CSC 221 Final project.

Hodgson tetteh

Hesham Auda

Introduction

The goal of this project was to create populate and control a database from java.After that was done we had to query the data base for specific values and put it on a Piechart displaying the proportions of the gpa in the class.

Method

My project consisted of three parts ,and in order to learn more I decided to not use listing 25 but use mysqlite so I could learn a little more about relational databases, the first part was the class that connected java to my sqllite database called connectTodb.java. I first connected to the database by creating a connection "`static Connection c = null`".

Then assigning the connection to the jdbc Drivermanger fuction that returns a connection from the database url. Also because java has no real inbuilt header to extract and control database a special driver the jdbc had to be imported. And if the connection goes through you are notified else you are also promoted. After this the user is given a bunch of options whether they want to populate the table or the gpa of students using a switch statement if the user wants to see the gpa the code queries the data base for the data that fit the query.

code of this class is given below.

```java
package com.db;

import static java.util.Map.Entry.comparingByValue;
import static java.util.stream.Collectors.toMap;

import java.sql.*;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.Scanner;

public class connecToDb  {

    static String codev = "CSc";
    static HashMap<Character, Integer> hash = new HashMap<>();

    static Connection c = null;

    public connecToDb() {
        String quan = "CSc";
    }

    public static HashMap<Character, Integer> createconnect() throws SQLException
{
        hash.put('A', 0);
        hash.put('B', 0);
        hash.put('C', 0);
        hash.put('D', 0);
        hash.put('E', 0);
        hash.put('F', 0);
        hash.put('W', 0);

        try {
            Class.forName("org.sqlite.JDBC");// load driver
            c = DriverManager.getConnection("jdbc:sqlite:info.db");
            System.out.println("connection succesful");

        } catch (Exception e) {
            System.out.println("connection failed");
            System.exit(0);
        }

        System.out.println("Please select option: 1:Populate tables , 2: See gpa
proportion");
        Scanner read = new Scanner(System.in);
        int process = read.nextInt();
        int process2 = 0;
        switch (process) {
        case (1): {
            System.out.println("Please select what table u want to populate:
");

            Scanner read2 = new Scanner(System.in);
            switch (process2) {
            case(1): {
                System.out.println("you selected Student with fields
studentID, " + "firstName, lastName, sex enter all inorder ,");
                Scanner in1 = new Scanner(System.in);
                Scanner in2 = new Scanner(System.in);
                Scanner in3 = new Scanner(System.in);
                Scanner in4 = new Scanner(System.in);
                System.out.println("enter studentid");
                String studentd = in1.nextLine();
```

```
                          System.out.println("enter firstname");
                          String usernme = in2.nextLine();
                          System.out.println("enter lastname");
                          String slastn = in3.nextLine();
                          System.out.println("enter sex");
                          String sexx = in4.nextLine();
                          String SQLAdd = "INSERT INTO Student(field1, field2, 
field3, field4) " + "VALUES ('" + studentd + "', '" + usernme+ "', '" + slastn + "', 
'" + sexx + "')";
                          
                          Statement stmt = c.createStatement();
                          stmt.executeUpdate(SQLAdd);
                          System.out.println("record added");
                          
                    }
                    
                    case(2): {
                          System.out.println("you selected courses with fields 
(courseID, courseTitle, department )");
                          Scanner in1 = new Scanner(System.in);
                          Scanner in2 = new Scanner(System.in);
                          Scanner in3 = new Scanner(System.in);
                          Scanner in4 = new Scanner(System.in);
                          System.out.println("enter studentid");
                          String courseid = in1.nextLine();
                          System.out.println("enter firstname");
                          String course = in2.nextLine();
                          System.out.println("enter lastname");
                          String department = in3.nextLine();
                          
                          String SQLAdd = "INSERT INTO Courses(field1, field2, 
field3) " + "VALUES ('" + courseid + "', '" + course+ "', '" + department+ "')";
                          
                          Statement stmt = c.createStatement();
                          stmt.executeUpdate(SQLAdd);
                          System.out.println("record added");
                          
                    }
                    
                    
                    
                    case(3): {
                          System.out.println("you selected Classes with fields 
(classCode, courseID, studentID, year, semester, GPA)");
                          Scanner in1 = new Scanner(System.in);
                          Scanner in2 = new Scanner(System.in);
                          Scanner in3 = new Scanner(System.in);
                          Scanner in4 = new Scanner(System.in);
                          System.out.println("enter classCode");
                          String studentd = in1.nextLine();
                          System.out.println("enter courseID");
                          String usernme = in2.nextLine();
                          System.out.println("enter studentID");
                          String slastn = in3.nextLine();
                          System.out.println("enter year");
```

```
                String yearn = in3.nextLine();
                System.out.println("enter semester");
                String sems = in3.nextLine();
                System.out.println("enter GPA");
                String gpaa= in4.nextLine();
                String SQLAdd = "INSERT INTO Student(classCode, courseID, 
studentID, year, semester, GPA) " + "VALUES ('" + studentd + "', '" + usernme+ "', '" 
+ slastn + "', '" + yearn + "', '" + sems+ "', '" + gpaa+ "')";

                Statement stmt = c.createStatement();
                stmt.executeUpdate(SQLAdd);
                System.out.println("record added");

            }


        }
            }
        case (2): {
            System.out.println("about to create statement");
            Statement stmt2 = c.createStatement();
            String SQLRead = "SELECT classCode,courseID, year, semester,GPA 
FROM  Classes";
            System.out.println("");
            ResultSet rs = stmt2.executeQuery(SQLRead);
            // System.out.println("classCode"+"\t"+"courseID");
            System.out.println("check1");
            while (rs.next()) {
                // int id = rs.getInt("id");
                String code = rs.getString("classCode");
                String crseid = rs.getString("courseID");
                String yearstr = rs.getString("year");
                String semesstr = rs.getString("semester");
                String GPAstr = rs.getString("GPA");

                if (code.equals(codev) && yearstr.equals("2019") && 
crseid.equals("211")) {
                    // System.out.println("we made it " + GPAstr);

                    if (hash.containsKey(GPAstr.charAt(0))) {

                        int value = hash.get(GPAstr.charAt(0));
                        hash.put(GPAstr.charAt(0), value + 1);

                    }

                }

            }
            System.out.println("check1");/// end of while loop

            /*
             * hash = hash.entrySet() .stream() .sorted(comparingByValue()) 
.collect(
             * toMap(e -> e.getKey(), e -> e.getValue(), (e1, e2) -> e2,
             * LinkedHashMap::new));
             */
            rs.close();
            stmt2.close();
            c.close();
        }

        }
        System.out.println("retured hash");

        return hash;

    }/// switch case



    /*public static void main(String[] args) throws SQLException {
        connecToDb try1 = new connecToDb();
        HashMap<Character, Integer> bash = try1.createconnect();
        for (char key : bash.keySet()) {
            System.out.println(key + ": " + bash.get(key));
        }


    }// TODO Auto-generated method stub*/

}
```

