

Lab 4 Document Henan Mu

Problem 1

Problem description

We need to generate a $n * m$ matrix randomly. In this matrix, every element is 0-9. After we generate it, our task is to find the row and column with the largest sum value.

Analysis

First of all, we let user to input the numbers of rows and columns.

```
Scanner input = new Scanner(System.in);
System.out.print("Enter the number of rows: ");
int rows = input.nextInt();
System.out.print("Enter the number of columns: ");
int columns = input.nextInt();
System.out.println();
input.close();
```

For problem 1, it seems like we can divide this problem into two tasks. First task is to **generate a $n * m$ matrix randomly. In this matrix, every element is 0-9.** Second task is to **find the row and column with the largest sum value.**

For the first task, we get firstly initial an 2D array. And then fill it with random numbers.

```
Random rnd = new Random();
int seed = 1331; //Optional: set seed to create repeatable results
rnd.setSeed(seed);
int[][] matrix = new int[rows][columns];
for (int r = 0; r < rows; r++)
    for (int c = 0; c < columns; c++)
        matrix[r][c] = rnd.nextInt(10);
```

For the second task, to find the row and column with the largest sum value, we need to implement method **sumRow** and **sumColumn**. For sumRow, we compute the sum of the array. For sumColumn, we compute the sum of a specific column.

```
public static int sumRow(int row[]) {
    // TODO: write your code here
    int sum = 0;
    for (int value:row) sum += value;
    return sum;
}

public static int sumColumn(int matrix[][], int column) {
```

```

// TODO: write your code here
int sum = 0;
int numberOfRows = matrix.length;
for (int r = 0; r < numberOfRows; r++) sum += matrix[r][column];
return sum;
}

```

We use these two methods to find the largest sum value.

```

int largestRow = 0;
int indexOfLargestRow = 0;
for (int r = 0; r < rows; r++) {
    if (sumRow(matrix[r]) > largestRow) {
        largestRow = sumRow(matrix[r]);
        indexOfLargestRow = r;
    }
}
System.out.printf("The index of the largest row: %d\n", indexOfLargestRow);

int largestCol = 0;
int indexOfLargestCol = 0;
for (int c = 0; c < columns; c++) {
    if (sumColumn(matrix, c) > largestCol) {
        largestCol = sumColumn(matrix, c);
        indexOfLargestCol = c;
    }
}
System.out.printf("The index of the largest column: %d\n", indexOfLargestCol);

```

Source code

```

package edu.northeastern.csye6200;

import java.util.Random;
import java.util.Scanner;

public class LAB8P1 {
    public static void main(String[] args) {
        // TODO: write your code here
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the number of rows: ");
        int rows = input.nextInt();
        System.out.print("Enter the number of columns: ");
        int columns = input.nextInt();
        System.out.println();
        input.close();

        System.out.println("The array content is:");
    }
}

