//<applet code="Hg" width="300" height="300">

//</applet>

import java.applet.Applet;

import java.awt.Color;

import java.awt.Graphics;

import java.util.Scanner;

import java.io.\*;

import java.\*;

public class Hg extends Applet

{

public void paint(Graphics g)

{

int x=10,y=10,width=50,cal1=0,cal2=0,cal3=0,cal4=0,k=0;

double inst1,inst2,inst3,inst4,lst1=0,hst1=0,lst2=0,hst2=0,lst3=0,hst3=0,lst4=0,hst4=0;

double []vrbl1=new double[150];

double []vrbl2=new double[150];

double []vrbl3=new double[150];

double []vrbl4=new double[150];

double rng1=0.0,rng2=0.0,rng3=0.0,rng4=0.0;

try

{

   vrbl1 = getarr1();

   vrbl2 = getarr2();

   vrbl3 = getarr3();

   vrbl4 = getarr4();

   lst1=vrbl1[0];

   hst1=vrbl1.length-1;

   hst1=vrbl1[(int) hst1];

   rng1=(hst1-lst1)/10.0;

   lst2=vrbl2[0];

   hst2=vrbl2.length-1;

   hst2=vrbl2[(int) hst2];

   rng2=(hst2-lst2)/10.0;

   lst3=vrbl3[0];

   hst3=vrbl3.length-1;

   hst3=vrbl3[(int) hst3];

   rng3=(hst3-lst3)/10.0;

   lst4=vrbl4[0];

   hst4=vrbl4.length-1;

   hst4=vrbl4[(int) hst4];

   rng4=(hst4-lst4)/10.0;

   System.out.println("rng1"+rng1);

   System.out.println("rng2"+rng2);

   System.out.println("rng3"+rng3);

   System.out.println("rng4"+rng4);

   g.drawString("Vrbl1", 10, 50);

// The code shows drawstring method ( ) in use with component’s paint method.

   while(lst1<=hst1)

   {

for (int i = 0; i < vrbl1.length; i++)

{

//if (vrbl1[i]==inst1)

   if ((vrbl1[i]>=lst1) && vrbl1[i]<=(lst1+rng1))

{

       cal1++;

}

        }

g.fillRect(x+200,y+(k\*10),c1\*10,20);

c1=0;

k++;

lst1=lst1+rng1;

   }

// vrbl 1’s while closed

   k=12;

   g.drawString("Vrbl2", 10, 120);

   while(lst2<=hst2)

   {

for (int i = 0; i < vrbl2.length; i++)

{

//if (vrbl2[i]==inst1)

   if ((vrbl2[i]>=lst2) && vrbl2[i]<=(lst2+rng2))

{

       cal1++;

}

        }

g.fillRect(x+200,y+(k\*10),c1\*10,20);

cal1=0;

k++;

lst2=lst2+rng2;

   }  // Close of While for vrbl 2

   k=26;

   g.drawString("Vrbl3", 10, 250);

   while(lst3<=hst3)

   {

for (int i = 0; i < vrbl3.length; i++)

{

//if (vrbl2[i]==inst1)

   if ((vrbl3[i]>=lst3) && vrbl3[i]<=(lst3+rng3))

{

       cal1++;

}

        }

g.fillRect(x+200,y+(k\*10),c1\*10,20);

cal1=0;

k++;

lst3=lst3+rng3;

   }  // Close of While for vrbl 3

   k=40;

   g.drawString("Vrbl4", 10, 450);

   while(lst4<=hst4)

   {

for (int i = 0; i < vrbl4.length; i++)

{

//if (vrbl4[i]==inst1)

   if ((vrbl4[i]>=lst4) && vrbl4[i]<=(lst4+rng4))

{

       cal1++;

}

        }

g.fillRect(x+200,y+(k\*10),cal1\*10,20);

cal1=0;

k++;

lst4=lst4+rng4;

   }  // Close of While for vrbl 4

           }

catch (IOException e)

{

}

}

public double [] getarr1() throws IOException

{

int val =0,fval=0,i=0;

int mt=0;

double []vrbl1=new double[150];

String csvFile = "G:/kent/Big Data/Assignments/Assignment 1/irisdata.csv";

BufferedReader br = null;

String line = "";

