



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 cardChosen = 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:

```
Question1.java X Question2.java Question3.java Question4.java Histogram.java
1 import javafx.application.Application;
2 import javafx.geometry.Pos;
3 import javafx.scene.Scene;
4 import javafx.scene.image.ImageView;
5 import javafx.scene.layout.GridPane;
6 import javafx.stage.Stage;
7 import java.util.Random;
8
9 public class Question1 extends Application {
10     @Override
11     public void start (Stage primaryStage) {
12
13         Random rn = new Random();
14
15         GridPane pane = new GridPane();
16         pane.setAlignment(Pos.CENTER);
17         pane.setHgap(5);
18
19         for (int i = 0; i < 3; i++) {
20             //Choose a random number between 54 and 1
21             int cardChosen = rn.nextInt(54) + 1;
22             //Create an image node according to the number chosen
23             String cardLocation = "Cards/" + cardChosen + ".png";
24             ImageView imageView1 = new ImageView(cardLocation);
25             //Add the image node to pane
26             pane.add(imageView1, i, 0);
27         }
28
29         // Create a scene and place it in the stage
30         Scene scene = new Scene(pane);
31         primaryStage.setTitle("Question 1"); // Set the stage title
32         primaryStage.setScene(scene); // Place the scene in the stage
33         primaryStage.show(); // Display the stage
34     }
35
36     public static void main(String[] args) {
37         launch (args);
38     }
39
40 }
41
```

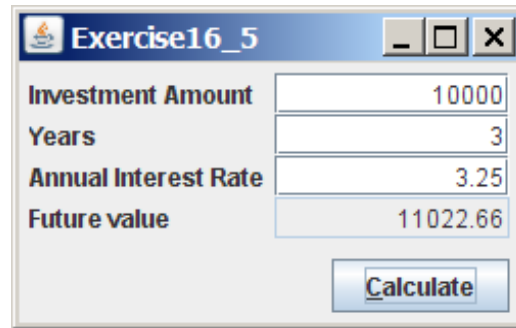


Question 2: Investment-Value calculator

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:

