Lab1

2) package primeNumber;

public class prime {

  public static void main(String[] args)

    {

        int num=10, count = 0, a;

        String x = "";

        while(num > 0)

        {

            a = num % 2;

            if(a == 1)

            {

                count++;

            }

            x = x + "" + a;

            num = num / 2;

        }

        System.out.println("Binary number:"+x);

    }

     }

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lab2

package lab2;

import java.util.Scanner;

public class Exe1 {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

System.out.println("please enter number");

int num=input.nextInt();

if(isPrime(num))

System.out.println("  is prime number");

else

System.out.println(" not prime number ");

}

public static boolean isPrime(int number){

for(int i=2;i<number/2;i++){

if(number %i==0)

return false;

}

return true;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package lab2;

import java.util.Scanner;

public class exe2 {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

System.out.print("please enter hourse worked(-1 to end ):");

double h=input.nextDouble();

for(;h!=-1;)

{

double salary=0;

System.out.print("please enter rate:");

double r=input.nextDouble();

if (h>=40)

salary=(0.40\*h)+(1.5\*r)\*(h-0.40);

else

salary=h\*r;

System.out.println("salary is"+salary);

System.out.println("please enter hourse worked(-1 to end ):");

 h=input.nextDouble();

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lab3

Lab4

package lab4;

import java.util.Scanner;

public class build {

public static void main(String[] args) {

int x,y,z;

Scanner input=new Scanner(System.in);

System.out.println("please enter the size ");

System.out.println("please enter the key ");

x=input.nextInt();

y=input.nextInt();

z=input.nextInt();

int [] []a=new int [x] [y];

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

for(int j=0;j<y;j++)

a [i][j]=(int)((Math.random())\*10);

boolean k=linearsearch(a,z);

if(k)

System.out.print("found");

else

System.out.print("not found");

}

public static boolean linearsearch(int [][] a,int number){

for(int i=0;i<a.length;i++)

for(int j=0;j<a[i].length;j++){

if(a[i][j]==number)

return true;

}

return false;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package lab4;

import java.util.Scanner;

public class building {

static int avg;

static

public static void main(String[] args) {

int countabove=0,countbellow=0;

Scanner input=new Scanner(System.in);

System.out.println("please enter # of apartment ");

int x=input.nextInt();

int [] []a=new int [x][];

for(int i=0;i<a.length;i++){

System.out.println("please enter # of people "+(i+1));

int a1=input.nextInt();

a[i]=new int [a1+1];

a[i][0]=a1;

for(int j=0;j<a.length;j++){

System.out.println("please enter # of age person ");

a[i][j]=input.nextInt();

}

System.out.println("the number of people "+ sum(a));

}

public static int sum(int [][]a){

int sum=0;

int x=0;

for(int i=0;i<a.length;i++){

for(int j=0;j<a[i].length;j++)

sum+=a[i][j];

}

return sum;

}

public static int bellow(int [][]a,int x){

int count=0,sum=0,local;

for(int i=0;i<a.length;i++){

for(int j=1;j<a[i].length;j++){

sum+=a[i][j];

}

local=sum/a[i][0];

if(local<avg){

count++;

}

}

return count;

}

public static int above(int [][]a,int y){

int count=0,sum=0,local;

for(int i=0;i<a.length;i++){

for(int j=1;j<a[i].length;j++){

sum+=a[i][j];

}

local=sum/a[i][0];

if(local>avg){

count++;

}

}

return count;

}

public static int age(int [][]a,int sum){

int count=0;

int x=0;

for(int i=0;i<a.length;i++){

for(int j=0;j<a[i].length;j++)

sum+=a[i][j];

}

return avg=x/sum;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package lab4;

public class student {

private int student;

private String studentname;

public student() {

}

public student(int student, String studentname) {

this.student = student;

this.studentname = studentname;

}

public int getStudent() {

return student;

}

public void setStudent(int student) {

this.student = student;

}

public String getStudentname() {

return studentname;