```

```

Random rnd = new Random();
int seed = 1331; //Optional: set seed to create repeatable results
rnd.setSeed(seed);
int[][] matrix = new int[rows][columns];
for (int r = 0; r < rows; r++)
    for (int c = 0; c < columns; c++)
        matrix[r][c] = rnd.nextInt(10);
for (int r = 0; r < rows; r++) {
    for (int c = 0; c < columns; c++) {
        System.out.print(matrix[r][c]);
        System.out.print(' ');
    }
    System.out.println();
}
System.out.println();

int largestRow = 0;
int indexOfLargestRow = 0;
for (int r = 0; r < rows; r++) {
    if (sumRow(matrix[r]) > largestRow) {
        largestRow = sumRow(matrix[r]);
        indexOfLargestRow = r;
    }
}
System.out.printf("The index of the largest row: %d\n", indexOfLargestRow);

int largestCol = 0;
int indexOfLargestCol = 0;
for (int c = 0; c < columns; c++) {
    if (sumColumn(matrix, c) > largestCol) {
        largestCol = sumColumn(matrix, c);
        indexOfLargestCol = c;
    }
}
System.out.printf("The index of the largest column: %d\n", indexOfLargestCol);
}

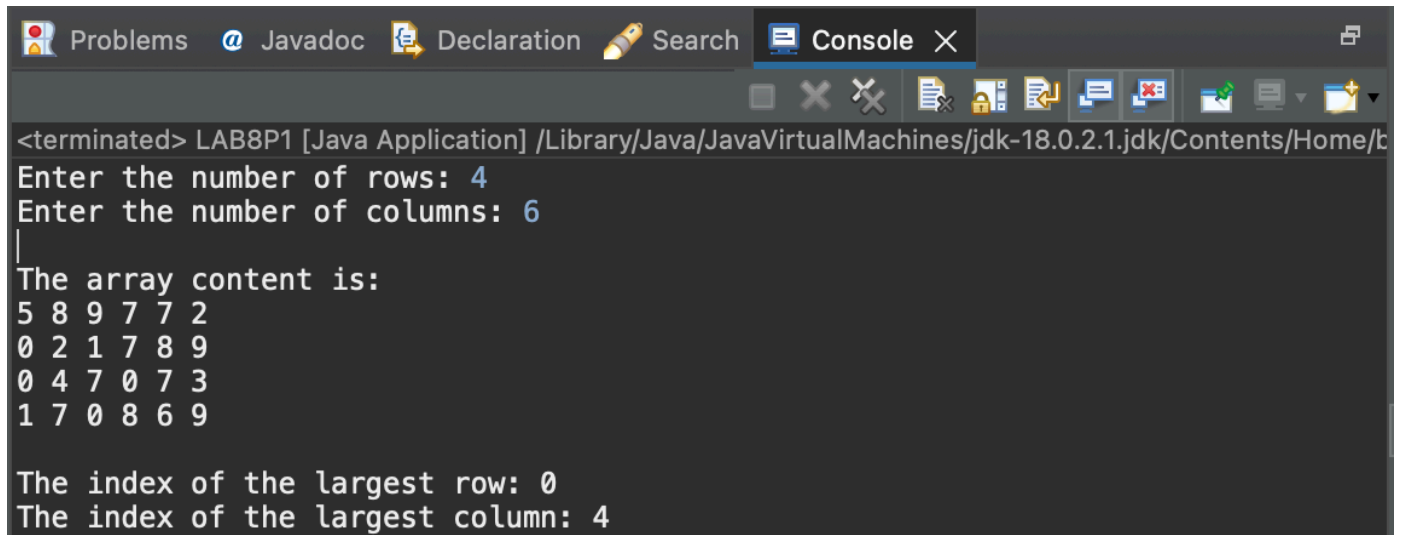
public static int sumRow(int row[]) {
    // TODO: write your code here
    int sum = 0;
    for (int value:row) sum += value;
    return sum;
}

public static int sumColumn(int matrix[][], int column) {
    // TODO: write your code here
    int sum = 0;
    int numberOfRows = matrix.length;
    for (int r = 0; r < numberOfRows; r++) sum += matrix[r][column];
}

```

```
    return sum;
}
}
```

Screenshots of sample runs



```
<terminated> LAB8P1 [Java Application] /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/b
Enter the number of rows: 4
Enter the number of columns: 6
|
The array content is:
5 8 9 7 7 2
0 2 1 7 8 9
0 4 7 0 7 3
1 7 0 8 6 9

The index of the largest row: 0
The index of the largest column: 4
```

Problem 2

Problem description

We need to generate a bar chart to display the percentages of the overall grade represented by project, exams, assignments, and the attendance using javafx. Each bar has its own color, name, position and height.

