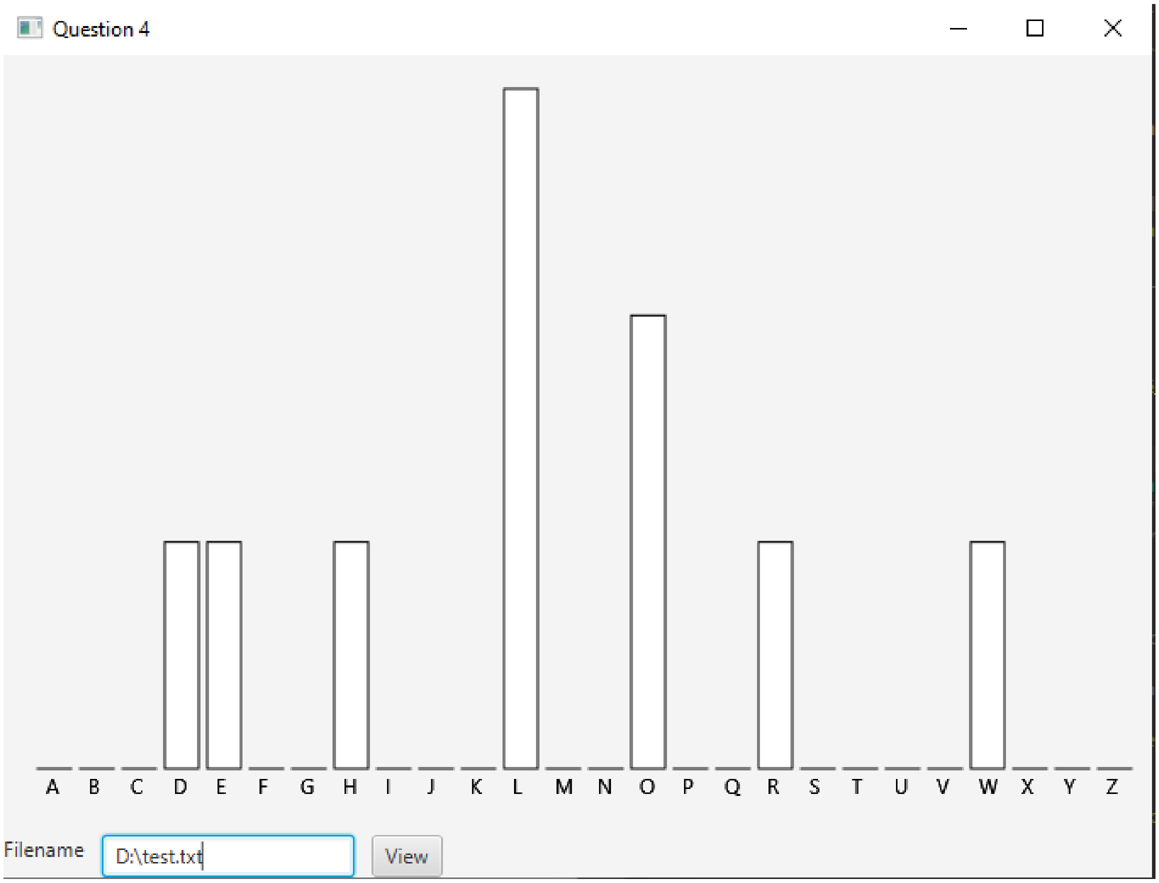
```java
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyCode;
import javafx.scene.control.Button;
import javafx.scene.layout.HBox;
import javafx.scene.layout.BorderPane;
import javafx.geometry.Insets;
import java.io.File;
import java.util.Scanner;

public class Question4 extends Application {
    static int[] counts;

    @Override
    public void start(Stage primaryStage) {
        HBox controlBox = new HBox(10);
        BorderPane pane = new BorderPane();

        //Set bar chart
        Histogram hist = new Histogram();
        // Set text field
        TextField textFilename = new TextField();
        // Set button
        Button buttonView = new Button("View");

        // Set text field to listen for ENTER
        textFilename.setOnKeyPressed(e -> {
            if (e.getCode() == KeyCode.ENTER) {
                readFile(textFilename,hist);
            }
        });
        // Set action on button pressed
        buttonView.setOnAction(e -> readFile(textFilename,hist));
        // Set control menu
        controlBox.getChildren().addAll(new Label("Filename"), textFilename, buttonView);
        // Set root
        pane.setCenter(hist);
        pane.setBottom(controlBox);
        pane.setMargin(hist, new Insets(20));

        // Create a scene and place it in the stage
        Scene scene = new Scene(pane);
        primaryStage.setTitle("Question 4"); // Set the stage title
        primaryStage.setScene(scene); // Place the scene in the stage
        primaryStage.show(); // Display the stage
    }

    //Read Text File
    private void readFile(TextField textFilename, Histogram hist) {
        counts = new int[26];
        File file = new File(textFilename.getText());

        try (Scanner input = new Scanner(file);) {
            String fileString = "";
            while (input.hasNext()) {
                fileString += input.nextLine().toLowerCase() + "\n";
            }
```



```java

            countLetters(fileString);
            hist.setCounts(counts);
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

    //Count the letters in text file
    public static void countLetters(String s) {
        for (int i = 0; i < s.length(); i++) {
            char character = s.charAt(i);

            if (Character.isLetter(character)) {
                counts[character - 97]++;
            }
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```



```java
import javafx.scene.layout.Pane;

public class Histogram extends Pane {
    private int[] counts;
    private int maxValue;
    private int paneHeight;

    public Histogram() {
        this.counts = new int[26];
        maxValue = 0;
        paneHeight = 400;
        drawHistogram();
    }

    //Create the bar chart
    public void drawHistogram() {
        for (int i = 0, x = 0; i < counts.length; i++, x += 25) {
            double rHeight = (double) counts[i] / maxValue * paneHeight;

            Rectangle r = new Rectangle(20, rHeight);
            r.setX(x);
            r.setY(paneHeight - rHeight);
            r.setFill(Color.WHITE);
            r.setStroke(Color.BLACK);

            Text t = new Text((char) (i + 65) + "");
            t.setX(x + 5);
            t.setY(paneHeight + 15);
            getChildren().addAll(r, t);
        }
    }

    //Set the count of letters
    public void setCounts(int[] counts) {
        this.counts = counts;
        maxValue = getMaxValue();
        drawHistogram();
    }

    //Get the highest number in count
    public int getMaxValue() {
        int m = counts[0];
        for (int i = 1; i < counts.length; i++) {
            if (counts[i] > m) {
                m = counts[i];
            }
        }
        return m;
    }
}
```



Question 4
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
Filename
D:\test.txt
View

Remember:

You need to complete this file and submit it in related **drop box on Blackboard**, in addition to uploading your codes in your **git repository**, before deadline.