}

public void setStudentname(String studentname) {

this.studentname = studentname;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_

package lab4;

public class driver {

public static void main(String[] args) {

student [] a=new student[4];

a[0]=new student(1,"aya");

System.out.println(a[0].getStudent()+ " "+a[0].getStudentname());

a[0]=new student(5,"ahmad");

System.out.println(a[0].getStudent()+ " "+a[0].getStudentname());

a[0]=new student(10,"lama");

System.out.println(a[0].getStudent()+ " "+a[0].getStudentname());

a[0]=new student(14,"duaa");

System.out.println(a[0].getStudent()+ " "+a[0].getStudentname());

}

}

|  |
| --- |
|  |

Lab5

Lab 5\_1

import java.util.Date;

public class Employee {

private int department;

private String name;

private long id;

public Date birthDate;

public Date hireDate;

public double basicSalary;

public Employee() {

this(10,"ahmad",14, new Date(), new Date(),10000);

}

public Employee(int department, String name, long id, Date birthDate, Date hireDate, double basicSalary) {

super();

this.department = department;

this.name = name;

this.id = id;

this.birthDate = birthDate;

this.hireDate = hireDate;

this.basicSalary = basicSalary;

}

public int getDepartment() {

return department;

}

public void setDepartment(int department) {

this.department = department;

}

public String getName() {

return name;

}

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public Date getBirthDate() {

return birthDate;

}

public Date getHireDate() {

return hireDate;

}

public double getBasicSalary() {

return basicSalary;

}

public void setBasicSalary(double basicSalary) {

this.basicSalary = basicSalary;

}

public void printEmployeeinfo(){

System.out.print("name"+this.name+ " department"+this.department+"id"+this.id+"birthDate"+this.birthDate+"hireDate"+this.hireDate+"basicSalary"+this.basicSalary);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import java.util.Date;

public class Driver {

public static void main(String[] args) {

Employee[] employee = new Employee[4];

employee[0] = new Employee(1, "aya", 45, new Date(), new Date(), 3000);

employee[1] = new Employee(2, "ahmad", 30, new Date(), new Date(), 4000);

employee[2] = new Employee(3, "mohammad", 50, new Date(), new Date(), 1000);

employee[3] = new Employee(4, "lama", 60, new Date(), new Date(), 3000);

Employee res=largeSalary(employee);

res.printEmployeeinfo();

}

public static Employee largeSalary(Employee[] employees) {

Employee large = employees[0];

for (int i = 0; i < employees.length; i++)

if (employees[i].basicSalary > employees[0].basicSalary)

large = employees[i];

return large;

}

}

package lab5\_2;

public class MyArray {

private int[] array;

public MyArray(int[] array1) {

array = new int[array1.length];

for (int i = 0; i < array1.length; i++) {

array[i] = array1[i];

}

}

public int min() {

int min = array[0];

for (int i = 0; i < array.length; i++) {

if (min < array[i]) {

min = array[i];

}

}

return min;

}

public int max() {

int max = array[0];

for (int i = 0; i < array.length; i++) {

if (max < array[i]) {

max = array[i];

}

}

return max;

}

public double avg() {

int sum = 0;

int count = 0;

int avg = 0;

for (int i = 0; i < array.length; i++) {

count++;

sum += array[i];

avg = sum / count;

}

return avg;

}

public void printArray() {

for (int i = 0; i < array.length; i++)

System.out.print(array[i]);

}

public int getSize() {

return array.length;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package lab5\_2;

public class driver {

public static void main(String[] args) {

int [] array1={1,2,3,10,4};

MyArray array= new MyArray(array1);

array.printArray();

System.out.println("\n max is  "+array.max());

System.out.println("\n min is  "+array.min());

System.out.println("\n avg is  "+array.avg());

System.out.println("\n"+array.getSize());

}

}

\_\_\_\_\_\_\_\_

package lab5\_3;

public class City {

private String cityName;

private double longitude;

private double latitude;

private int temperature;

public City() {

}

public City(String cityName, double longitude, double latitude, int temperature) {

super();

this.cityName = cityName;

this.longitude = longitude;

this.latitude = latitude;

this.temperature = temperature;

