

# А л г о р и т м и   т а   с т р у к т у р и   д а н и х .

---

## О с н о в и   а л г о р и т м і з а ц і ї

*Додаток 1*

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації  
і управління

Звіт

з лабораторної роботи №9  
з дисципліни «Алгоритми та структури даних-1.  
Основи алгоритмізації»  
«ДСЛІДЖЕННЯ АЛГОРИТМІВ ОБХОДУ МАСИВІВ»  
Варіант 9

Виконав  
студент

ІП-13, Гончаров Євген Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірів

Наталія Вечерковська Сергіївна

( прізвище, ім'я, по батькові)

# А л г о р и т м и   т а   с т р у к т у р и   д а н и х .

## О с н о в и   а л г о р и т м і з а ц і ї

### Лабораторна робота 9

#### Дослідження алгоритмів пошуку та сортування

**Мета** – дослідити алгоритми обходу масивів, набуті практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Індивідуальне завдання

#### Варіант 8

#### Завдання

8	Задано матрицю дійсних чисел $A[n,n]$ , ініціалізувати матрицю обходом по рядках. На головній діагоналі матриці знайти перший від'ємний і останній додатний елементи, та поміняти їх місцями з елементами побічної діагоналі .
---	--

### 1. П о с т а н о в к а   з а д а ч і

Введемо висоту та ширину двовірного масиву і, оскільки ми працюємо з дійсними числами, введемо к-ть знаків після крапки. Згенеруємо матрицю і заповнимо її випадковими числами(включно мінусовими) за допомогою спеціальної ф-ї, результат запишемо. Наступним кроком зманимо місцями перший від'ємний та останній додатний елементи, якщо таких нема – виведемо відповідне повідомлення.

### 2. П о б у д о в а   м а т е м а т и ч н о ї   м о д е л і.

С к л а д е м о   т а б л и ц ю   і м е н   з м і н н и х .

Змінна	Тип	Ім'я	Призначення
Змінна height	Цілочисельний	height	Вхідні данні
Змінна weight	Цілочисельний	weight	Вхідні данні
Змінна depth	Цілочисельний	depth	Вхідні данні
Змінна matrix	Дійсночисельний[][]	matrix[][]	Проміжні та кінцеві данні
Змінна generateDoubleMass	Функція(цілочисельний)	generateDoubleMass (height, weight, depth)	Обчислення
Змінна round	Функція(дійсночисельний)	Round(x,depth)	Обчислення
Змінна changeFirstMinusAndLastPlus	Функція(дійсночисельний[][])	changeFirstMinusAndLastPlus (matrix[][])	Обчислення

Опишемо генерацію випадкового цілого числа від max до min через базову функцію  $\text{Math.random()} * (\text{max} - \text{min} + 1) + \text{min}$

Опишемо піднесення до степеня  $x^y$  як  $\text{Math.pow}(x,y)$

Опишемо знаходження довжини масиву через `Array.length`

#### Опис підпрограм

`generateDoubleMass (height, weight, depth)` – генерує матрицю випадкових чисел(дійсних).

# А л г о р и т м и   т а   с т р у к т у р и   д а н и х .

---

## О с н о в и   а л г о р и т м і з а ц і ї

`round(x, depth)` – округляє `x` до `depth` знаків після крапки .

`changeFirstminusAndLastPlus(matrix[][])` – змінює перший від'ємний та останній додатний елементи головної діагоналі матриці місцями.

### Р о з в' я з а н н я

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначемо основні дії

Крок 2. Введемо змінні `height`, `weight` та `depth`

Крок 3. Запишемо в змінну `matrix` результат виконання `generateDoubleMass(height, weight, depth)`

Крок 4. Запишемо у змінну `matrix` результат виконання `changeFirstMinusAndLastPlus (matrix[][])`

