# **Національний технічний університет України**

# **«Київський політехнічний інститут»**

# **Факультет інформатики та обчислювальної техніки**

# **Кафедра обчислювальної техніки**

# **Лабораторна робота №4**

### **з програмування**

виконав студент групи

Номер залікової книжки:

**Тема:** Масиви у мові програмування Java.

**Мета:** Ознайомлення з масивами в Java. Здобуття навичок у використанні, сортуванні та обробці масивів в Java.

**Завдання:**

Створити клас, який складається з методів, що виконують сортування заданого рядка та стовпця матриці із заданим типом елементів, використовуючи стандартні засоби сортування, та методів підрахунку середнього арифметичного заданого рядка та стовпця. Всі необхідні параметри передаються до методів. Всі змінні повинні бути описані та значення їх задані у виконавчому методі.

Тип елементів матриці: byte.

**Лістинг класу:**

**import** java.util.Arrays;

**public** **class** CArray{

**public** **static** **void** main(String[] args) {

**byte**[][] array = {{6, 2, 3, 0, 6}, {8, 4, 3, 7, 5}, {7, 5, 9, 5, 1}, {2, 1, 7, 8, 4}}; //даний масив

**int** row = 1, col = 5; //номери рядків і стовпців, для який виконуються операції сортування

//копіюємо вихідний масив для сортування, щоб не змінювати вхідні дані

**byte**[][] copiedArray1 = **new** **byte**[array.length][array[0].length];

**byte**[][] copiedArray2 = **new** **byte**[array.length][array[0].length];

copiedArray1 = *copyArray*(array, copiedArray1);

copiedArray2 = *copyArray*(array, copiedArray2);

System.*out*.println("Масив:"); //даний масив

*showArray*(array);

**boolean** b1 = *sortRow*(copiedArray1, row);

**if** (b1 == **true**){

System.*out*.println("Відсортований масив за " + (row + 1) + " рядком:");

*showArray*(copiedArray1);

}

**else**{

System.*out*.println("Сортування неможливе. Такого рядка не існує");

}

**boolean** b2 = *sortColumn*(copiedArray2, col);

**if** (b2 == **true**){

System.*out*.println("Відсортований масив за " + (col + 1) + " стовпчиком:");

*showArray*(copiedArray2);

}

**else**{

System.*out*.println("Сортування неможливе. Такого стовпчика не існує");

}

Double d1 = **new** Double(CArray.*averageRow*(array, row));

**if** (d1.isNaN() == **false**){

System.*out*.println("Середнє значення " + (row + 1) + " рядка: " + CArray.*averageRow*(array, row));

}

**else**{

System.*out*.println("Відшукання середнього неможливе. Такого рядка не існує");

}

Double d2 = **new** Double(CArray.*averageRow*(array, col));

**if** (d2.isNaN() == **false**){

System.*out*.println("Середнє значення " + (col + 1) + " стовпчика: " + CArray.*averageColumn*(array, col));

}

**else**{

System.*out*.println("Відшукання середнього неможливе. Такого стовпчика не існує");

}

}

//повертає середнє арифметичне вибраного рядка

**public** **static** **double** averageRow(**byte**[][] array, **int** row){

**int** s = 0;

**double** aver = 0.0;

**if** ((row >= 0) && (row < array.length)){

**for**(**int** j = 0; j < array[row].length; j++){

s += array[row][j];

}

aver = (**double**)s/array[row].length;

**return** aver;

}

**else**{

**return** Double.*NaN*;

}

}

//повертає середнє арифметичне вибраного стовпчика

**public** **static** **double** averageColumn(**byte**[][] array, **int** col){

**int** s = 0;

**double** aver = 0;

**if** ((col >= 0) && (col < array[0].length)){

**for**(**int** i = 0; i < array.length; i++){

s += array[i][col];

}

aver = (**double**)s/array.length;

**return** aver;

}

**else**{

**return** Double.*NaN*;

}

}

//метод сортування по рядку

**public** **static** **boolean** sortRow(**byte**[][] array, **int** row){

**if** ((row >= 0) && (row < array.length) ){

Arrays.*sort*(array[row]);

**return** **true**;

}

**return** **false**;

}

//метод сортування по стовпчику

**public** **static** **boolean** sortColumn(**byte**[][] array, **int** col){

**byte**[] mas = **new** **byte**[array.length];//створюємо новий одновимірний масив

**if** ((col >= 0) && (col < array[0].length) ){

**for**(**int** i = 0; i < array.length; i++){

mas[i] = array[i][col];

}

Arrays.*sort*(mas); //сортуємо одновимірний масив

//переписуємо відсортований стовпчик у вихідний масив

**for**(**int** i = 0; i < array.length; i++){

array[i][col] = mas[i];

}

**return** **true**;

}

**return** **false**;

}

//метод для друку на екран масиву

**public** **static** **void** showArray(**byte**[][] array){

**for**(**int** i = 0; i < array.length; i++){

**for**(**int** j = 0; j < array[i].length; j++){

System.*out*.printf(" " + array[i][j]);

}

System.*out*.println();

}

}

//метод копіювання масивів

**public** **static** **byte**[][] copyArray(**byte**[][] array, **byte**[][] copiedArray){

**for**(**int** i = 0; i < array.length; i++){

**for**(**int** j = 0; j < array[i].length; j++){

copiedArray[i][j] = array[i][j];

}

}

**return** copiedArray;

}

}

**Висновок:** під час виконання цієї роботи я навчився працювати з масивами. Дізнався про стандартні методи сортування масивів.