}

public String getCityName() {

return cityName;

}

public void setCityName(String cityName) {

this.cityName = cityName;

}

public double getLongitude() {

return longitude;

}

public void setLongitude(double longitude) {

this.longitude = longitude;

}

public double getLatitude() {

return latitude;

}

public void setLatitude(double latitude) {

this.latitude = latitude;

}

public int getTemperature() {

return temperature;

}

public void setTemperature(int temperature) {

this.temperature = temperature;

}

public void printCityInfo() {

System.out.println("cityname " + this.cityName + "\n" + "longitude" + longitude + "\n" + "latitude" + latitude

+ "\n" + "temperature" + temperature);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package lab5\_3;

public class driver {

public static void main(String[] args) {

City[] city = new City[4];

city[0] = new City("soso", 10.5, 11.5, 25);

city[1] = new City("lolo", 16.5, 11.5, 22);

city[2] = new City("momo", 14.5, 11.5, 23);

city[3] = new City("fofo", 17.5, 11.5, 29);

belowAverage(city, 30);

}

public static void belowAverage(City[] cities, int avg) {

for (int i = 0; i < cities.length; i++) {

if (cities[i].getTemperature() < avg)

System.out.println(cities[i].getCityName());

}

}

}

|  |
| --- |
|  |

Lab6

package lab6;

public class MyString {

public static String reverseString(String x) {

String z = "";

for (int i = x.length() - 1; i >= 0; i--) {

z = z + x.charAt(i);

}

return z;

}

public static boolean isPalindrome(String x) {

String z = x.toLowerCase().trim();

StringBuilder b = new StringBuilder();

char r;

for (int i = 0; i < x.length(); i++) {

r = x.charAt(i);

if (r >= 'a' && r <= 'z') {

b.append(r);

}

}

String d=b.toString();

String f = reverseString(d);

return d.equals(f);

}

|  |
| --- |
|  |

lab7

import java.util.Date;

public class Account {

private int id;

private double balance;

private Date datecreated;

public Account(){

this (0,0.0);

}

public Account(int id,double balance){

this.id=id;

this.balance=balance;

datecreated = new Date();

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public double getBalance() {

return balance;

}

public void setBalance(double balance) {

this.balance = balance;

}

public Date getDatecreated() {

return datecreated;

}

public int withDrow(double amount){

if(balance>0 && amount <=balance){

balance-=amount;

return 0;

}

else

return -1;

}

public int deposit(double amount){

if(amount>0){

balance+=amount;

return 0;

}

else

return -1;

}

@Override

public String toString() {

return "Account [id=" + id + ", balance=" + balance + ", datecreated=" + datecreated + "]";

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public class Checking extends Account  {

private double overdraft;

public Checking() {

super();

}

public Checking(int id, double balance,double overdraft) {

super(id, balance);

this.overdraft=overdraft;

}

public int withDrow(double amount){

if (amount <= getBalance()+overdraft){

setBalance(getBalance()-amount );

return 0;

}

else

return -1;

}

@Override

public String toString() {

return "checking [overdraft=" + overdraft + "]";

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public class Saving extends Account {

public Saving() {

super();

}

public Saving(int id, double balance) {

super(id, balance);

}

@Override

public String toString() {

return "Saving []";

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public class test {

public static void main(String[] args) {

Account account = new Account (5,1000);

 account.deposit(200);

 System.out.println(account.getBalance());

 Checking s = new Checking(1,100,1000);

 System.out.println(s.withDrow(1100));

}

}

Array list

package arraylist;

import java.util.ArrayList;

import java.util.Arrays;

public class q1 {

public static void main(String[] args) {

Integer [] array={5,7,8,9,10};

