Project code

**WrongSin.java:**

public class WrongSin extends Exception

{

String msg = "The sin calculate is not correct";

public WrongSin()

{

System.out.println(msg);

}

public WrongSin(String t)

{

super(t);

}

public String getMessage()

{

return msg;

}

public String toString()

{

String out;

out = super.toString()+"The sin calculate exceed limits";

return out;

}

}

**WrongCos.java**

public class WrongCos extends Exception

{

String msg = "The cos calculate is not correct";

public WrongCos()

{

System.out.println(msg);

}

public WrongCos(String t)

{

super(t);

}

public String getMessage()

{

return msg;

}

public String toString()

{

String out;

out = super.toString()+"The cos calculate exceed limits";

return out;

}

}

**Charges.java**

import java.util.\*;

import java.io.\*;

public abstract class Charges implements INT,Serializable

{

double x = 0.0;

double y = 0.0;

double charge = 0.0;

static int id=0;

int realid =0;

public Charges()

{

x = 0.0;

y = 0.0;

charge = 0.0;

id++;

realid = id;

}

public Charges(double a, double b, double c)

{

x = a;

y =b;

charge = c;

id++;

realid = id;

}

public double getX()

{

return x;

}

public double getY()

{

return y;

}

public double getCharge()

{

return charge;

}

public int getID()

{

return realid;

}

public void setX(double d)

{

x=d;

}

public void setY(double e)

{

y=e;

}

public void setCharge(double f)

{

charge =f;

}

public String toString()

{

String out;

out = "The charge coordinate is: "+"("+x+", "+y+")"+" "+"The charge is: "+charge;

return out;

}

public abstract double calculateXComponent(Points a);

public abstract double calculateYComponent(Points a);

}

**PositiveCharge.java**

import java.util.\*;

import java.io.\*;

public class PositiveCharge extends Charges implements INT,Serializable

{

public PositiveCharge()

{

super();

}

public PositiveCharge(double a, double b, double c)

{

super(a,b,c);

}

public double calculateSin(Points a) throws WrongSin

{

double x1 = 2\*a.getX()-this.getX();

double y1 = 2\*a.getY()-this.getY();

double xminus = Math.pow(x1-a.getX(),2);

double yminus = Math.pow(y1-a.getY(),2);

double distance = Math.sqrt(xminus+yminus);

double side = y1-a.getY();

double sin = side/distance;

try

{

if(sin>1||sin<-1)

{

throw new WrongSin();

}

}

catch(WrongSin ws)

{

System.out.println(ws.getMessage());

sin = 10000;

}

finally

{

return sin;

}

}

public double calculateCos(Points a) throws WrongCos

{

double x1 = 2\*a.getX()-this.getX();

double y1 = 2\*a.getY()-this.getY();

double xminus = Math.pow(x1-a.getX(),2);

double yminus = Math.pow(y1-a.getY(),2);

double distance = Math.sqrt(xminus+yminus);

double side = x1-a.getX();

double cos = side/distance;

try

{

if(cos>1||cos<-1)

{

throw new WrongCos();

}

}

catch(WrongCos wc)

{

System.out.println(wc.getMessage());

cos = 10000;

}

finally

{

return cos;

}

}

public double calculateMagnitude(Points a)

{

double x1 = 2\*a.getX()-this.getX();

double y1 = 2\*a.getY()-this.getY();

double xminus = Math.pow(x1-a.getX(),2);

double yminus = Math.pow(y1-a.getY(),2);

double distance = Math.sqrt(xminus+yminus);

double k = Math.pow(10,9)\*8.99;

double c = this.getCharge();

double e = (k\*c)/Math.pow(distance,2);

return e;

}

//component of x and y

public double calculateXComponent(Points a)

{

double comp;

try{

comp = this.calculateMagnitude(a)\*this.calculateCos(a);

}

catch(WrongCos wc)

{

System.out.println("cos exceed limit");

comp = 100000;

}

return comp;

}

public double calculateYComponent(Points a)

{

double comp;

try{

comp = Math.abs(this.calculateMagnitude(a)\*this.calculateSin(a));

}

catch(WrongSin ws)

{

System.out.println("sin exceed limit");

comp = 10000;

}

return comp;

}

public String toString()

