

## Лабораторна робота № 7

### Опрацювання масивів на Java. Одновимірні масиви. Пошук в одновимірних масивах.

**Мета роботи:** вивчення основних прийомів роботи з одновимірними масивами на мові Java.

#### **Короткі теоретичні відомості.**

**Масив** - це впорядкований набір однотипних елементів для звертання до яких використовують спільне ім'я. В Java можна створювати масиви будь-якого типу, вони можуть бути як одновимірні так і багатовимірні. До окремого елемента масиву звертаються за його індексом, індекс записують в квадратних дужках.

Загальна форма оголошення одновимірного масиву:

```
<тип елементів> <назва масиву>[];
```

Квадратні дужки після імені змінної власне і вказують, що змінна є масивом, їх можна записувати також і після типу:

```
<тип елементів>[] <назва масиву>;
```

Обидві форми є ідентичними.

В Java масиви є об'єктами, тому перед їх використанням слід створювати екземпляр масиву оператором `new`. При цьому вказується прогнозований розмір (кількість елементів) масиву, наприклад:

```
double[] x;  
int a[];  
x = new double[10];  
a = new int[135];
```

Зазвичай команди опису та створення масиву об'єднують:

```
float[] p = new float[20];
```

Розмір масиву в Java можна змінювати в процесі виконання програми за допомогою того ж оператора `new`. Для роботи з елементами масиву, використовують цикл `for`, на кшталт:

```
for(int i=0;i<20;i++)  
    System.out.println(p[i]);
```

Ввід елементів масиву можна здійснити за допомогою циклу, але для зручності дозволяється ініціалізувати масив елементами відразу при його описі, наприклад:

```
int p[] = {1, 3, 4, 5, 2, 6, 5, 9};
```

В такому випадку автоматично створюється екземпляр масиву, а його розмір визначається за кількістю елементів, записаних в фігурних дужках. Масиви є об'єктами – відповідно мають визначені поля і методи. Зокрема, поле `length` містить інформацію про кількість елементів масиву на даному етапі виконання програми і дозволяє опрацювати усі елементи масиву циклом

```
for(int i=0;i<p.length;i++)<дії з p[i]>;
```

**Завдання.** Скласти Java-програму розв'язання задачі відповідно до варіанта. Задано три масиви дійсних чисел  $A[10]$ ,  $B[10]$  та  $C[10]$ , кожен містить по 10 елементів. Масив  $A$  заповнити довільно в коді програми при його ініціалізації. Масив  $B$  заповнити за вказаним правилом. Масив  $C$  утворити з елементів масивів  $A$  та  $B$  згідно варіанта. Знайти в кожному з масивів вказану величину, вивести на консоль елементи кожного масиву у порядку зростання. Алгоритми пошуку, сортування та друку масивів запрограмувати у вигляді окремих статичних методів класу.

1. Масив  $B$  утворити за правилом  $B_k = \frac{(-1)^k (2k^2 + 1)}{k}$ ,  $k = 0, 1, \dots, 9$ . Масив  $C$  утворити з масиву  $B$ , замінивши в ньому всі від'ємні елементи відповідними елементами масиву  $A$  (з тими ж порядковими номерами). В кожному масиві знайти середнє арифметичне додатних елементів.
2. Масив  $B$  заповнити випадковими числами з відрізка  $[-2; 3]$  (скористатися методом `random()` класу `Math`). Масив  $C$  утворити з масиву  $B$ , замінивши в ньому всі додатні елементи максимальним елементом масиву  $A$ . В кожному масиві знайти кількість елементів, більших за їх середнє арифметичне.
3. Масив  $B$  утворити за правилом  $B_k = \frac{99 \sin k}{(k + 1)^2}$ ,  $k = 0, 1, \dots, 9$ . Масив  $C$  утворити з масиву  $B$ , замінивши в ньому всі від'ємні елементи мінімальним елементом масиву  $A$ . В кожному масиві знайти кількість елементів, абсолютна величина яких більша за 5.
4. Масив  $B$  утворити за правилом  $B_k = (-1)^k \sqrt{k!}$ ,  $k = 0, 1, \dots, 9$ . Масив  $C$  утворити за правилом  $C_k = 2A_k - 3B_k$ . В кожному масиві знайти суму додатних елементів.
5. Масив  $B$  утворити за правилом  $B_k = 11 \sin\left(\sqrt{(k + 3)^3}\right)$ ,  $k = 0, 1, \dots, 9$ . Масив  $C$  утворити за правилом  $C_k = \max\left\{\frac{A_k}{B_k}; \frac{B_k}{A_k}\right\}$ . В кожному масиві знайти суму елементів, абсолютна величина яких менша за 1.
6. Масив  $B$  заповнити випадковими числами з відрізка  $[-4; 2]$  (скористатися методом `random()` класу `Math`). Масив  $C$  утворити з масиву  $B$ , замінивши в ньому всі від'ємні елементи середнім значенням елементів масиву  $A$ . В кожному масиві знайти елемент, значення якого найближче до числа 1.
7. Масив  $B$  утворити за правилом  $B_k = k \sin(2k) + 2k \sin k$ ,  $k = 0, 1, \dots, 9$ . Масив  $C$  утворити за правилом  $C_k = \max\{A_k; B_k\}$ . В кожному масиві знайти різницю максимального та мінімального елемента (розмах значень елементів масиву).
8. Масив  $B$  утворити за правилом  $B_k = 20 \cos k - k$ ,  $k = 0, 1, \dots, 9$ . Масив  $C$  утворити з масиву  $B$ , віднявши від кожного його елемента відповідний елемент масиву  $A$ . В кожному масиві знайти суму квадратів мінімального та максимального елементів.