Крок 5. Виведемо `matrix`

### П с е в д о к о д

#### Основна програма:

##### Початок

Визначемо основні дії

Введення `height`

Введення `weight`

Введення `depth`

`matrix = generateDoubleMass(height, weight, depth)`

`matrix = changeFirstMinusAndLastPlus (matrix[][])`

Виведення `matrix`

##### Кінець

### Підпрограми:

```
generateDoubleMass(height, weight, depth)
    дійснoчисельний array[][]
    повторити для i від 0 до height:
        повторити для j від 0 до weidht:
            array[i][j] = round(Math.random()*198 - 99,
depth)
        все повторити
    все повторити
    повернути array
кінець generateDoubleMass

changeFirstMinusAndLastPlus(matrix)
    дійснoчисельний firstMinus = 0.0
    indexMinus = 0
    lastPlus = 0.0
    indexPlus = 0
    повторити для i від 0 до matrix.length:
        якщо matrix[i][i] < 0 та firstMinus == 0.0 то:
            firstMinus = matrix[i][i]
            indexMinus = i
        інакше якщо matrix[i][i] > 0 то:
            lastPlus = matrix[i][i]
            indexPlus = i
    все якщо
    все повторити
    якщо firstMinus == 0.0 то:
        Вивести "Немає від'ємних елементів на головній
діагоналі матриці!"
        повернути matrix
    інакше якщо lastPlus == 0.0 то:
        Вивести "Немає додатніх елементів на головній
діагоналі матриці!"
        повернути matrix
    все якщо
    matrix[indexMinus][indexMinus] +=
matrix[indexPlus][indexPlus]
```

# Алгоритми та структури даних.

---

## Основи алгоритмізації

```
matrix[indexPlus][indexPlus] =  
matrix[indexMinus][indexMinus] -  
matrix[indexPlus][indexPlus]  
matrix[indexMinus][indexMinus] =  
matrix[indexMinus][indexMinus] -  
matrix[indexPlus][indexPlus]  
повернути matrix;
```

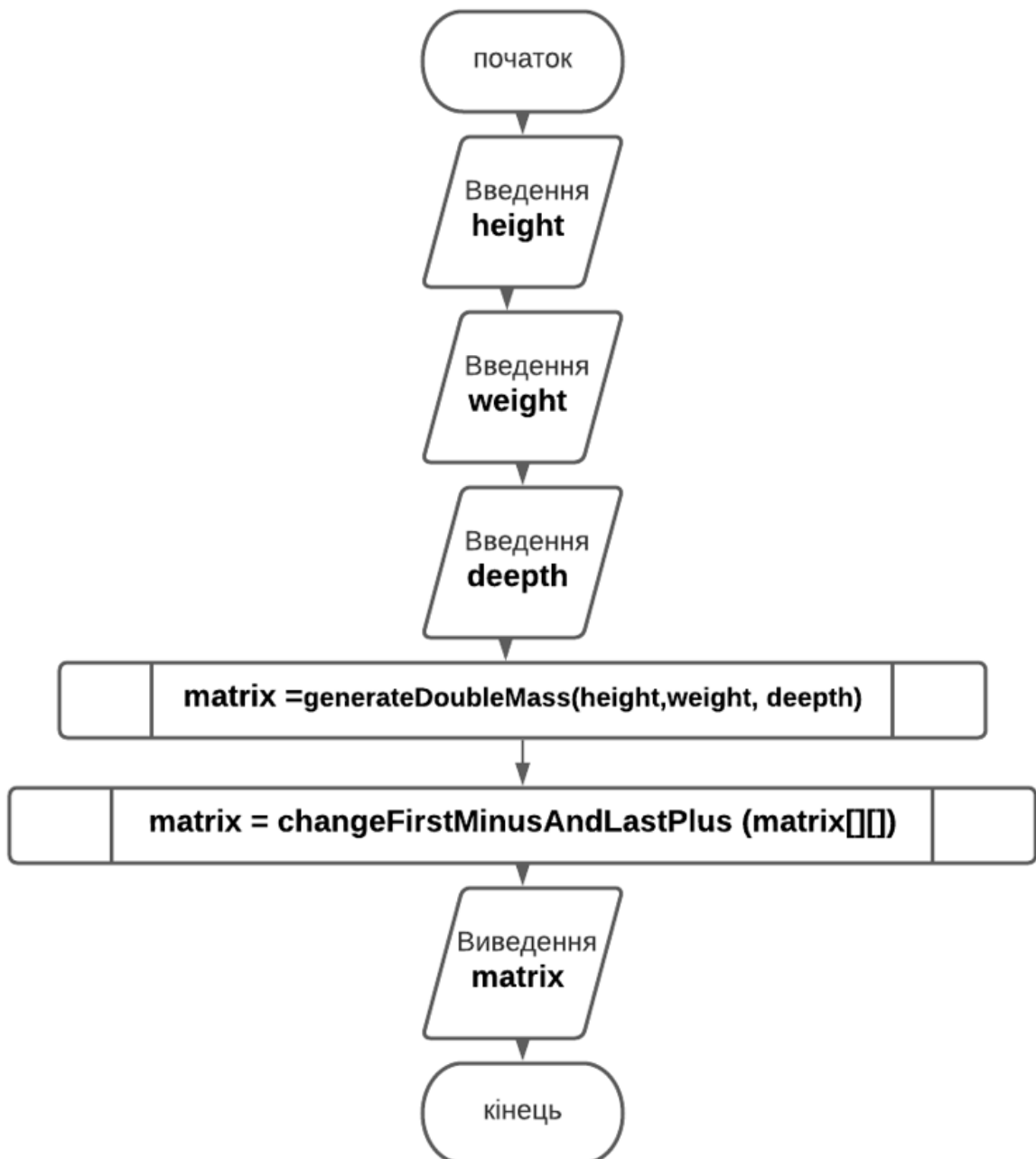
кінець changeFirstMinusAndLastPlus

```
round(x, depth)  
    x = цілочисельний(x*Math.pow(10, depth))  
    x /= Math.pow(10,depth)  
    повернути x  
кінець round
```