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



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:

```

Question1.java Question2.java X Question3.java Question4.java Histogram.java
1 import java.text.DecimalFormat;
2 import javafx.application.Application;
3 import javafx.geometry.Insets;
4 import javafx.geometry.Pos;
5 import javafx.scene.Scene;
6 import javafx.scene.control.Button;
7 import javafx.scene.control.Label;
8 import javafx.scene.control.TextField;
9 import javafx.scene.layout.GridPane;
10 import javafx.stage.Stage;
11
12 public class Question2 extends Application {
13
14     @Override // Override the start method in the Application class
15     public void start(Stage primaryStage) throws Exception {
16
17         GridPane pane = new GridPane();
18         pane.setAlignment(Pos.CENTER);
19         pane.setHgap(5);
20         pane.setVgap(5);
21         pane.setPadding(new Insets(25,25,25,25));
22
23         //Create Investment Amount label and textfield
24         TextField investmentAmount = new TextField();
25         investmentAmount.setAlignment(Pos.BASELINE_RIGHT);
26         investmentAmount.setPrefColumnCount(14);
27         pane.add(new Label("Investment Amount: "), 0, 1);
28         pane.add(investmentAmount, 1, 1);
29
30         //Create Years label and textfield
31         TextField years = new TextField();
32         years.setAlignment(Pos.BASELINE_RIGHT);
33         years.setPrefColumnCount(14);
34         pane.add(new Label("Years: "), 0, 2);
35         pane.add(years, 1, 2);
36
37         //Create Annual Interest Rate label and textfield
38         TextField annualInterestRate = new TextField();
39         annualInterestRate.setAlignment(Pos.BASELINE_RIGHT);
40         annualInterestRate.setPrefColumnCount(14);
41         pane.add(new Label("Annual Interest Rate: "), 0, 3);
42         pane.add(annualInterestRate, 1, 3);
43
44         //Create Future Value label and textfield
45         TextField futureValue = new TextField();
46         futureValue.setAlignment(Pos.BASELINE_RIGHT);
47         futureValue.setPrefColumnCount(14);
48         futureValue.setDisable(true);
49         pane.add(new Label("Future Value: "), 0, 4);
50         pane.add(futureValue, 1, 4);
51
52         //Create Calculate Button
53         Button calc_button = new Button("Calculate");
54         pane.add(calc_button, 1, 5);
55
56         //Set action handler on button
57         calc_button.setOnAction(e -> {
58             DecimalFormat f = new DecimalFormat("###.00");
59             double result = Double.parseDouble(investmentAmount.getText()) * Math.pow((1 +
60                 Double.parseDouble(annualInterestRate.getText()) / 1200),
61                 (Double.parseDouble(years.getText()) * 12));
62         });
63     }
64 }

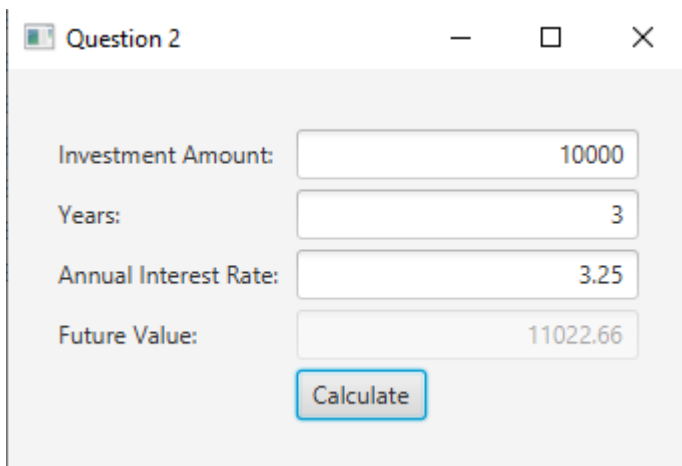
```



```

61         (Double.parseDouble(years.getText()) * 12));
62         futureValue.setText(f.format(result) + "");
63     });
64
65     Scene scene = new Scene(pane);
66     primaryStage.setTitle("Question 2");
67     primaryStage.setScene(scene);
68     primaryStage.show();
69
70 }
71
72 public static void main(String[] args) {
73     launch(args);
74 }
75 }

```



Question 2

Investment Amount: 10000

Years: 3

Annual Interest Rate: 3.25

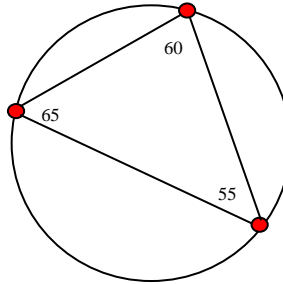
Future Value: 11022.66

Calculate

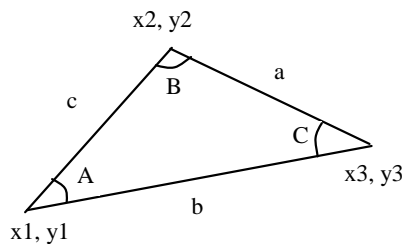
Question 3: Dragging Points on a Circle

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:



$$\begin{aligned} A &= \arccos((a * a - b * b - c * c) / (-2 * b * c)) \\ B &= \arccos((b * b - a * a - c * c) / (-2 * a * c)) \\ C &= \arccos((c * c - b * b - a * a) / (-2 * a * b)) \end{aligned}$$