ArrayList<Integer> array1=new ArrayList <> (Arrays.asList(array));

int x = 0;

for (int i = 0; i < array.length-1; i++) {

for (int j = i+1; j < array.length; j++) {

if(array[i]>array[j])

x=array1.get(i);

array1.set(i, array1.get(j));

array1.set(j, x);

}

}

}

}

Arraylist

package arraylist;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Scanner;

public class q1 {

public static void main(String[] args) {

ArrayList<Integer> array1=new ArrayList <> ();

Scanner input=new Scanner (System.in);

int x=input.nextInt();

for (int i = 0; i < array1.size()-1; i++) {

array1.add(input.nextInt());

sort(array1);

}

}

public static void sort(ArrayList<Integer> array){

int x;

for (int i = 0; i < array.size()-1; i++) {

for (int j = i+1; j < array.size(); j++) {

if(array.get(i)>array.get(j));

x=array.get(i);

array.set(i,array.get(j));

array.set(array.get(j),x);

}

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package arraylist;

import java.lang.reflect.Array;

import java.util.ArrayList;

import java.util.Scanner;

public class q2 {

public static void main(String[] args) {

ArrayList<Double> array1=new ArrayList <> ();

Scanner input=new Scanner (System.in);

int x=input.nextInt();

for (int i = 0; i < array1.size()-1; i++) {

array1.add(input.nextDouble());

}

System.out.println(sum(array1));

}

public static double sum(ArrayList<Double> array){

int x=0;

for (int i = 0; i < array.size()-1; i++)

x+=array.get(i);

return x;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package arraylist;

import java.util.ArrayList;

import java.util.Scanner;

public class q3 {

public static void main(String[] args) {

ArrayList<Integer> array1 = new ArrayList<>();

Scanner input = new Scanner(System.in);

int x = input.nextInt();

for (int i = 0; i < x; i++) {

array1.add(input.nextInt());

}

removeDoublecet(array1);

System.out.println(array1);

}

public static void removeDoublecet(ArrayList<Integer> array) {

int x = 0;

for (int i =0; i < array.size()-1; i++)

for (int j = i+1 ; j < array.size(); j++) {

if (array.get(i).equals(array.get(i)))

array.remove(j);

j--;

}

}

}

Lab8\_1

package arraylist;

public  abstract class  Employee {

private String firstName;

private String lastName;

private int ID;

public Employee(String firstName, String lastName, int iD) {

super();

this.firstName = firstName;

this.lastName = lastName;

ID = iD;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public int getID() {

return ID;

}

public void setID(int iD) {

ID = iD;

}

public abstract double earning();

@Override

public String toString() {

return "Employee [firstName=" + firstName + ", lastName=" + lastName + ", ID=" + ID + "]";

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package arraylist;

public class SalariedEmployee extends Employee{

private double Weeklysalary ;

public SalariedEmployee(String firstName, String lastName, int iD, double Weeklysalary) {

super(firstName, lastName, iD);

this.Weeklysalary = Weeklysalary;

}

@Override

public double earning() {

return Weeklysalary;

}8

@Override

public String toString() {

return super.toString() + "SalariedEmployee [Weeklysalary=" + Weeklysalary + "]";

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package arraylist;

public class CommisionEmployee extends Employee{

private double rate ;

private double grossSales ;

public CommisionEmployee(String firstName, String lastName, int iD,  double rate,double grossSales ) {

super(firstName, lastName, iD);

        this.rate=rate;

       this.grossSales= grossSales;

}

@Override

public double earning() {

return 1;

}

@Override

public String toString() {

return  super.toString() +"CommisionEmployee [rate=" + rate + ", grossSales=" + grossSales + "]";

}

}

|  |
| --- |
|  |

Lab8\_2

package lab8;

public abstract class Shape implements Comparable<Shape>{

protected String color;

protected boolean filed;

public Shape() {

}

public int compareTo(Shape s){

if(this.getArea()==s.getArea())

return 0 ;

else  if(this.getArea()> s.getArea())

return 1 ;

else

return -1;

}

public Shape(String color, boolean filed) {

super();

this.color = color;

this.filed = filed;