The second part of my project was the piechart class was this class utilizes the piechart class from the previous section and creates a chart. Firstly it a map from the connecttodb class. It then traverses through each values storing the keys and values into arrays form easy manipulation. This is then sorted and the various frequencies of the gpa are stored in an array. The grapgicscontext object is also passed when this is called and its draws a piechart with various sections code below.

```
package regexdemo;

import java.util.regex.Pattern;

import javafx.scene.canvas.GraphicsContext;

import javafx.scene.shape.ArcType;
import java.util.Random;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.Map;
import java.util.Random;
import java.util.Map.Entry;
import java.lang.Math;

import static java.util.stream.Collectors.*;
import static java.util.Map.Entry.*;
```

```
public class Piechart

{

        Random rnd2 = new Random();

         MyColor ca2 = MyColor.RED ;

         public static Character[] keys = null;

  public  static Integer[] values = null;

        public static frequency f1 = new frequency();

        public static Map<Character, Integer> h1 = new HashMap<Character,Integer>();

        public  static double sum = 0;









        public Piechart(Map<Character, Integer> h1) throws FileNotFoundException

        {
```

```
h1 = f1.freq();

  keys = new Character[h1.size()];

  values = new Integer[h1.size()];

int index = 0;

for (Entry<Character, Integer> mapEntry : h1.entrySet())

{

    keys[index] = mapEntry.getKey();

    values[index] = mapEntry.getValue();

    index++;

}

for(int i = 0; i<= 25; i++)

{

            sum = values[i]+ sum;
```

```
        }

        for(int i = 0; i<= 25; i++)

        {

            System.out.println("keys:  "+ keys[25-i] +"  values:  "+ values[25-i]);

        }

        System.out.println("And sum is  "+sum);



    }




     /********************************************/

/*  public static void main(String[] args) throws FileNotFoundException

    {

        Piechat();

    }*/



     /********************************************/

      public void draw(GraphicsContext g, int frequency)
```

```
 {

        double angle = (frequency/sum)*360;

        g.setFill(ca2.MixColor(rnd2.nextInt(256), rnd2.nextInt(256), rnd2.nextInt(256) ));

        g.fillArc(200, 200, 400, 400, 0, angle,  ArcType.ROUND);

 }




 /******************************************/

public void drawpiechart(GraphicsContext g, int N)

{

          g.strokeOval(300, 100, 400, 400);

          double anglen = 0;

          double angle =0;

          for(int i = 0; i< N; i++)

                 {

                     angle = 0+angle+anglen;



                     anglen = (values[25-i]/sum)*360 ;
}
              System.out.println("strt "+angle + " enda "+ anglen );

              int x,y,z;

              x=rnd2.nextInt(256);

              y = rnd2.nextInt(256);

              z = rnd2.nextInt(256);

            g.setFill(ca2.MixColor(x,y , z ));

            g.fillArc(300, 100, 400, 400, angle, anglen,  ArcType.ROUND);

            String s = String.valueOf(values[25-i]/sum);

            String sm = s+"  "+ keys[25-i];

            g.fillRect(10, 30*i+90, 30, 40); g.strokeText(sm, 50, 30*i+100);







            }

        }



    }
```