{

String out;

out = "a positive charge with"+super.toString();

return out;

}

}

**NegativeCharge.java**

import java.util.\*;

import java.io.\*;

public class NegativeCharge extends Charges implements INT,Serializable

{

public NegativeCharge()

{

super();

}

public NegativeCharge(double a, double b, double c)

{

super(a,b,c);

}

public double calculateSin(Points a) throws WrongSin

{

double x1 = this.getX();

double y1 = this.getY();

double xminus = Math.pow(x1-a.getX(),2);

double yminus = Math.pow(y1-a.getY(),2);

double distance = Math.sqrt(xminus+yminus);

double side = y1-a.getY();

double sin = side/distance;

try

{

if(sin>1||sin<-1)

{

throw new WrongSin();

}

}

catch(WrongSin ws)

{

System.out.println(ws.getMessage());

sin = 10000;

}

finally

{

return sin;

}

}

public double calculateCos(Points a) throws WrongCos

{

double x1 = this.getX();

double y1 = this.getY();

double xminus = Math.pow(x1-a.getX(),2);

double yminus = Math.pow(y1-a.getY(),2);

double distance = Math.sqrt(xminus+yminus);

double side = a.getX()-x1;

double cos = side/distance;

try

{

if(cos>1||cos<-1)

{

throw new WrongCos();

}

}

catch(WrongCos wc)

{

System.out.println(wc.getMessage());

cos = 10000;

}

finally

{

return cos;

}

}

public double calculateMagnitude(Points a)

{

double x1 = this.getX();

double y1 = this.getY();

double xminus = Math.pow(x1-a.getX(),2);

double yminus = Math.pow(y1-a.getY(),2);

double distance = Math.sqrt(xminus+yminus);

double k = Math.pow(10,9)\*8.99;

double c = this.getCharge();

double e = (k\*c)/Math.pow(distance,2);

return e;

}

public double calculateXComponent(Points a)

{

double comp;

try{

comp = this.calculateMagnitude(a)\*this.calculateCos(a);

}

catch(WrongCos wc)

{

System.out.println("cos exceed limit");

comp = 100000;

}

return comp;

}

public double calculateYComponent(Points a)

{

double comp;

try{

comp = Math.abs(this.calculateMagnitude(a)\*this.calculateSin(a));

}

catch(WrongSin ws)

{

System.out.println("sin exceed limit");

comp = 10000;

}

return comp;

}

public String toString()

{

String out;

out = "a negative charge with"+super.toString();

return out;

}

}

**Points.java**

import java.util.\*;

import java.io.\*;

public class Points

{

String name =" ";

double x =0.0;

double y =0.0;

static int id = 0;

int realid =0;

public Points()

{

name = "P0";

x =0.0;

y = 0.0;

}

public Points(String a, double b, double c)

{

name =a;

x = b;

y = c;

}

public String getName()

{

return name;

}

public double getX()

{

return x;

}

public double getY()

{

return y;

}

public void setX(double d)

{

x = d;

}

public void setY(double e)

{

y=e;

}

public void setName(String f)

{

name =f;

}

public String toString()

{

String out;

out = "The point: "+name+"."+" "+"The coordinate is: ("+x+","+" "+y+")";

return out;

}

}

**INT.java**

import java.io.\*;

import java.util.\*;

public interface INT

{

public double calculateSin(Points a) throws WrongSin;

public double calculateCos(Points a) throws WrongCos;

public double calculateMagnitude(Points a);

}

**MainClass.java**

import java.util.\*;

import java.io.\*;

import java.text.\*;

public class MainClass

{

public static void main (String [] args)

{

DecimalFormat dec = new DecimalFormat("#,###,###,##0.00");

Scanner scan1 = new Scanner(System.in);

System.out.println("what is the name of the data file? including the .txt");

String li = scan1.nextLine();

//read the text file

ArrayList<Points> point = new ArrayList<Points>();

ArrayList<Charges> charge = new ArrayList<Charges>();

FileInputStream filestream;

BufferedReader buff;

try

{

filestream = new FileInputStream(li);

buff = new BufferedReader(new InputStreamReader(filestream));

String read = " ";

while((read=buff.readLine())!=null)

{

StringTokenizer stok = new StringTokenizer(read,":");

