class TestArray:

**begin**

arrayInit()

ArrayCreator arr = new ArrayCreator();

double[][] massiv = arr.arrayInit();

double[][] arrTest = arr.test();

int m, n; //Array size

double minVal, maxVal; //range value

//maxVal > minVal

new double[m][n] arr;

arr[i][j] = minVal + (maxVal - minVal) \* random.nextDouble()

ArrayMethods(massiv);

ArrayMethods(arrTest);

arr [][]

**end**

**end**

class ArrayCreator:

test()

**begin**

**end**

int m, n; //Array size

double minVal, maxVal; //range value

//maxVal > minVal

public double[][] arrayInit()

public double[][] test()

double[][] arrayTest = {{1,1,1,0,1,1},

1,0,1,1,2,1},

{1,1,1,1,1,1},

{1,0,1,3,1,1},

{1,1,1,1,1,1},

{1,1,1,1,1,1}};

**end**

Class ArrayMethods:

double[] row: a

double elem: row

double a [][]

sum=0; multipl = 1;

lengthArray=a.length\* \*a[0].length;

meanArithGeom()

end

elem

row[]

elem

Y

N

Y

N

minMaxArr ()

end

elem < min

min = elem

double a [][]

min = max = a[0][0]

double[] row: a

double elem: row

elem > max

max = elem

min

max

elem

row[]

double elem: row

double[] row: a

double a [][]

min = max = a[0][0]

printArray ()

sum += elem;

multipl \*= elem;

elem

row[]

sum /= lengthArray;

Y

(multipl < 0) && (lengthArray % 2 == 0)

N

Math.pow(multipl, 1.0/lengthArray)

end

Complex number

N

Y

end

b[j][i] = a[i][j]

new double b [a[0].length][a.length]

int i = 0;

i < a.length ;

i++

int j = 0;

i < a.length[0] ;

j++

j

i

b[][]

a[][]

j

i

temp = a[i][j];

a[i][j] = a[j][i];

a[j][i] = temp;

int i = 1;

i < a.length - 1;

i++

int j = I + 1;

i < a.length[0] - 1;

j++

a.length == a[0].length

double a [][]

transpon ()

Y

N

j

i

localMin, iFirstMin,

jFirstMin;

end

localMin = a[i][j];

iFirstMin = i;

jFirstMin = j;

i = a.length-1;

j = a[0].length-1;

localMin () //localMax()

double a [][]

double localMin = -1;

int iFirstMin = -1;

int jFirstMin = -1;

int i = 1;

i < a.length - 1;

i++

int j = 1;

i < a.length[0] - 1;

j++

(a[i][j] < a[i][j-1]) &&

(a[i][j] < a[i-1][j]) &&

(a[i][j] < a[i+1][j]) &&

(a[i][j] < a[i][j-1])