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].getY(),
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:

```

Question1.java Question2.java Question3.java Question4.java Histogram.java
1 import javafx.application.Application;
2 import javafx.collections.ObservableList;
3 import javafx.geometry.Insets;
4 import javafx.geometry.Point2D;
5 import javafx.scene.Scene;
6 import javafx.scene.input.MouseEvent;
7 import javafx.scene.layout.Pane;
8 import javafx.scene.paint.Color;
9 import javafx.scene.shape.Circle;
10 import javafx.scene.shape.Polygon;
11 import javafx.scene.text.Text;
12 import javafx.stage.Stage;
13 import java.util.concurrent.ThreadLocalRandom;
14
15 public class Question3 extends Application {
16
17     private Point2D[] points = new Point2D[3];
18     private Text[] texts = new Text[3];
19     private int[] triAngles = new int[3];
20     private ObservableList<Double> lines;
21
22     @Override
23     public void start(Stage primaryStage) {
24
25         Pane pane = new Pane();
26         pane.setPadding(new Insets(25, 25, 25, 25));
27
28         //Create a Central Circle
29         Circle cir = setCircle();
30         pane.getChildren().add(cir);
31
32         //Create three random triangle points
33         setTrianglePoints(cir);
34
35         // Create the Triangle according to the points
36         Polygon tri = setTriangle ();
37         pane.getChildren().add(tri);
38
39         //Calculate the initial Angles of the Triangle
40         setTriangleAngles();
41
42         // Set three small red circles on top of each triangle points and label each point with its Angles
43         setSmallCircleAndAngles (pane, cir);
44
45         // Create a scene and place it in the stage
46         Scene scene = new Scene(pane);
47         primaryStage.setTitle("Question 3"); // Set the stage title
48         primaryStage.setScene(scene); // Place the scene in the stage
49         primaryStage.show(); // Display the stage
50     }
51
52     //Set three small red circles on top of each triangle points and label each point with its Angles
53     private void setSmallCircleAndAngles (Pane pane, Circle cir) {
54         for (int i = 0; i < 3; i++) {
55
56             //Set circle points
57             Circle point = new Circle(points[i].getX(), points[i].getY(), 5);
58             point.setFill(Color.RED);
59             point.setStroke(Color.BLACK);
60

```

```

Question1.java Question2.java Question3.java x Question4.java Histogram.java
60
61 //Set angle texts
62 Text text = new Text();
63 text.setText(triAngles[i] + "");
64 text.relocate(point.getCenterX(), point.getCenterY());
65 texts[i] = text;
66
67 //Set each circles to be movable
68 makePointsMoveable(point, text, lines, i, cir);
69 pane.getChildren().addAll(text, point);
70 }
71 }
72
73 //Create a Central Circle
74 private Circle setCircle () {
75     Circle cir = new Circle();
76     cir.setCenterX(200);
77     cir.setCenterY(150);
78     cir.setRadius(100);
79     cir.setFill(Color.WHITE);
80     cir.setStroke(Color.BLACK);
81
82     return cir;
83 }
84
85 //Create three random triangle points
86 private void setTrianglePoints (Circle cir) {
87     for (int i = 0; i < 3; i++) {
88         double random = ThreadLocalRandom.current().nextDouble(0, 2 * Math.PI);
89         // Y
90         double Y = Math.round(cir.getCenterY() + cir.getRadius() * Math.sin(random));
91         // X
92         double X = Math.round(cir.getCenterX() + cir.getRadius() * Math.cos(random));
93         points[i] = new Point2D(X, Y);
94     }
95 }
96
97 //Create the Triangle according to the points
98 private Polygon setTriangle () {
99
100     Polygon tri = new Polygon();
101     tri.getPoints().addAll(points[0].getX(), points[0].getY(),
102                             points[1].getX(), points[1].getY(),
103                             points[2].getX(), points[2].getY());
104     tri.setFill(Color.WHITE);
105     tri.setStroke(Color.BLACK);
106     lines = tri.getPoints();
107
108     return tri;
109 }
110
111 //Calculate the initial Angles of the Triangle
112 private void setTriangleAngles () {
113
114     double sideC = points[1].distance(points[0]);
115     double sideB = points[2].distance(points[0]);
116     double sideA = points[1].distance(points[2]);
117
118     triAngles[0] = (int) Math
119         .toDegrees(Math.acos((sideA * sideA - sideB * sideB - sideC * sideC) / (-2 * sideB * sideC)));
120     triAngles[1] = (int) Math

