流程圖:



說明檔:

宣告一個scanner物件，此變數可以掃描從螢幕輸入的資料，存成變數

輸出"親愛的，請輸入積分範圍之上下限" + 換行 + "上限值為:"

宣告字串u,掃描輸入上限值

輸出"下限值為:"

宣告字串l,掃描輸入下限值

輸出"請輸入積分之區間數" + 換行 + "區間數為:"

宣告字串C,掃描輸入區間數

輸出"函數一"

輸出"選擇矩形法輸入1，梯形法輸入2"

宣告整數decision,掃描輸入值

宣告整數n=字串C轉為整數(積分區間格數)

宣告實數x0 = 將字串l轉成數字(設定積分下限)

宣告實數xn = 將字串u轉成數字(設定積分上限)

宣告實數x = x0(設定自變項;初始值為積分下限)

宣告實數dx = (xn - x0) / n(計算積分區間間隔值)

宣告實數A = 0(函數底下面積,計算前的初始值為0)

宣告整數amount為1

如果decision = 1

//矩形法1

當x < xn(當自變項的值不超過上限時,執行計算小矩形的指令)

宣告實數y = x2 ex;

宣告實數dA = y \* dx(計算小矩形面積)

A = A + dA(逐次累加的小矩形面積)

x = x + dx(計算一個小面積後，進行下一步前先累加x值，準備下一次的計算)

x = Math.round(x \* 1000) \* 0.001(四捨五入到小數點第3位)

從1開始迴圈(每次加一次)

宣告實數g =(x2-2x+2)ex (帶入函數)

輸出"解析解: " + g + "數值解: " + A

又如果decision = 2

//梯形法2

用do...while迴圈，一定要做第一次

宣告實數y = x2 ex (帶入積分函數，計算上底長度)

宣告實數yy = x2 e(x +dx)(計算下底長度)

宣告實數dA = (y + yy) \* dx / 2(計算小梯形面積)

A = A + dA(逐次累加的小矩形面積)

x = x + dx(計算一個小面積後，進行下一步前先累加x值，準備下一次的計算)

x = Math.round(x \* 1000) \* 0.001(四捨五入到小數點第3位)

從1開始迴圈(每次加一次)

當x < xn時，執行下列程式

宣告實數y = (x2-2x+2)ex(帶入函數)

輸出"解析解: " + y + "數值解: " + A

其他

輸出"搗蛋鬼誰准你輸入1跟2以外的數字"

跳出迴圈

//斜率

宣告整數n=字串R轉為整數(積分區間格數)

宣告實數x0 = 將字串d轉成數字(設定積分下限)

宣告實數xn = 將字串u轉成數字(設定積分上限)

宣告實數x = x0(設定自變項;初始值為積分下限)

宣告實數dx = (xn - x0) / n(計算積分區間間隔值)

用do...while迴圈，一定要做第一次

宣告實數mg = (x2-2x+2)ex(帶入函數)

宣告實數mmg = ((x+dx)2-2(x+dx)+2)e(x+dx)(帶入函數)

宣告實數mmmg = (mmg - mg) / dx(帶入函數)

宣告實數my= x2 ex(帶入函數)

輸出"g(x)下的斜率=" + mmmg + "f(x)=" + my

x = x + dx(計算一個斜率後，進行下一步前先累加x值，準備下一次的計算)

當x < xn時，執行do裡面的程式

//作業2-2

輸出"函數二"

宣告實數x0 = 將字串l轉成數字(設定積分下限)

宣告實數xn = 將字串u轉成數字(設定積分上限)

輸出"請輸入a值(不可高於下限值)為:"

宣告實數a,掃描輸入的a值

如果a > x0

輸出"a值不再合理範圍內，請離場!”

跳出迴圈

其他

輸出"選擇矩形法輸入1，梯形法輸入2"

宣告整數choose,掃描輸入的選擇

宣告實數p = 2

宣告實數q= 3

宣告整數n=字串C轉為整數(積分區間格數)

宣告實數x = x0(設定自變項;初始值為積分下限)

宣告實數dx = (xn - x0) / n(計算積分區間間隔值)

宣告實數A = 0(函數底下面積,計算前的初始值為0)

宣告整數amount為1

如果choose = 1

當x < xn(當自變項的值不超過上限時,執行計算小矩形的指令)

宣告實數y = (x - a)0.5;

宣告實數dA = y \* dx(計算小矩形面積)

A = A + dA(逐次累加的小矩形面積)

x = x + dx(計算一個小面積後，進行下一步前先累加x值，準備下一次的計算)

x = Math.round(x \* 1000) \* 0.001(四捨五入到小數點第3位)

從1開始迴圈(每次加一次)

宣告實數g = 2/3(x-a)3/2 (帶入函數)

輸出"解析解: " + g + "數值解: " + A

又如果choose = 2

//梯形法2

用do...while迴圈，一定要做第一次

宣告實數y = (x - a)0.5 (帶入積分函數，計算上底長度)

宣告實數yy = ((x +dx)- a)0.5 (計算下底長度)