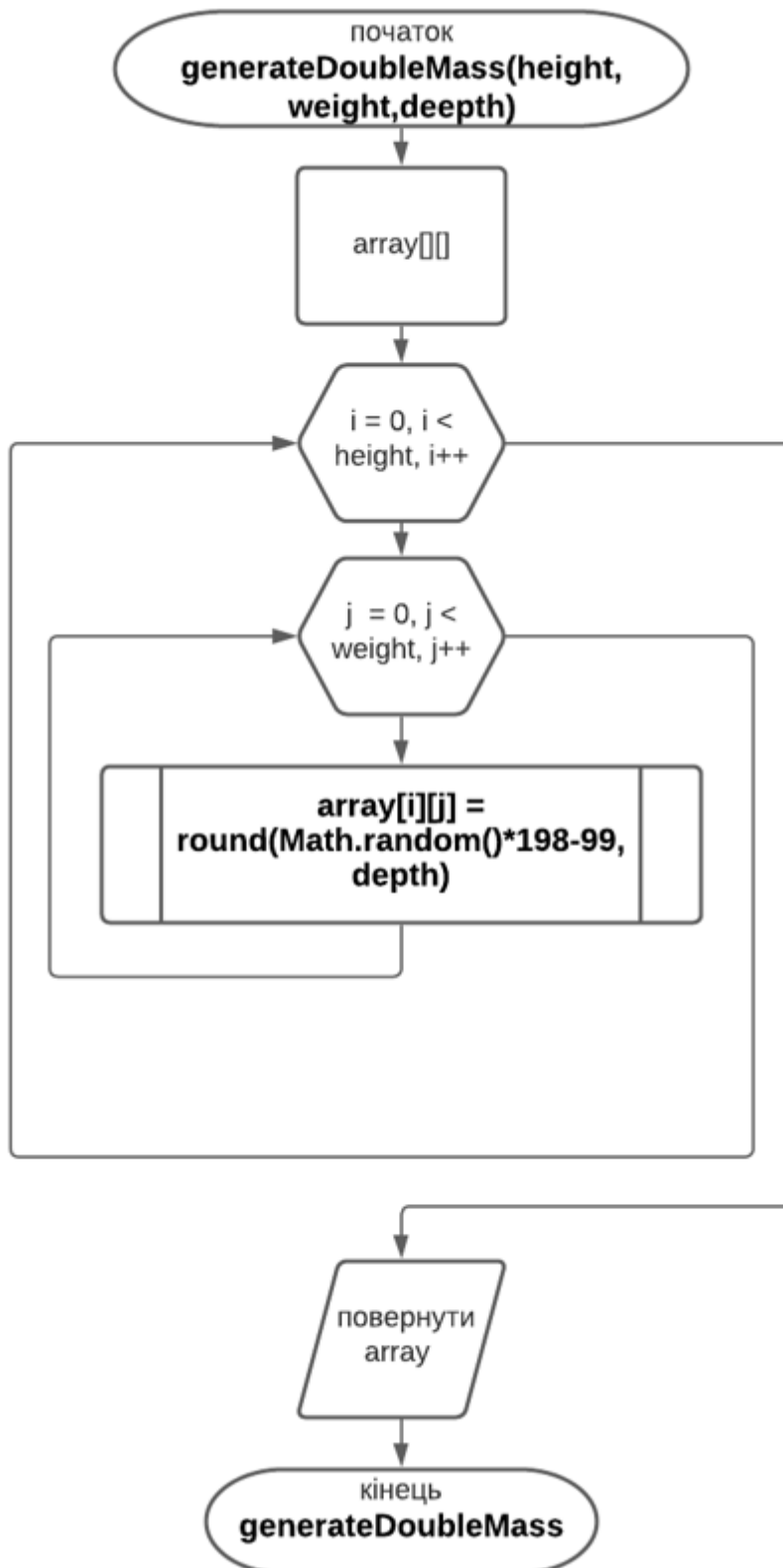
## Основи алгоритмізації