```



```

121         .toDegrees(Math.acos((sideB * sideB - sideA * sideA - sideC * sideC) / (-2 * sideA * sideC)));
122         triAngles[2] = (int) Math
123             .toDegrees(Math.acos((sideC * sideC - sideB * sideB - sideA * sideA) / (-2 * sideA * sideB)));
124     }
125
126     //Update each angles when a point is move
127     private void updateAngles () {
128
129         double sideC = points[1].distance(points[0]);
130         double sideB = points[2].distance(points[0]);
131         double sideA = points[1].distance(points[2]);
132
133         triAngles[0] = (int) Math
134             .toDegrees(Math.acos((sideA * sideA - sideB * sideB - sideC * sideC) / (-2 * sideB * sideC)));
135         triAngles[1] = (int) Math
136             .toDegrees(Math.acos((sideB * sideB - sideA * sideA - sideC * sideC) / (-2 * sideA * sideC)));
137         triAngles[2] = (int) Math
138             .toDegrees(Math.acos((sideC * sideC - sideB * sideB - sideA * sideA) / (-2 * sideA * sideB)));
139
140         texts[0].setText(triAngles[0] + "");
141         texts[1].setText(triAngles[1] + "");
142         texts[2].setText(triAngles[2] + "");
143     }
144
145
146     //Make each points movable
147     private void makePointsMoveable(Circle point, Text text, ObservableList<Double> line, int index, Circle cir)
148     point.setOnMouseDragged((MouseEvent me) -> {
149
150         Point2D center = new Point2D(cir.getCenterX(), cir.getCenterY());
151         Point2D mouse = new Point2D(me.getX(), me.getY());
152         Point2D centerToMouse = mouse.subtract(center);
153         Point2D centerToNewPoint = centerToMouse.normalize().multiply(cir.getRadius());
154         Point2D newPoint = centerToNewPoint.add(center);
155         Point2D newAnglePoint = centerToNewPoint.add(center).subtract(20, 20);
156
157         points[index] = newPoint;
158         point.setCenterX(newPoint.getX());
159         point.setCenterY(newPoint.getY());
160
161         line.set(index * 2, newPoint.getX());
162         line.set(index * 2 + 1, newPoint.getY());
163
164         updateAngles();
165
166         text.relocate(newAnglePoint.getX(), newAnglePoint.getY());
167     });
168 }
169
170 public static void main(String[] args) {
171     Launch(args);
172 }
173
174 }

```

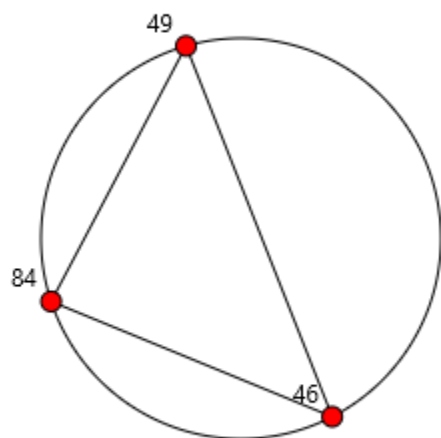
```

10 import javafx.application.Application;
14
15 public class Question3 extends Application {
16
17     private Point2D[] points = new Point2D[3];
18     private Text[] texts = new Text[3];
19     private int[] triAngles = new int[3];
20     private ObservableList<Double> lines;
21
22
23     public void start(Stage primaryStage) {}
24
25
26     private void setSmallCircleAndAngles (Pane pane, Circle cir) {}
27
28
29     private Circle setCircle () {}
30
31
32     private void setTrianglePoints (Circle cir) {}
33
34
35     private Polygon setTriangle () {}
36
37
38     private void setTriangleAngles () {}
39
40
41     private void updateAngles () {}
42
43
44     private void makePointsMoveable(Circle point, Text text, ObservableList<Double> line, int index, Circle cir) {}
45
46     public static void main(String[] args) {}
47
48 }