宣告實數dA = (y + yy) \* dx / 2(計算小梯形面積)

A = A + dA(逐次累加的小矩形面積)

x = x + dx(計算一個小面積後，進行下一步前先累加x值，準備下一次的計算)

x = Math.round(x \* 1000) \* 0.001(四捨五入到小數點第3位)

從1開始迴圈(每次加一次)

當x < xn時，執行下列程式

宣告實數y = 2/3(x-a)3/2(帶入函數)

輸出"解析解: " + y + "數值解: " + A

其他

輸出輸出"是不是要你不要調皮輸入1跟2以外的數字"

跳出迴圈

//斜率

宣告整數n=字串C轉為整數(積分區間格數)

宣告實數x0 = 將字串l轉成數字(設定積分下限)

宣告實數xn = 將字串u轉成數字(設定積分上限)

宣告實數x = x0(設定自變項;初始值為積分下限)

宣告實數dx = (xn - x0) / n(計算積分區間間隔值)

用do...while迴圈，一定要做第一次

宣告實數mg = 2/3(x-a)3/2(帶入函數)

宣告實數mmg = 2/3((x+dx)-a)3/2(帶入函數)

宣告實數mmmg = (mmg - mg) / dx(帶入函數)

宣告實數my= (x - a)0.5(帶入函數)

輸出"g(x)下的斜率=" + mmmg + "\t" + "f(x)=" + my

x = x + dx(計算一個斜率後，進行下一步前先累加x值，準備下一次的計算)

當x < xn時，執行do裡面的程式

程式碼:

import java.util.Scanner;

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author PTT

\*/

public class B109026309\_hw02\_2 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

Scanner sc = new Scanner(System.in);//宣告一個scanner物件，此變數可以掃描從螢幕輸入的資料，存成變數

System.out.println("親愛的,請輸入積分範圍之上下限" + "\n" + "上限值為:");//輸出"親愛的,請輸入積分範圍之上下限" + "\n" + "上限值為:"

String u = sc.nextLine();//宣告字串u,掃描輸入上限值

System.out.println("下限值為:");//輸出"下限值為:"

String l = sc.nextLine();//宣告字串l,掃描輸入下限值

System.out.println("請輸入積分之區間數" + "\n" + "區間數為:");//輸出"請輸入積分之區間數" + "\n" + "區間數為:"

String C = sc.nextLine();//設定區間數

//作業2-1

System.out.println("函數一");//輸出"函數一"

System.out.println("選擇矩形法輸入1，梯形法輸入2");//輸出"選擇矩形法輸入1，梯形法輸入2"

int decision = sc.nextInt();//宣告整數decision,掃描輸入值

{

int n = Integer.parseInt(C);//積分區間格數

double x0 = Double.parseDouble(l);//設定積分下限(將字串轉成數字)

double xn = Double.parseDouble(u);//設定積分上限(將字串轉成數字)

double x = x0;//設定自變項;初始值為積分下限

double dx = (xn - x0) / n;//計算積分區間間隔值

double A = 0;//函數底下面積,計算前的初始值為0

{

int amount = 1;//宣告整數amount為1

if (decision == 1) {

//矩形法1

while (x < xn) {//當自變項的值不超過上限時,執行計算小矩形的指令

double y = (Math.pow(x, 2)) \* Math.pow(Math.E,x);

double dA = y \* dx;//計算小矩形面積

A = A + dA;//逐次累加的小矩形面積

x = x + dx;//計算一個小面積後，進行下一步前先累加x值，準備下一次的計算

x = Math.round(x \* 1000) \* 0.001;//四捨五入到小數點第3位

amount++;//從1開始迴圈

}

double g = (Math.pow(x, 2) - (2 \* x) + 2) \* Math.pow(Math.E,x);//帶入函數

System.out.println("解析解: " + g + "數值解: " + A);//輸出"解析解: " + g + "數值解: " + A

} else if (decision == 2) {

//梯形法2

do {//使用do...while迴圈，一定要做第一次

double y = (Math.pow(x, 2)) \* Math.pow(Math.E,x);//帶入積分函數，計算上底長度

double yy = (Math.pow(x+dx, 2)) \* Math.pow(Math.E,x+dx);//計算下底長度

double dA = (y + yy) \* dx / 2;//計算小梯形面積

A = A + dA;//逐次累加的小矩形面積

x = x + dx;//計算一個小面積後，進行下一步前先累加x值，準備下一次的計算

x = Math.round(x \* 1000) \* 0.001;//四捨五入到小數點第3位

amount++;//從1開始迴圈

} while (x < xn);//當x < xn時，執行下列程式

double y = (Math.pow(x, 2) - (2 \* x) + 2) \*Math.pow(Math.E,x);//帶入函數

System.out.println("解析解: " + y + "\t" + "數值解: " + A);//輸出"解析解: " + y + "數值解: " + A

} else {

System.out.println("搗蛋鬼誰准你輸入1跟2以外的數字");//輸出"搗蛋鬼誰准你輸入1跟2以外的數字"