}

public String getColor() {

return color;

}

public void setColor(String color) {

this.color = color;

}

public boolean isFiled() {

return filed;

}

public void setFiled(boolean filed) {

this.filed = filed;

}

@Override

public String toString() {

return "Shape [color=" + color + ", filed=" + filed + "]";

}

public  abstract double  getArea();

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package lab8;

public class Circle extends Shape{

protected double radius;

public Circle() {

super();

}

public Circle(String color, boolean filed,double radius) {

super(color, filed);

    this.radius= radius;

}

public double getRadius() {

return radius;

}

public void setRadius(double radius) {

this.radius = radius;

}

public double getArea() {

return  radius\* radius\*3.14;

}

@Override

public String toString() {

return super.toString()+ "Circle [radius=" + radius + "]";

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package lab8;

public class Rectangle extends Shape{

protected double width;

protected double length;

public Rectangle() {

super();

}

public Rectangle(String color, boolean filed,double width, double length) {

super(color,filed);

this.width = width;

this.length = length;

}

public double getWidth() {

return width;

}

public void setWidth(double width) {

this.width = width;

}

public double getLength() {

return length;

}

public void setLength(double length) {

this.length = length;

}

@Override

public double getArea() {

return width\*length;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package lab8;

import java.util.ArrayList;

import java.util.Collections;

public class Test {

public static void main(String[] args) {

ArrayList<Shape> shape=new ArrayList<>();

shape.add(new Circle("green",true,5));

shape.add(new Circle("red",false,4));

shape.add(new Circle("blue",true,7));

shape.add(new Rectangle("red", false, 7,2));

shape.add(new Rectangle("white", true, 5,4));

        Collections.sort(shape);

        for (int i = 0; i < shape.size(); i++) {

         System.out.println(shape.get(i));

}

}

}

}

LAB10\_3

package lab10;

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.control.RadioButton;

import javafx.scene.control.TextField;

import javafx.scene.control.ToggleGroup;

import javafx.scene.layout.BorderPane;

import javafx.scene.layout.FlowPane;

import javafx.scene.layout.HBox;

import javafx.scene.layout.Pane;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

public class checkboard extends Application{

public static void main(String[] args) {

Application.launch(args);

}

public void start(Stage primaryStage) throws Exception {

Label l1=new Label("Text Field");

TextField f1=new TextField();

HBox h =new HBox();

h.getChildren().addAll(l1,f1);

RadioButton r1=new RadioButton("Left");

RadioButton r2=new RadioButton("Center");

r2.setSelected(true);

RadioButton r3=new RadioButton("Right");

ToggleGroup group=new ToggleGroup();

r1.setToggleGroup(group);

r2.setToggleGroup(group);

r3.setToggleGroup(group);

Label l2=new Label("Column Size");

TextField f2=new TextField();

HBox h1=new HBox();

h1.getChildren().addAll(r1,r2,r3,l2,f2);

h1.setSpacing(10);

    VBox v= new VBox();

    v.getChildren().addAll(h,h1);

Scene s=new Scene(v);

primaryStage.setScene(s);

primaryStage.show();

}

}

LAB 10

package lab10;

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.RadioButton;

import javafx.scene.control.ToggleGroup;

import javafx.scene.layout.BorderPane;

import javafx.scene.layout.HBox;

import javafx.scene.layout.Pane;

import javafx.scene.layout.VBox;

import javafx.scene.paint.Color;

import javafx.scene.shape.Circle;

import javafx.scene.shape.Rectangle;

import javafx.stage.Stage;

public class traffic extends Application{

public static void main(String[] args) {

Application.launch(args);

}

public void start(Stage primaryStage) throws Exception {

Rectangle r=new Rectangle(100,5,30,90);

r.setFill(Color.WHITE);

r.setStroke(Color.BLACK);

Circle c1=new Circle(115,20,10);

c1.setFill(Color.WHITE);

c1.setStroke(Color.BLACK);