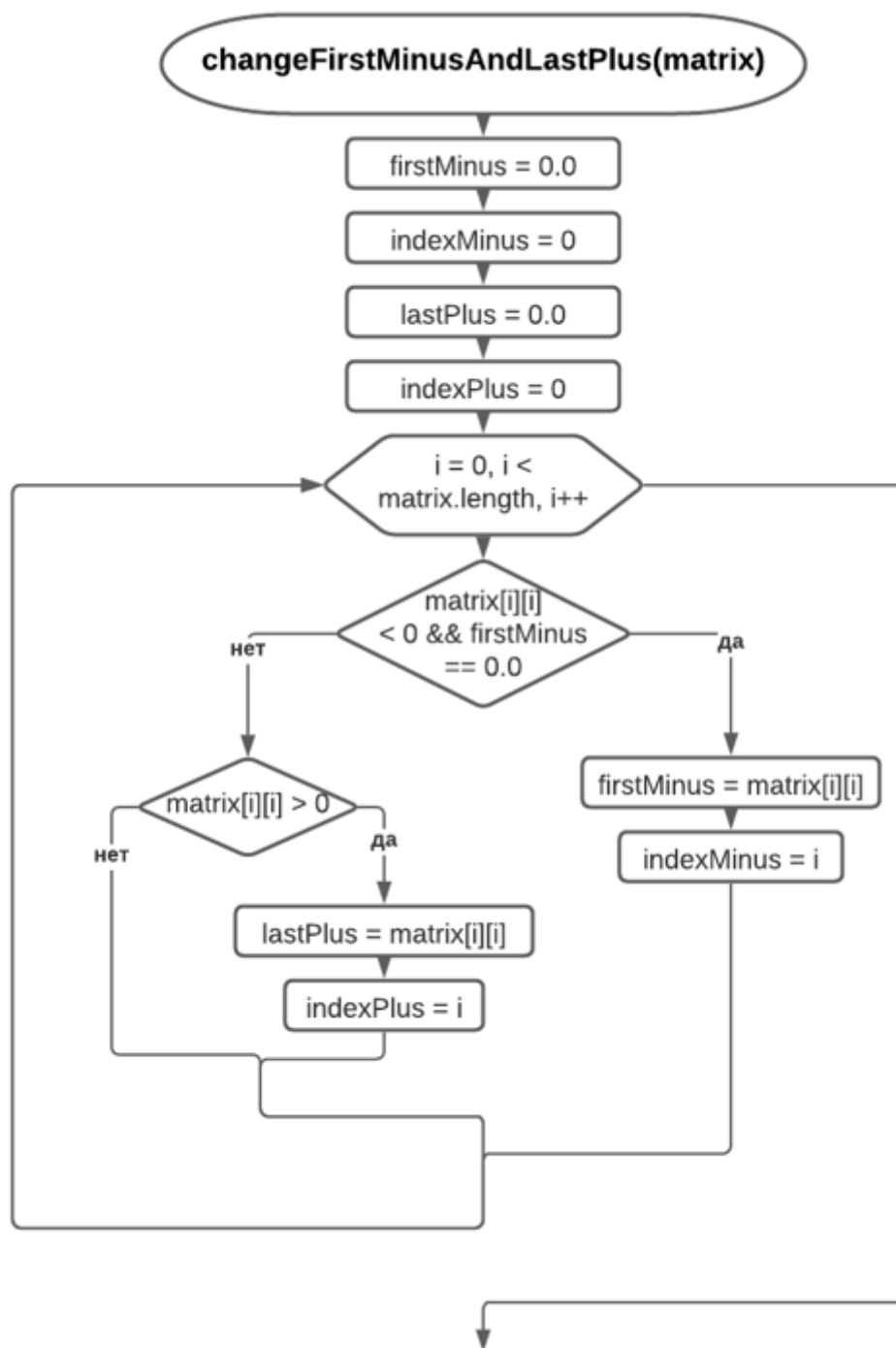
### Блок-схема

Основна програма:

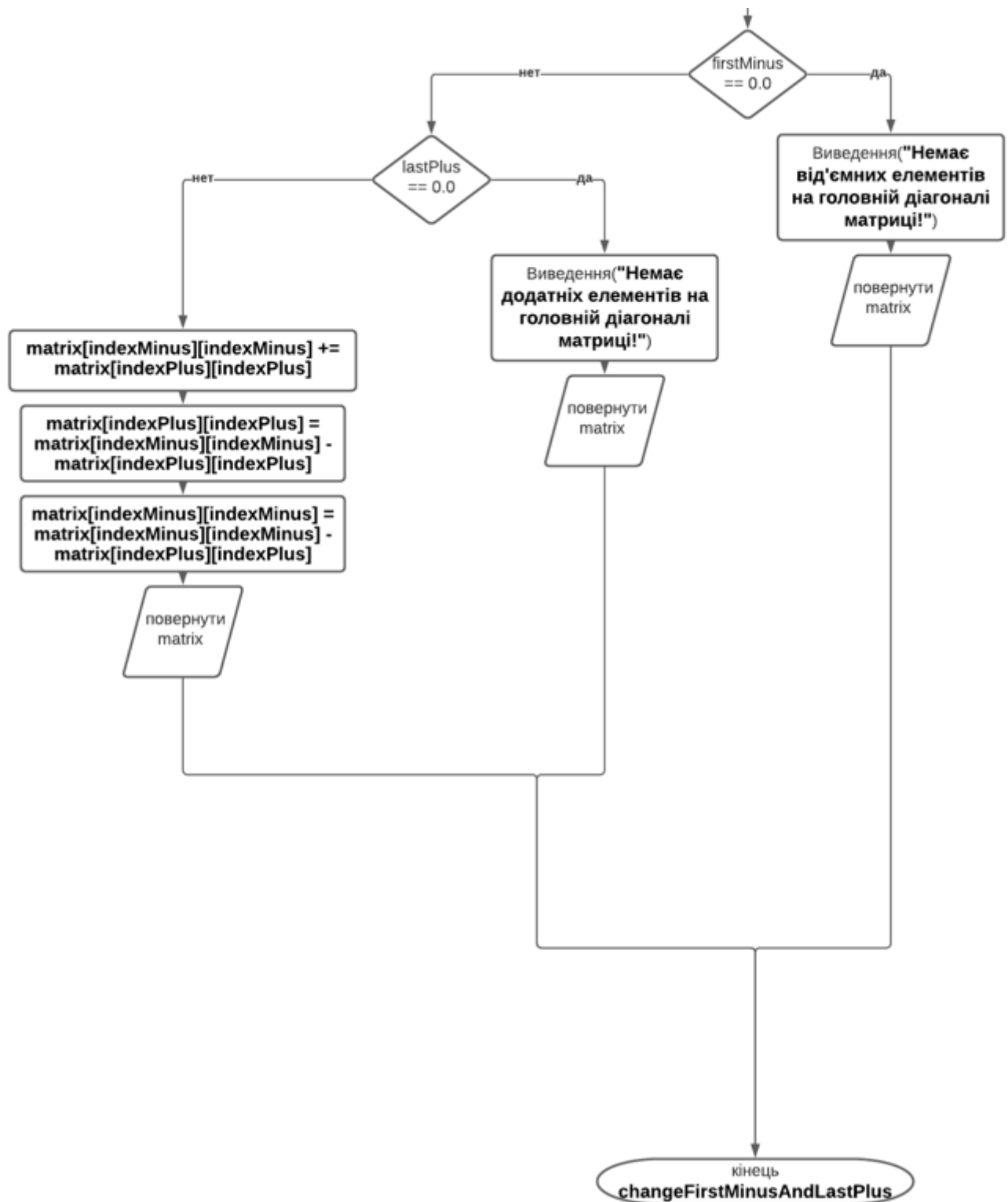


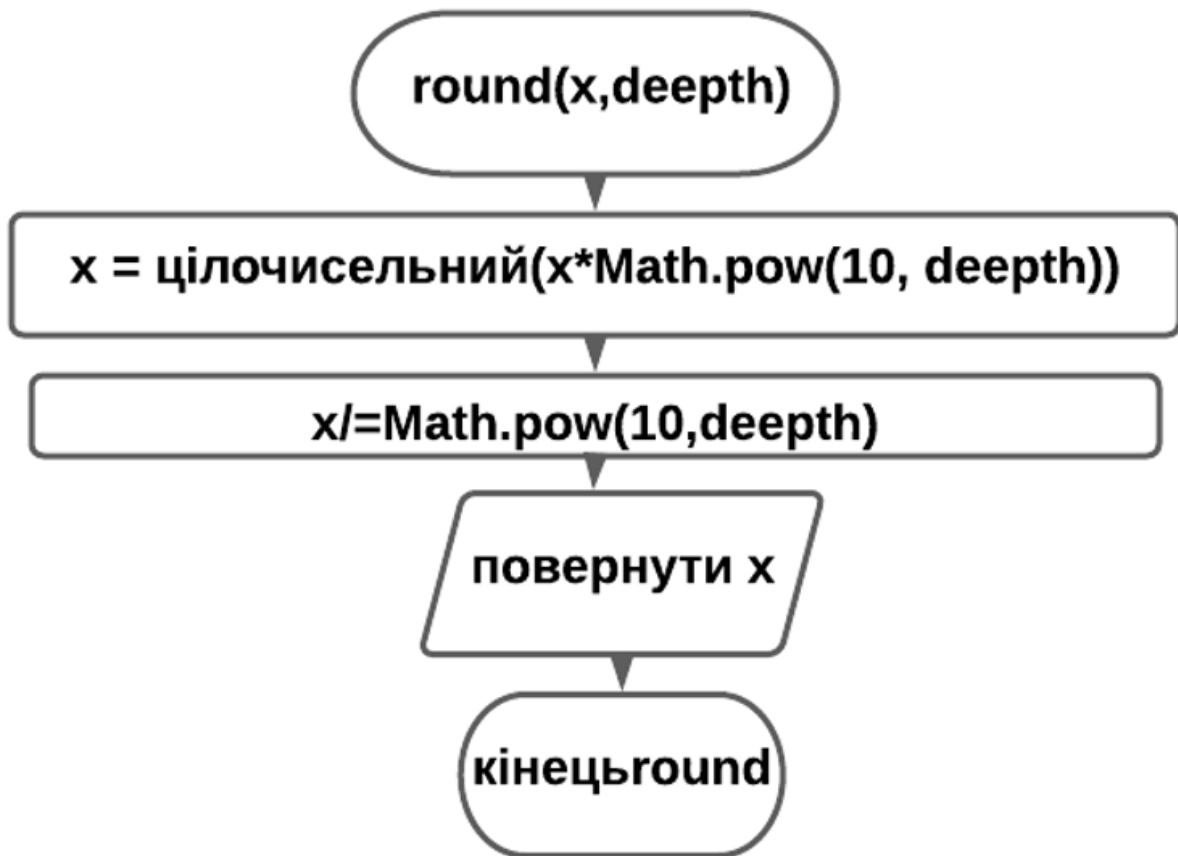
## Підпрограми:











### Тестування

```
LabaASD_9 x
C:\Users\gonch\.jdk\openjdk-16.0.2\bin\java.
Введіть висоту масиву: 5
Введіть ширину масиву: 5
Введіть к-ть знаків після коми: 2

Згенерований масив
| -45.91  -79.04   63.09  98.11  62.91 |
| -8.49   -51.45   49.29 -93.73  -31.73 |
| 80.55  -14.88  -81.06  75.32  18.54 |
| -3.74   -93.43  -20.87  46.15 -83.26 |
| -7.83    6.16  -47.18  -46.79 -31.75 |

Оброблений масив
| 46.15  -79.04   63.09  98.11  62.91 |
| -8.49   -51.45   49.29 -93.73  -31.73 |
| 80.55  -14.88  -81.06  75.32  18.54 |
| -3.74   -93.43  -20.87  -45.91 -83.26 |
| -7.83    6.16  -47.18  -46.79 -31.75 |
```