System.exit(0);//跳出迴圈

}

}}

//斜率

{

int n = Integer.parseInt(C);//積分區間格數

double x0 = Double.parseDouble(l);//設定積分下限(將字串轉成數字)

double xn = Double.parseDouble(u);//設定積分上限(將字串轉成數字)

double x = x0;//設定自變項;初始值為積分下限

double dx = (xn - x0) / n;//計算積分區間間隔值

double A = 0;//函數底下面積,計算前的初始值為0

do {

double mg = (Math.pow(x, 2) - (2 \* x) + 2) \* Math.pow(Math.E, x);//帶入函數

double mmg = (Math.pow(x + dx, 2) - (2 \* (x + dx)) + 2) \* Math.pow(Math.E, (x + dx));//帶入函數

double mmmg = (mmg - mg) / dx;//帶入函數

double my = Math.pow(x, 2) \* Math.pow(Math.E, x);//帶入函數

System.out.println("g(x)下的斜率=" + mmmg + "\t" + "f(x)=" + my);//輸出"g(x)下的斜率=" + mmmg + "\t" + "f(x)=" + my

x = x + dx;//計算一個斜率後，進行下一步前先累加x值，準備下一次的計算

} while (x < xn);//當x < xn時，執行do裡面的程式

}

//作業2-2

System.out.println("函數二");//輸出"函數二"

System.out.println("請輸入a值(不可高於下限值)為:");//輸出"請輸入a值(不可高於下限值)為:"

double a = sc.nextDouble();//宣告實數a,掃描輸入的a值

double x0 = Double.parseDouble(l);//設定積分下限(將字串轉成數字)

double xn = Double.parseDouble(u);//設定積分上限(將字串轉成數字)

if (a > x0) {//如果a > x0

System.out.println("a值不在合理範圍內，請離場!");//輸出"a值不再合理範圍內，請離場!

System.exit(0);//跳出迴圈

} else {

System.out.println("選擇矩形法輸入1，梯形法輸入2");//輸出"選擇矩形法輸入1，梯形法輸入2"

}

int choose = sc.nextInt();//宣告整數choose,掃描輸入的選擇

{

double p = 2;//宣告實數p為2

double q = 3;//宣告實數q為3

int n = Integer.parseInt(C);//積分區間格數

double x = x0;//設定自變項;初始值為積分下限

double dx = (xn - x0) / n;//計算積分區間間隔值

double A = 0;//函數底下面積,計算前的初始值為0

int amount = 1;//宣告整數amount為1

if (choose == 1) {

//矩形法1

while (x < xn) {//當自變項的值不超過上限時,執行計算小矩形的指令

double y = Math.pow(x - a, 0.5);

double dA = y \* dx;//計算小矩形面積

A = A + dA;//逐次累加的小矩形面積

x = x + dx;//計算一個小面積後，進行下一步前先累加x值，準備下一次的計算

x = Math.round(x \* 1000) \* 0.001;//四捨五入到小數點第3位

amount++;//從1開始迴圈

}double g = (p / q) \* Math.pow((x - a), 1.5);//帶入函數

System.out.println("解析解: " + g + "\t" + "數值解: " + A);

} else if (choose == 2) {

//梯形法2

do {//使用do...while迴圈，一定要做第一次

double y = Math.pow(x - a, 0.5);//帶入積分函數，計算上底長度

double yy = Math.pow((x+dx) - a, 0.5) ;//計算下底長度

double dA = (y + yy) \* dx / 2;//計算小梯形面積

A = A + dA;//逐次累加的小梯形面積

x = x + dx;//計算一個小面積後，進行下一步前先累加x值，準備下一次的計算

x = Math.round(x \* 1000) \* 0.001;//四捨五入到小數點第3位

amount++;

} while (x < xn);//當x < xn時，執行下列程式

double y = (p / q) \* Math.pow((x - a), 1.5);//帶入函數

System.out.println("解析解: " + y + "數值解: " + A);//輸出"解析解: " + y + "數值解: " + A

} else {

System.out.println("是不是要你不要調皮輸入1跟2以外的數字!");//輸出"是不是要你不要調皮輸入1跟2以外的數字!"

System.exit(0);//跳出迴圈

}

}

//斜率

double p = 2;//宣告實數p為2

double q = 3;//宣告實數q為3

int n = Integer.parseInt(C);//積分區間格數

double x = x0;//設定自變項;初始值為積分下限

double dx = (xn - x0) / n;//計算積分區間間隔值

do {

double mg = (p / q) \* Math.pow((x - a), 1.5);//帶入函數

double mmg = (p / q) \* Math.pow(((x + dx) - a), 1.5);//帶入函數

double mmmg = (mmg - mg) / dx;//帶入函數

double my = Math.pow(x - a, 0.5);//帶入函數

System.out.println("g(x)下的斜率=" + mmmg + "\t" + "f(x)=" + mg);//輸出"g(x)下的斜率=" + mmmg + "\t" + "f(x)=" + my

x = x + dx;//計算一個斜率後，進行下一步前先累加x值，準備下一次的計算

} while (x < xn);//當x < xn時，執行do裡面的程式

}

}