Circle c2=new Circle(115,50,10);

c2.setFill(Color.WHITE);

c2.setStroke(Color.BLACK);

Circle c3=new Circle(115,80,10);

c3.setFill(Color.WHITE);

c3.setStroke(Color.BLACK);

Pane pane=new Pane();

pane.getChildren().addAll(r,c1,c2,c3);

RadioButton r1=new RadioButton("Red");

r1.setSelected(true);

RadioButton r2=new RadioButton("Green");

RadioButton r3=new RadioButton("Yellow");

ToggleGroup group=new ToggleGroup();

r1.setToggleGroup(group);

r2.setToggleGroup(group);

r3.setToggleGroup(group);

HBox h=new HBox();

h.getChildren().addAll(r1,r2,r3);

h.setSpacing(10);

BorderPane p=new BorderPane();

p.setCenter(pane);

p.setBottom(h);

Scene s=new Scene(p);

primaryStage.setScene(s);

primaryStage.show();

r1.setOnAction(x -> {

c1.setFill(Color.RED);

c2.setFill(Color.WHITE);

c3.setFill(Color.WHITE);

});

r2.setOnAction(x -> {

c1.setFill(Color.WHITE);

c2.setFill(Color.GREEN);

c3.setFill(Color.WHITE);

});

r3.setOnAction(x -> {

c1.setFill(Color.WHITE);

c2.setFill(Color.WHITE);

c3.setFill(Color.YELLOW);

});

}

}

LAB 12

public interface MajorInterface {

int FIVE\_YEAR = 5;

int FOUR\_YEAR = 4;

String CS = "CS";

String CSE = "CSE";

void displayInfo();

int getPlan();

}

public abstract class Major implements MajorInterface{

private String title;

private int plan;

private int years;

private String description;

public Major(String title, int plan, int years, String description) {

super();

this.title = title;

this.plan = plan;

this.years = years;

this.description = description;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public int getPlan() {

return plan;

}

public void setPlan(int plan) {

this.plan = plan;

}

public int getYears() {

return years;

}

public void setYears(int years) {

this.years = years;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

@Override

public abstract String toString();

@Override

public void displayInfo() {

System.out.println("Major [title=" + title + ", plan=" + plan + ", years=" + years + ", description=" + description + "]");

}

}

public class CS extends Major{

public CS(){

super(MajorInterface.CS, 165, MajorInterface.FOUR\_YEAR, "Best Major ever!");

}

@Override

public String toString() {

return MajorInterface.CS + " Major";

}

}

public class CSE extends Major{

public CSE(){

super(MajorInterface.CSE, 165, MajorInterface.FIVE\_YEAR, "Enda7ak 3alena mtele3 kahrbaa kolo");

}

@Override

public String toString() {

return MajorInterface.CSE + " Major";

}

}

public class Student implements Comparable<Student>{

private int id;

private String name;

private Major major;

private double average;

private static int numOfCreatedStudents;

public Student(int id, String name, Major major, double average) {

super();

this.id = id;

this.name = name;

this.major = major;

this.average = average;

numOfCreatedStudents++;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Major getMajor() {

return major;

}

public void setMajor(Major major) {

this.major = major;

}

public double getAverage() {

return average;

}

public void setAverage(double average) {

this.average = average;

}

public static int getNumOfCreatedStudents() {

return numOfCreatedStudents;

}

public static void setNumOfCreatedStudents(int numOfCreatedStudents) {

Student.numOfCreatedStudents = numOfCreatedStudents;

}

@Override

public String toString() {

return "Student [id=" + id + ", name=" + name + ", major=" + major + ", average=" + average + "]";

}

@Override

public int compareTo(Student o) {

return (int)(this.average - o.getAverage());

}

}

import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.Scanner;

import javafx.application.Application;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.layout.\*;

import javafx.scene.layout.HBox;

import javafx.stage.Stage;

public class Driver extends Application{

public static ArrayList<Student> students = new ArrayList<>();

public static void main(String[] args) {

File fofo = new File("students.txt");

if (fofo.exists()) {

try {

Scanner soso = new Scanner(fofo);