```

Question 3

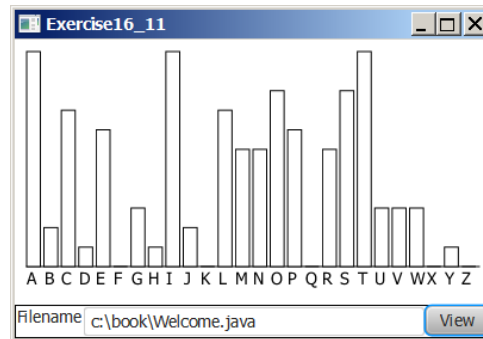
— □ ×



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.



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:

```

1 import javafx.application.Application;
2 import javafx.stage.Stage;
3 import javafx.scene.Scene;
4 import javafx.scene.control.Label;
5 import javafx.scene.control.TextField;
6 import javafx.scene.input.KeyCode;
7 import javafx.scene.control.Button;
8 import javafx.scene.layout.HBox;
9 import javafx.scene.layout.BorderPane;
10 import javafx.geometry.Insets;
11 import java.io.File;
12 import java.util.Scanner;
13
14 public class Question4 extends Application {
15     static int[] counts;
16
17     @Override
18     public void start(Stage primaryStage) {
19         HBox controlBox = new HBox(10);
20         BorderPane pane = new BorderPane();
21
22         //Set bar chart
23         Histogram hist = new Histogram();
24         // Set text field
25         TextField textFilename = new TextField();
26         // Set button
27         Button buttonView = new Button("View");
28
29         // Set text field to listen for ENTER
30         textFilename.setOnKeyPressed(e -> {
31             if (e.getCode() == KeyCode.ENTER) {
32                 readFile(textFilename, hist);
33             }
34         });
35         // Set action on button pressed
36         buttonView.setOnAction(e -> readFile(textFilename, hist));
37         // Set control menu
38         controlBox.getChildren().addAll(new Label("Filename"), textFilename, buttonView);
39         // Set root
40         pane.setCenter(hist);
41         pane.setBottom(controlBox);
42         pane.setMargin(hist, new Insets(20));
43
44         // Create a scene and place it in the stage
45         Scene scene = new Scene(pane);
46         primaryStage.setTitle("Question 4"); // Set the stage title
47         primaryStage.setScene(scene); // Place the scene in the stage
48         primaryStage.show(); // Display the stage
49     }
50
51     //Read Text File
52     private void readFile(TextField textFilename, Histogram hist) {
53         counts = new int[26];
54         File file = new File(textFilename.getText());
55
56         try (Scanner input = new Scanner(file)) {
57             String fileString = "";
58             while (input.hasNext()) {
59                 fileString += input.nextLine().toLowerCase() + "\n";
60             }
61         }

```

```

61
62         countLetters(fileString);
63         hist.setCounts(counts);
64     } catch (Exception ex) {
65         System.out.println(ex);
66     }
67 }
68
69 //Count the letters in text file
70 public static void countLetters(String s) {
71     for (int i = 0; i < s.length(); i++) {
72         char character = s.charAt(i);
73
74         if (Character.isLetter(character)) {
75             counts[character - 97]++;
76         }
77     }
78 }
79
80 public static void main(String[] args) {
81     launch(args);
82 }
83 }