Analysis

Firstly, we assign some general values like:

```
int widthOfStage = 800;
int hightOfStage = 200;
int widthOfRectangle = widthOfStage / 5;
int hightOfRectangle = 3;
int gap = (int) (widthOfRectangle * 0.2);
```

We use some arrays to store information of each bar:

```
int[] percents = new int[]{35, 30, 30, 5};
Paint[] colors = new Paint[]{Color.BLUE, Color.GREEN, Color.RED,
Color.ORANGE};
String[] texts = new String[]{"Project", "Exams", "Assignments",
"Attendance"};
```

We use a for loop to draw each bar. For each bar:

- use Rectangle
- compute its layout x by its order
- compute its layout y by its height, in other words, by its ratio
- compute its height by its ratio
- use color from Paint[] colors

For each text:

- use Label
- compute its layout x by its order
- compute its layout y by the height of bar
- use text from String[] texts

```

    Pane pane = new Pane();
    for (int i = 0; i < 4; i++) {
        Rectangle rectangle = new Rectangle();
        rectangle.setLayoutX(gap * (i + 1) + i * widthOfRectangle);
        rectangle.setLayoutY(hightOfStage - hightOfRectangle * percents[i]);
        rectangle.setWidth(widthOfRectangle);
        rectangle.setHeight(hightOfRectangle * percents[i]);
        rectangle.setFill(colors[i]);
        pane.getChildren().add(rectangle);
        Label label = new Label(texts[i] + " -- " + String.valueOf(percents[i]) +
'%');
        label.setLayoutX(gap * (i + 1) + i * widthOfRectangle);
        label.setLayoutY(hightOfStage - hightOfRectangle * percents[i] - 20);
        pane.getChildren().add(label);
    }

```

Source code

```

package edu.northeastern.csye6200;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.paint.Paint;
import javafx.scene.shape.Rectangle;
import javafx.stage.Stage;

public class LAB8P2 extends Application{

    @Override
    public void start(Stage primaryStage) throws Exception {
        // TODO: write your code here
        try {

```

```

        int widthOfStage = 800;
        int hightOfStage = 200;
        int widthOfRectangle = widthOfStage / 5;
        int hightOfRectangle = 3;
        int gap = (int) (widthOfRectangle * 0.2);
        int[] percents = new int[]{35, 30, 30, 5};
        Paint[] colors = new Paint[]{Color.BLUE, Color.GREEN, Color.RED,
Color.ORANGE};

        String[] texts = new String[]{"Project", "Exams", "Assignments",
"Attendance"};

        Pane pane = new Pane();
        for (int i = 0; i < 4; i++) {
            Rectangle rectangle = new Rectangle();
            rectangle.setLayoutX(gap * (i + 1) + i * widthOfRectangle);
            rectangle.setLayoutY(hightOfStage - hightOfRectangle * percents[i]);
            rectangle.setWidth(widthOfRectangle);
            rectangle.setHeight(hightOfRectangle * percents[i]);
            rectangle.setFill(colors[i]);
            pane.getChildren().add(rectangle);
            Label lable = new Label(texts[i] + " -- " + String.valueOf(percents[i]) +
'%' );

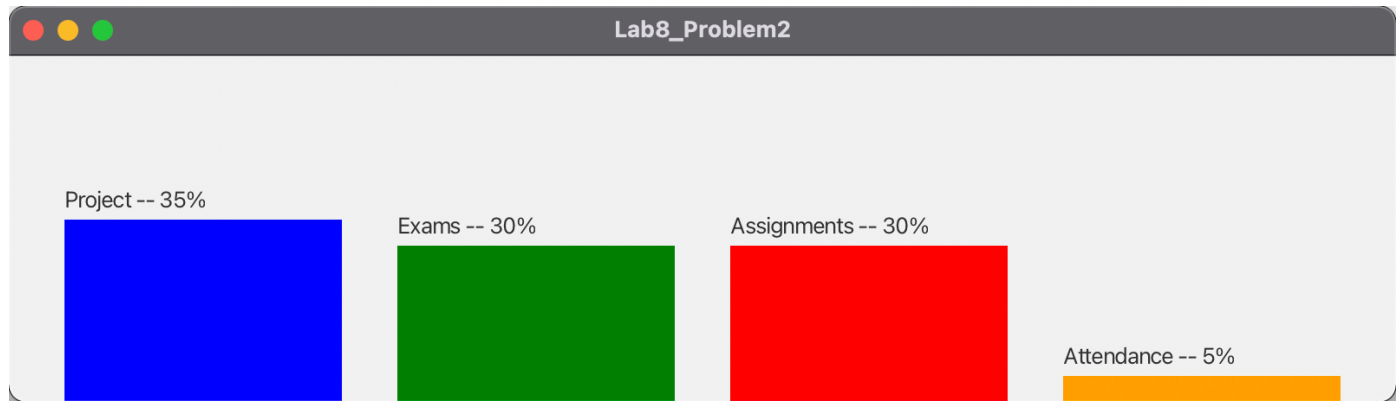
            lable.setLayoutX(gap * (i + 1) + i * widthOfRectangle);
            lable.setLayoutY(hightOfStage - hightOfRectangle * percents[i] - 20);
            pane.getChildren().add(lable);
        }

        Scene scene = new Scene(pane, widthOfStage, hightOfStage); // width height
        primaryStage.setTitle("Lab8_Problem2");
        primaryStage.setScene(scene);
        primaryStage.show();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

/**
 * The main method is only needed for the IDE with limited JavaFX support.
 * Not needed for running from the command line.
 */
public static void main(String[] args) {
    // TODO: write your code here
    launch(args);
}
}