String line;

String[] tokens;

while(soso.hasNext()){

line = soso.nextLine();

tokens = line.split(":");

if(tokens.length == 4){

if(tokens[2].trim().equals(MajorInterface.CS)){

students.add(new Student(Integer.parseInt(tokens[0]), tokens[1],

new CS(), Double.parseDouble(tokens[3])));

}

else if(tokens[2].trim().equals(MajorInterface.CSE)){

students.add(new Student(Integer.parseInt(tokens[0]), tokens[1],

new CSE(), Double.parseDouble(tokens[3])));

}

}

}

} catch (FileNotFoundException f) {

System.out.println(f);

}

} else

System.out.println("Error: file not found!!!!");

System.out.println(""+studentsAverage("CSE"));

launch(args);

}

public static Student topStudent(String major) {

Student max = null;

for (int i = 0; i < students.size(); i++) {

if(students.get(i).getMajor().getTitle().equals(major)) {

max = students.get(i);

break;

}

}

for (int i = 0; i < students.size(); i++) {

if(students.get(i).getMajor().getTitle().equals(major)) {

if(max.getAverage() < students.get(i).getAverage())

   max = students.get(i);

}

}

return max;

}

public static double studentsAverage(String major) {

double sum=0;

int count=0;

for (int i = 0; i < students.size(); i++) {

if(students.get(i).getMajor().getTitle().equals(major)) {

sum += students.get(i).getAverage();

count++;

}

}

return sum / count;

}

@Override

public void start(Stage soso) throws Exception {

BorderPane bp = new BorderPane();

HBox hb = new HBox();

hb.setAlignment(Pos.CENTER);

RadioButton csRB = new RadioButton("CS");

RadioButton cseRB = new RadioButton("CSE");

ToggleGroup tg = new ToggleGroup();

csRB.setToggleGroup(tg);

cseRB.setToggleGroup(tg);

csRB.setSelected(true);

hb.setSpacing(20);

hb.getChildren().addAll(new Label("Major: "), csRB, cseRB);

bp.setTop(hb);

GridPane gp = new GridPane();

Button topB = new Button("Top Major Student");

Button avgB = new Button("Major Average");

TextField topTF = new TextField();

TextField avgTF = new TextField();

gp.add(topB, 0, 0);

gp.add(avgB, 0, 1);

gp.add(topTF, 1, 0);

gp.add(avgTF, 1, 1);

bp.setCenter(gp);

gp.setHgap(20);

gp.setVgap(10);

gp.setAlignment(Pos.CENTER);

bp.setAlignment(gp, Pos.CENTER);

Label l = new Label("Yahoooooooooooooooooo");

bp.setBottom(l);

csRB.setOnAction(e -> {

int count=0;

for (int i = 0; i < students.size(); i++)

if(students.get(i).getMajor().getTitle().equals("CS"))

count++;

l.setText("The total number of students is : " + count);

});

cseRB.setOnAction(e -> {

int count=0;

for (int i = 0; i < students.size(); i++)

if(students.get(i).getMajor().getTitle().equals("CSE"))

count++;

l.setText("The total number of students is : " + count);

});

topB.setOnAction(e -> {

topTF.setText(""+topStudent(csRB.isSelected()?"CS":"CSE"));

});

avgB.setOnAction(e -> {

avgTF.setText("The avergae is: "+studentsAverage(csRB.isSelected()?"CS":"CSE"));

});

Scene s = new Scene(bp, 400, 200);

soso.setScene(s);

soso.show();

}

}

1171970: Isra :CS: 87

1172740: Samera : CS : 90

1170600: Dima : CS : 98

1171740: Salsabel: CSE : 60

1171203: Yasmeen: CS                    : 120

1109123: Abbas : CSE : 55

1109123: SOSO : ENG : 55

1111111: Abdallah: CSE: 70

1109123: FOFO : MTH : 55