```

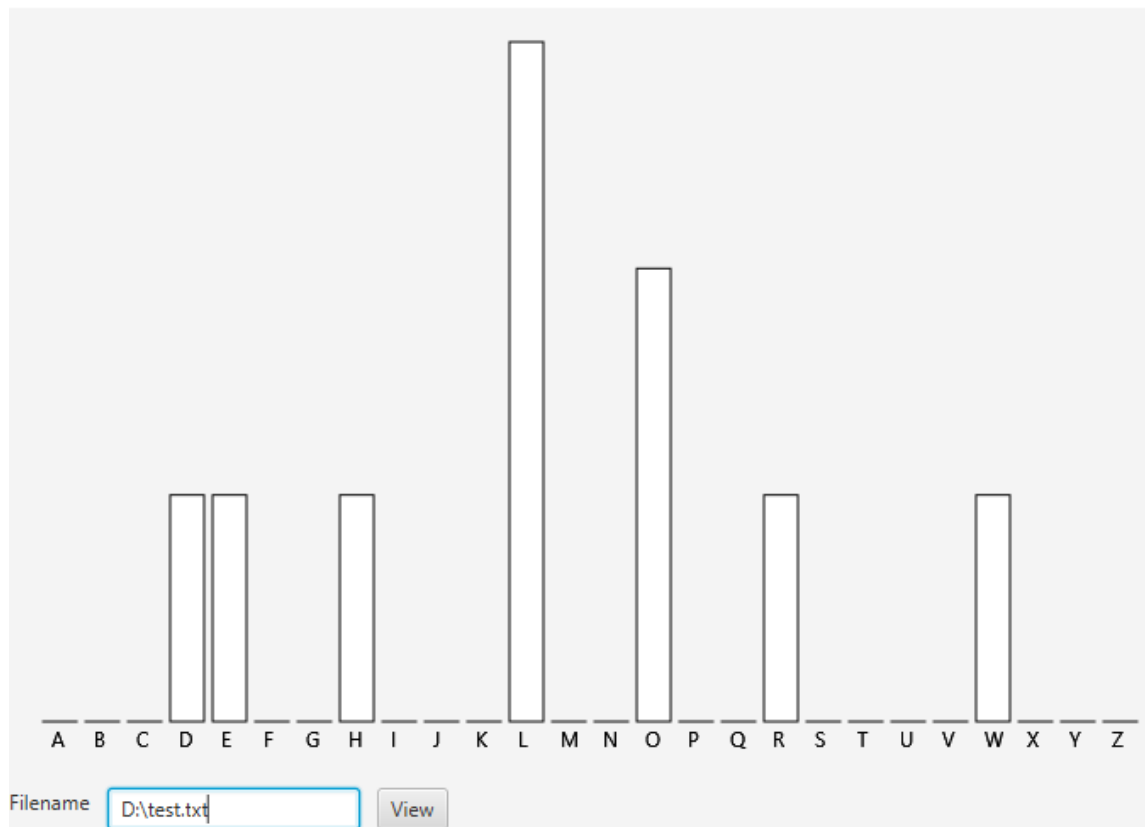


```

1 import javafx.scene.layout.Pane;
2
3 public class Histogram extends Pane {
4     private int[] counts;
5     private int maxValue;
6     private int paneHeight;
7
8     public Histogram() {
9         this.counts = new int[26];
10        maxValue = 0;
11        paneHeight = 400;
12        drawHistogram();
13    }
14
15    //Create the bar chart
16    public void drawHistogram() {
17        for (int i = 0, x = 0; i < counts.length; i++, x += 25) {
18            double rHeight = (double) counts[i] / maxValue * paneHeight;
19
20            Rectangle r = new Rectangle(20, rHeight);
21            r.setX(x);
22            r.setY(paneHeight - rHeight);
23            r.setFill(Color.WHITE);
24            r.setStroke(Color.BLACK);
25
26            Text t = new Text((char) (i + 65) + "");
27            t.setX(x + 5);
28            t.setY(paneHeight + 15);
29            getChildren().addAll(r, t);
30        }
31    }
32
33    //Set the count of letters
34    public void setCounts(int[] counts) {
35        this.counts = counts;
36        maxValue = getMaxValue();
37        drawHistogram();
38    }
39
40    //Get the highest number in count
41    public int getMaxValue() {
42        int m = counts[0];
43        for (int i = 1; i < counts.length; i++) {
44            if (counts[i] > m) {
45                m = counts[i];
46            }
47        }
48        return m;
49    }
50 }

```

Question 4



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.