# Алгоритми та структури даних.

## Основи алгоритмізації

### Код

```
LabaASD_9.java x
1  import java.util.Scanner;
2
3  public class LabaASD_9 {
4
5      public static void main(String[] args) {
6          Scanner in = new Scanner(System.in);
7          System.out.print("Введіть висоту масиву: ");
8          int height = in.nextInt();
9          System.out.println();
10         System.out.print("Введіть ширину масиву: ");
11         int weight = in.nextInt();
12         System.out.println();
13         System.out.print("Введіть к-ть знаків після коми: ");
14         int depth = in.nextInt();
15         System.out.println();
16         System.out.println("Згенерований масив");
17         double[][] matrix = generateDoubleMass(height, weight, depth);
18         printDoubleMass(matrix, depth);
19         System.out.println("Оброблений масив");
20         printDoubleMass(changeFirstMinusAndLastPlus(matrix), depth);
21     }
22 }
```

```
24  @ public static double[][] generateDoubleMass(int height, int weight, int depth){
25      double[][] array = new double[height][weight];
26
27      for(int i = 0; i < height; i++){
28          for(int j = 0; j < weight; j++){
29              array[i][j] = round(Math.random()*(198) - 99, depth);
30          }
31      }
32      return array;
33  }
```

# Алгоритми та структури даних.

## Основи алгоритмізації

```
35  @ public static double[][] changeFirstMinusAndLastPlus(double[][] matrix){
36      double firstMinus = 0.0;
37      int indexMinus = 0;
38      double lastPlus = 0.0;
39      int indexPlus = 0;
40      for(int i = 0; i < matrix.length; i++){
41          if(matrix[i][i] < 0 && firstMinus == 0.0){
42              firstMinus = matrix[i][i];
43              indexMinus = i;
44          }
45          else if(matrix[i][i] > 0){
46              lastPlus = matrix[i][i];
47              indexPlus = i;
48          }
49      }
50      if(firstMinus == 0.0){
51          System.out.println("Немає від'ємних елементів на головній діагоналі матриці!");
52          return matrix;
53      }else if(lastPlus == 0.0){
54          System.out.println("Немає додатних елементів на головній діагоналі матриці!");
55          return matrix;
56      }
57      matrix[indexMinus][indexMinus] += matrix[indexPlus][indexPlus];
58      matrix[indexPlus][indexPlus] = matrix[indexMinus][indexMinus] - matrix[indexPlus][indexPlus];
59      matrix[indexMinus][indexMinus] = matrix[indexMinus][indexMinus] - matrix[indexPlus][indexPlus];
60      return matrix;
61  }
62 }
```

```
64      public static double round(double x, double depth){
65          x = (int)(x*(Math.pow(10,depth)));
66          x = x / Math.pow(10,depth);
67          return x;
68      }
```

# Алгоритми та структури даних.

## Основи алгоритмізації

```
70 @ public static void printDoubleMass(double[][] array, int depth){
71     for(int i = 0; i < array.length; i++){
72         System.out.print("| ");
73         for(int j = 0; j < array[i].length; j++){
74
75             int counter = 0;
76             double clone = array[i][j];
77             while(clone * Math.pow(10, counter) % 1 != 0){
78                 counter++;
79             }
80             if(clone >= 0){
81                 System.out.print(" ");
82             }
83             if(counter + 1 == depth){
84                 System.out.print(clone + " ");
85             }else if(counter < depth) {
86                 depth -= counter;
87                 System.out.print(clone);
88                 while (depth !=0){
89                     System.out.print(" ");
90                     depth--;
91                 }
92                 System.out.print(" ");
93             }else{
94                 System.out.print(clone + " ");
95             }
96             if(clone < 10){
97                 System.out.print(" ");
98             }
99
100     }
101     System.out.println("|");
102 }
103 }
```

# А л г о р и т м и   т а   с т р у к т у р и   д а н и х .

---

## О с н о в и   а л г о р и т м і з а ц і ї

### **Висновки**

Ми дослідили алгоритми обходу масивів, та набути практичних навичок їх використання під час складання програмних специфікацій.

В результаті виконання лабораторної роботи ми отримали алгоритм для знаходження за зміни місцями першого від'ємного та останнього додатнього елементів головної діагоналі матриці місцями, декомпозували задачу на 5 кроків: визначили основні дії, ввели данні для генерації масиву, згенерували масив, знайшли перший від'ємний та останній додатній елементи головної діагоналі матриці, змінили їх місцями, результат вивели.