String cvsPartBy = ",";

int count = 0,count1 = 0;

double sum=0,sum1=0;

br = new BufferedReader(new FileReader(csvFile));

while ((line = br.readLine()) != null)

{

             System.out.println("i"+i);

             //i=i+1;

       // use comma as separator

String[] atg = line.Part(cvsPartBy);

//System.out.println("Atg [code= " + atg[4]

                              //  + " , name=" + atg[5] + "]");

vrbl1[i]=Double.parseDouble(atg[0]);

i=i+1;

}

for (i = 0; i < vrbl1.length; i++)

{

for (int j = i + 1; j < vrbl1.length; j++)

{

if (vrbl1[i] > vrbl1[j])

{

vrbl1[j]=vrbl1[i]+vrbl1[j]-(vrbl1[i]=vrbl1[j]);

}

}

}

return vrbl1;

}

public double [] getarr2() throws IOException

{

int val =0,fval=0,i=0;

int mt=0;

double []vrbl1=new double[150];

String csvFile = "G:/kent/Big Data/Assignments/Assignment 1/irisdata.csv";

BufferedReader br = null;

String line = "";

String cvsPartBy = ",";

int count = 0,count1 = 0;

double sum=0,sum1=0;

br = new BufferedReader(new FileReader(csvFile));

while ((line = br.readLine()) != null)

{

             System.out.println("i"+i);

             //i=i+1;

       // use comma as separator

String[] atg = line.Part(cvsPartBy);

//System.out.println("Atg [code= " + atg[4]

                              //  + " , name=" + atg[5] + "]");

vrbl1[i]=Double.parseDouble(atg[1]);

i=i+1;

}

for (i = 0; i < vrbl1.length; i++)

{

for (int j = i + 1; j < vrbl1.length; j++)

{

if (vrbl1[i] > vrbl1[j])

{

vrbl1[j]=vrbl1[i]+vrbl1[j]-(vrbl1[i]=vrbl1[j]);

}

}

}

return vrbl1;

}

public double [] getarr3() throws IOException

{

int val =0,fval=0,i=0;

int mt=0;

double []vrbl1=new double[150];

String csvFile = "G:/kent/Big Data/Assignments/Assignment 1/irisdata.csv";

BufferedReader br = null;

String line = "";

String cvsPartBy = ",";

int count = 0,count1 = 0;

double sum=0,sum1=0;

br = new BufferedReader(new FileReader(csvFile));

while ((line = br.readLine()) != null)

{

             System.out.println("i"+i);

             //i=i+1;

       // use comma as separator

String[] atg = line.Part(cvsPartBy);

//System.out.println("Atg [code= " + atg[4]

                              //  + " , name=" + atg[5] + "]");

vrbl1[i]=Double.parseDouble(atg[2]);

i=i+1;

}

for (i = 0; i < vrbl1.length; i++)

{

for (int j = i + 1; j < vrbl1.length; j++)

{

if (vrbl1[i] > vrbl1[j])

{

vrbl1[j]=vrbl1[i]+vrbl1[j]-(vrbl1[i]=vrbl1[j]);

}

}

}

return vrbl1;

}

public double [] getarr4() throws IOException

{

int val =0,fval=0,i=0;

int mt=0;

double []vrbl1=new double[150];

String csvFile = "G:/kent/Big Data/Assignments/Assignment 1/irisdata.csv";

BufferedReader br = null;

String line = "";

String cvsPartBy = ",";

int count = 0,count1 = 0;

double sum=0,sum1=0;

br = new BufferedReader(new FileReader(csvFile));

while ((line = br.readLine()) != null)

{

             System.out.println("i"+i);

             //i=i+1;

       // use comma as separator

String[] atg = line.Part(cvsPartBy);

//System.out.println("Atg [code= " + atg[4]

                              //  + " , name=" + atg[5] + "]");

vrbl1[i]=Double.parseDouble(atg[3]);

i=i+1;

}

for (i = 0; i < vrbl1.length; i++)

{

for (int j = i + 1; j < vrbl1.length; j++)

{

if (vrbl1[i] > vrbl1[j])

{

vrbl1[j]=vrbl1[i]+vrbl1[j]-(vrbl1[i]=vrbl1[j]);

}

}

}

return vrbl1;

}

}