

import java.text.DecimalFormat;

import java.util.StringTokenizer;

import javax.swing.JOptionPane;

public class Main {

public static void main(String[] args) {

String in, out;

in = JOptionPane.showInputDialog("enter data");

String delim= ",";

String token="" ;

String pattern;

//create a string tokenizer. //delimter is the seperator.

StringTokenizer stk = new StringTokenizer(in, delim);

//ask to get the tokens.

int count = stk.countTokens();

double [] x = new double [count];

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

token = stk.nextToken();

token = token.trim(); //trim removes spaces

x[i] = Double.parseDouble(token);

}

//Input the number of decimal places

in = JOptionPane.showInputDialog("Enter the Number of Places For Decimal Output");

int placesCount = Integer.parseInt (in);

//Build pattern string

pattern = ".0";

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

pattern = pattern + "0";

//Create a DecimalFormat object. Pass it the pattern string.

DecimalFormat df = new DecimalFormat ( pattern);

//Call format method of DecimalFormat object to convert a double into formatted string.

//double min = 99.999999;

//object // class object = new class()

Statistics stat = new Statistics (x);

//calling the objects

double min = stat.findMin();

double max = stat.findMax();

double mean = stat.findMean();

double median = stat.findMedian();

double[] sortedData = stat.getSortedData();

double[] origData = stat.getorigData();

//calling the objects

//data type variable = class. method()

double min2 = Statistics.computeMin(x);

double max2 = Statistics.computeMax(x);

double mean2 = stat.computeMean(x);

double median2 = stat.computeMedian(x);

double[] sortedData2 = stat.computeSorted(x);

//double[] origData2 = stat.getorigDatax();

/\*

int dplaces = 3;

String pattern = ",";

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

pattern = pattern + ".0";

}

DecimalFormat df = new DecimalFormat(pattern);

DecimalFormat df2 = new DecimalFormat(".000");

/\*

String sMin = df.format(min);

String sMax = df.format(max);

String sMean = df.format(mean);

String sMedian = df.format(median);

String sMin = df.format(min);

\*/

out ="original data\n";

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

{

out = out + origData[i]+ " ";

}

out = out + "\n";

out = out + "sorted data\n";

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

{

out = out + sortedData[i]+ " ";

}

//static methods

out = out + "\n";

out = out + "\nusing public methods\n";

out = out + "\n";

out = out + "sorted data\n";

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

{

out = out + sortedData[i]+ " ";

}

//pass the double throught he fomrat

out = out + "\nmin: " + df.format(min) + "\n";

out = out + "max: " + df.format(max) + "\n";

out = out + "mean: " + df.format(mean) + "\n";

out = out + "median: " + df.format(median) + "\n\n";

//public methods

out = out + "using static methods\n" ;

out = out + "\n";

out = out + "sorted data\n";

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

{

out = out + sortedData2[i]+ " ";

}

//pass the double throught he fomrat

out = out + "\nmin: " + df.format (min2) + "\n";

out = out + "max: " + df.format(max2) + "\n";

out = out + "mean: " + df.format(mean2) + "\n";

out = out + "median: " + df.format(median2) + "\n\n";

out = out + "number of times the object was passed " + Statistics.count;

JOptionPane.showConfirmDialog(null, out);

}

}

import java.util.Arrays;

public class Statistics {

//instance variable array

private double [] data;

//another array for the sorting

private double []sdata;

public static int count = 0;

//constructor of the class to recieve (ref variabel,

public Statistics (double [] d){

data = d.clone(); //pass the array. now it is just a copy. only pass the aray

sdata = d.clone();

Arrays.sort(sdata); //sort the data

Statistics.count++; //static.count. for counting the objects

}

public double findMin() {

return sdata[0];

}

public double findMax() {

return (sdata[sdata.length-1]);

}

public double findMean(){

double sum =0;

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

{

sum = sum + sdata[i];

}

double mean = sum / sdata.length;

return mean;

}

public double findMedian(){

double median=0;

//Declare variables

int index, indexHi, indexLo;

//Determine if the length is odd or even.

if ( (sdata.length %2) != 0 )

{

index = sdata.length / 2;

median = sdata [index];

}

else

{

indexHi = sdata.length / 2;

indexLo = indexHi - 1;

median = (sdata[indexLo] + sdata[indexHi] ) / 2;

}

return median;

}

public double [] getSortedData(){

return sdata;

}

public double[] getorigData() {

return data;

}

// static methods. they use the objects.

public static double computeMin(double[] d){

Statistics stat = new Statistics(d);

double min = stat.findMin();

return min;

}

public static double computeMax(double[] d){

Statistics stat = new Statistics(d);

double max = stat.findMax();

return max;

}

//2 static methods. they use the objects

public static double computeMean(double[] d){

Statistics stat = new Statistics(d);

double mean = stat.findMean();

return mean;

}

public static double computeMedian(double[] d){

Statistics stat = new Statistics(d);

double median = stat.findMedian();

return median;

}

public static double[] computeSorted(double[] d){

Statistics stat = new Statistics(d);

double[] sortedData = stat.getSortedData();

return sortedData;

}

}