String check = stok.nextToken();

if(check.equals("P"))

{

String info1 = stok.nextToken();

StringTokenizer stok2 = new StringTokenizer(info1,",");

String pointname = stok2.nextToken();

String xstring = stok2.nextToken();

double xreal = Double.parseDouble(xstring);

String ystring = stok2.nextToken();

double yreal = Double.parseDouble(ystring);

Points p1 = new Points(pointname,xreal,yreal);

point.add(p1);

}

if(check.equals("Q"))

{

String info2 = stok.nextToken();

StringTokenizer stok3 = new StringTokenizer(info2,",");

String xstr = stok3.nextToken();

double xre = Double.parseDouble(xstr);

String ystr = stok3.nextToken();

double yre = Double.parseDouble(ystr);

String cha = stok3.nextToken();

double charg = Double.parseDouble(cha);

if(charg>=0)

{

PositiveCharge pos = new PositiveCharge(xre,yre,charg);

charge.add(pos);

}

if(charg<0)

{

NegativeCharge neg = new NegativeCharge(xre,yre,charg);

charge.add(neg);

}

}

}

filestream.close();

buff.close();

}

catch(IOException ioe)

{

System.out.println("something wrong with the file");

}

//calculate the charges for each point

for(Points a: point)

{

double xtotal =0.0;

double ytotal =0.0;

//iterate through all charge to calculate all the impact

for(Charges b: charge)

{

System.out.println("------Calculating point: "+a.toString()+" "+"With charge: #"+b.getID());

double mag1 = b.calculateMagnitude(a);

String mag = dec.format(mag1);

System.out.println("The magnitude at point"+ a.getName()+" due to charge point "+b.getID() + " "+"is: "+mag);

double co1=10000;

double si1=10000;

try{

co1 = b.calculateCos(a);

String co = dec.format(co1);

System.out.println("Cos of the angle is: "+co);

double xtemp1 = b.calculateXComponent(a);

String xtemp = dec.format(xtemp1);

System.out.println("The x component is: " + xtemp);

xtotal= xtotal+ xtemp1;

}

catch(WrongCos wc)

{

System.out.println(wc.getMessage());

}

try{

si1 = b.calculateSin(a);

String si = dec.format(si1);

System.out.println("Sine of the angle is: "+si);

double ytemp1 = b.calculateYComponent(a);

String ytemp = dec.format(ytemp1);

System.out.println("The y component is: "+ytemp);

ytotal =ytotal+ytemp1;

}

catch(WrongSin ws)

{

System.out.println(ws.getMessage());

}

System.out.println("--------------------------------------------");

}

double totalcharge1 = Math.sqrt(Math.pow(xtotal,2)+Math.pow(ytotal,2));

String totalcharge = dec.format(totalcharge1);

System.out.println("The total charge of point: "+a.getName()+" is: "+totalcharge);

System.out.println("--------------------------------------------");

}

//output to file charges

String filename="charges.ser";

FileOutputStream fos=null;

ObjectOutputStream out=null;

try

{

fos=new FileOutputStream(filename,false);

out=new ObjectOutputStream(fos);

for(Charges ar: charge)

{

out.writeObject(ar);

}

out.close();

System.out.println("-----------charges.ser ready to read----------------------");

}

catch(IOException ioe)

{

ioe.printStackTrace();

}

//read file

Scanner scan = new Scanner(System.in);

System.out.println("Do you want to have the charges.ser binary coded file read?");

String check2 = scan.nextLine();

if(check2.equals("yes"))

{

ObjectInputStream ob = null;

FileInputStream fis = null;

Charges cha = null;

ArrayList<Charges> charge2 = new ArrayList<Charges>();

try

{

fis = new FileInputStream("charges.ser");

ob = new ObjectInputStream(fis);

try

{

while(true)

{

cha =(Charges)ob.readObject();

charge2.add(cha);

}

}

catch(EOFException eof)

{

System.out.println("end of file");

}

ob.close();

fis.close();

}

catch(IOException ioe)

{

ioe.printStackTrace();

}

catch(ClassNotFoundException cnf)

{

cnf.printStackTrace();

}

for(Charges f: charge2)

{

System.out.println(f.toString());

}

}

else

{

System.out.println("Terminating program");

System.exit(0);

}

}

}