The last part of this project was the javafx class called gpadraw class this class takes extends application and creates a canvas that the piechart will be drawn in after that is passes the canves to the piechart class to draw to final chart on. Code below

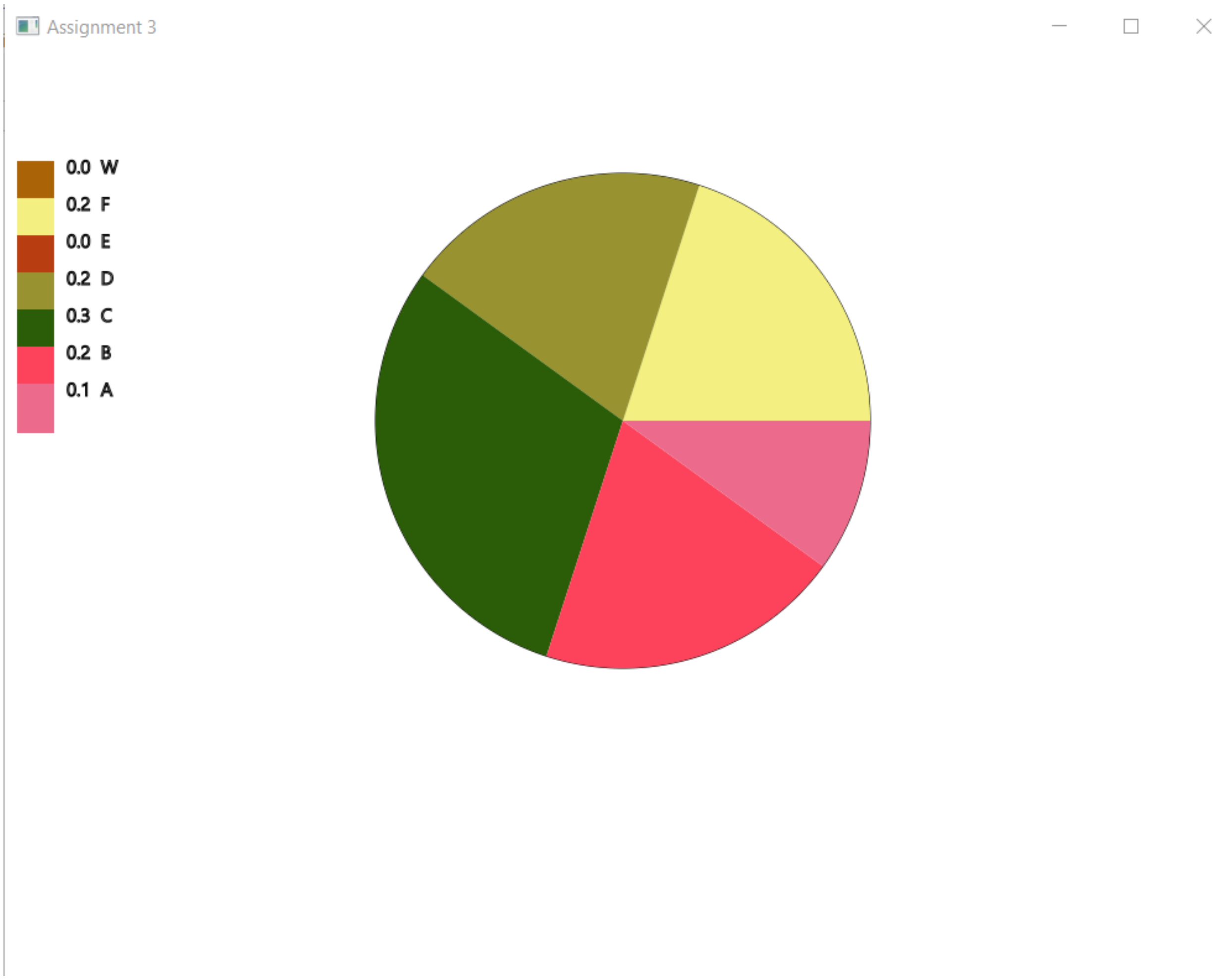
```
import java.util.Random;
import java.util.Scanner;
import java.util.regex.Matcher;

import com.db.connecToDb;

import javafx.scene.canvas.GraphicsContext;
import javafx.scene.shape.ArcType;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.Map;
import java.util.Random;
import java.util.Map.Entry;
import java.lang.Math;
import java.sql.SQLException;

import static java.util.stream.Collectors.*;
import static java.util.Map.Entry.*;

public class gpadraw {
    static Random rnd2 = new Random();
    static MyColor ca2 = MyColor.RED;
    public static Character[] keys = null;
    public static Integer[] values = null;
    static int sum = 0;
    public static Map<Character, Integer> h1 = new HashMap<Character, Integer>();

    public static void getter() throws SQLException {
        HashMap<Character, Integer> gotmap = new HashMap<Character, Integer>();
        connecToDb try1 = new connecToDb();
        gotmap = try1.createconnect();
        h1 = gotmap;

    }

    public static void drawpiechart(GraphicsContext g, int N) {
        try {
            gpadraw.getter();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
```

```
             keys = new Character[h1.size()];
             values = new Integer[h1.size()];

             for (char key : h1.keySet()) {
                    System.out.println(key + ": " + h1.get(key));
                    System.out.println(h1.size());
             }
             int index = 0;
             for (Entry<Character, Integer> mapEntry : h1.entrySet()) {
                    keys[index] = mapEntry.getKey();
                    values[index] = mapEntry.getValue();
                    index++;
             }

             for (int i = 0; i < h1.size(); i++) {
                    sum = values[i] + sum;
             }

             h1 = h1.entrySet().stream().sorted(comparingByValue())
                          .collect(toMap(e -> e.getKey(), e -> e.getValue(), (e1,
e2) -> e2, LinkedHashMap::new));

             for (int i = 0; i < h1.size(); i++) {
                    System.out.println("keys:  " + keys[i] + "  values:  " +
values[i]);
             }
             System.out.println("And sum is  " + sum);

             g.strokeOval(300, 100, 400, 400);