```

Screenshots of sample runs



Problem 3

Problem description

Use TextField to draw a 10-by-10 square matrix. Every element in the matrix should be 0 or 1, randomly generated. Use TextField's setText method to set value 0 or 1 as a string.

Analysis

For the number we can use **Random** to generate random number and convert it to a String. Use TextField's setText method to set it.

```
textField.setText(String.valueOf(rnd.nextInt(2))); // textField is a TextField
```

For the matrix, we need to generate $10 \times 10 = 100$ TextFields. For each TextField:

- compute its layout x by its column and side length
- compute its layout y by its row and side length
- center Text

```
for (int r = 0; r < 10; r++) {
    for (int c = 0; c < 10; c++) {
        TextField textField = new TextField();
        textField.setLayoutX(50 * c);
        textField.setLayoutY(50 * r);
        textField.setText(String.valueOf(rnd.nextInt(2)));
        textField.setPrefSize(50, 50);
        textField.setAlignment(Pos.CENTER);
        pane.getChildren().add(textField);
    }
}
```

Source code

```
package edu.northeastern.csye6200;

import java.util.Random;

import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.TextField;
import javafx.scene.layout.Pane;
import javafx.stage.Stage;

public class LAB8P3 extends Application {

    @Override
    public void start(Stage primaryStage) {
        // TODO: write your code here
        try {
            Pane pane = new Pane();
            Random rnd = new Random();
            int seed = 1331; // Optional: set seed to create repeatable results
            rnd.setSeed(seed);
            for (int r = 0; r < 10; r++) {
                for (int c = 0; c < 10; c++) {
                    TextField textField = new TextField();
                    textField.setLayoutX(50 * c);
                    textField.setLayoutY(50 * r);
                    textField.setText(String.valueOf(rnd.nextInt(2)));
                    textField.setPrefSize(50, 50);
                    textField.setAlignment(Pos.CENTER);
                    pane.getChildren().add(textField);
                }
            }
            Scene scene = new Scene(pane, 500, 500); // width height
            primaryStage.setTitle("Lab8_Problem3 - Extra Credit");
            primaryStage.setScene(scene);
            primaryStage.show();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * The main method is only needed for the IDE with limited JavaFX support.
     * Not needed for running from the command line.
     */
    public static void main(String[] args) {
        // TODO: write your code here
    }
}
```



```
launch(args);  
}  
  
}
```

Screenshots of sample runs

Lab8_Problem3 - Extra Credit									
1	0	1	1	0	1	0	0	0	0
0	1	0	1	1	0	0	0	1	1
1	1	0	0	1	0	0	1	0	0
0	1	1	0	1	1	1	1	0	1
0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	1	0	1	0	0
1	1	1	1	1	0	1	1	0	0
1	1	1	0	1	0	0	1	1	0
1	0	0	0	1	0	1	1	1	0