             double anglen = 0;
             double angle = 0;
             for (int i = 0; i <h1.size() ; i++) {
                    angle = 0 + angle + anglen;

                    anglen = ((double)(values[h1.size() - 1 - i]) / sum) * 360;
                    System.out.println("strt " + angle + " enda " + anglen);
                    int x, y, z;
                    x = rnd2.nextInt(256);
                    y = rnd2.nextInt(256);
                    z = rnd2.nextInt(256);
                    g.setFill(ca2.MixColor(x, y, z));
                    g.fillArc(300, 100, 400, 400, angle, anglen, ArcType.ROUND);
                    String s = String.valueOf((double)(values[h1.size() - 1 - i]) /
sum);

                    String sm = s + "  " + keys[h1.size() - 1 - i];
                    g.fillRect(10, 30 * i + 90, 30, 40);
                    g.strokeText(sm, 50, 30 * i + 100);

             }
             /*
              * double anglen = 0; double angle =0; for(int i = 0; i< N; i++) { angle
=
              * 0+angle+anglen;
              *
              * anglen = (values[25-i]/sum)*360 ; System.out.println("strt "+angle +
              * " enda "+ anglen ); int x,y,z; x=rnd2.nextInt(256); y =
rnd2.nextInt(256); z
              * = rnd2.nextInt(256); g.setFill(ca2.MixColor(x,y , z ));
g.fillArc(300, 100,
              * 400, 400, angle, anglen, ArcType.ROUND); String s =
              * String.valueOf(values[25-i]/sum); String sm = s+"  "+ keys[25-i];
              * g.fillRect(10, 30*i+90, 30, 40); g.strokeText(sm, 50, 30*i+100);




       }
}
```

Sql code

```
CREATE TABLE "Classes" (
       "classCode"   TEXT,
       "courseID"    TEXT,
       "year" TEXT,
       "studentID"   TEXT,
       "semester"    TEXT,
       "GPA"  TEXT
)
CREATE TABLE "Courses" (
       "field1"     INTEGER,
       "field2"     TEXT,
       "field3"     TEXT
)
CREATE TABLE "Students" (

       "field1"     INTEGER,

       "field2"     TEXT,

       "field3"     TEXT,

       "field4"     TEXT

)
```

```
"INSERT INTO Student(field1, field2, field3, field4) " + "VALUES ('" + studentd + "',
'" + usernme+ "', '" + slastn + "', '" + sexx + "')";

INSERT INTO Courses(field1, field2, field3) " + "VALUES ('" + courseid + "', '" +
course+ "', '" + department+ "')";

INSERT INTO Student(classCode, courseID, studentID, year, semester, GPA) " + "VALUES
('" + studentd + "', '" + usernme+ "', '" + slastn + "', '" + yearn + "', '" + sems+
"', '" + gpaa+ "')";
```


Assignment 3
0.0 W
0.2 F
0.0 E
0.2 D
0.3 C
0.2 B
0.1 A