9. Масив В утворити за правилом  $B_k = (-1)^k \sqrt{(k+2)(k+1)}$ ,  $k = 0, 1, \dots, 9$ . Масив С утворити за правилом  $C_k = \min\{A_{9-k}; B_k\}$ . В кожному масиві знайти різницю між максимальним елементом та середнім арифметичним усіх елементів.

10. Масив В утворити заповнити випадковими числами з відрізка  $[-5; 5]$  (скористатися методом `random()` класу `Math`). Масив С утворити з елементів масиву А, додавши до кожного середнє арифметичне від'ємних елементів масиву В. В кожному масиві знайти відношення максимального елемента до мінімального.

11. Масив В утворити за правилом  $B_k = 15 \cos k - 12 \sin(10 - k)$ ,  $k = 0, 1, \dots, 9$ . Масив С утворити за правилом  $C_k = \min\{3A_k; B_k\}$ . В кожному масиві знайти відношення абсолютної величини суми від'ємних елементів до суми додатних елементів.

12. Масив В утворити за правилом  $B_k = \frac{k \sin(2k)}{(k+1)!}$ ,  $k = 0, 1, \dots, 9$ . Масив С утворити за правилом  $C_k = 2A_k + B_k$ . В кожному масиві знайти суму від'ємних елементів.

13. Масив В утворити за правилом  $B_k = 12 \cos\left(k + \sqrt{k^2}\right)$ ,  $k = 0, 1, \dots, 9$ . Масив С утворити за правилом  $C_k = \max\{2A_k; 5B_k\}$ . В кожному масиві знайти номер елемента, найближчого за величиною до числа 5.

14. Масив В утворити заповнити випадковими числами з відрізка  $[-2.5; 1.5]$  (скористатися методом `random()` класу `Math`). Масив С утворити за правилом  $C_k = \min\{A_k; B_{9-k}\}$ . В кожному масиві знайти найбільший серед від'ємних елементів.

15. Масив В утворити за правилом  $B_k = \frac{\cos(12k)}{k!}$ ,  $k = 0, 1, \dots, 9$ . С утворити з масиву В, замінивши в ньому всі додатні елементи мінімальним елементом масиву А. В кожному масиві знайти номер елемента, найближчого за величиною до середнього арифметичного усіх елементів масиву.

16. Масив В утворити за правилом  $B_k = 2 \sin k + \cos k$ ,  $k = 0, 1, \dots, 9$ . Масив С утворити додаванням оберненого масиву А до масиву В, тобто  $C_0 = A_9 + B_0$ ,  $C_1 = A_8 + B_1$ ,  $C_2 = A_7 + B_2$ , і т.д. В кожному масиві знайти кількість елементів, значення яких належать інтервалу  $(-1; 1)$ .

Хід виконання. Масиви будемо передавати методам друку сортування та пошуку в якості параметра. За рахунок поля `length` в методі завжди можна дізнатися скільки елементів містить переданий масив, наприклад, вивести на екран елементи масиву можна методом:

```
public static void printArray(double[] ar){
    String s="";
    for(int i=0; i<ar.length;i++)s+=(ar[i]+" ");
    System.out.println(s);
}
```

Для знаходження кількості елементів з інтервалу  $(-1; 1)$  створюємо метод:

```

public static int searchInArray(double[] ar){
    int k =0;
    for(int i=0; i<ar.length;i++){
        if(ar[i]>-1 & ar[i]<1)k++;
    }
    return k;
}

```

Для сортування масивів, скористаємося методом мінімального елемента, який полягає в наступному. Спочатку знаходимо найменший елемент масиву та його порядковий номер, всі елементи, які розташовані в масиві перед мінімальним зміщуємо на одну позицію далі. Тим самим звільняється перша позиція, куди записуємо мінімальний елемент. Далі розглядаємо масив без першого елемента. В ньому знову знаходимо найменший елемент і переносимо його на першу (для початкового масиву це друга) позицію, змістивши потрібні елементи далі. Потім розглядаємо масив без перших двох елементів і повторюємо процедуру переміщення мінімального елемента на початок масиву, і т. д. Орієнтовний код методу сортування:

```

public static void sortArray(double[] ar){
    for(int k=0; k<ar.length-1; k++){
        double min=ar[k];
        int iMin = k;
        for(int i=k; i<ar.length; i++){
            if(min>ar[i]){min=ar[i]; iMin=i;}
        }
        for(int i=iMin;i>k;i--) ar[i]=ar[i-1];
        ar[k]=min;
    }
}

```

В головному методі класу залишається утворити задані масиви та по черзі викликати відповідні методи:

```

public static void main(String[] args) {
    double[] a = {2.1,0.4,0.1,-0.5,0.6,7.1,-2.4,-1.3,-0.7,-9};
    double b[] = new double[10];
    double c[] = new double[10];
    for(int i=0; i<10;i++) b[i]=2*Math.sin(i)+Math.cos(i);
    for(int i=0; i<10;i++)
        c[i] = a[9-i]+b[i];
    System.out.println("Початкові масиви");
    System.out.println("Масив a: ");
    printArray(a);
    System.out.println("Масив b: ");
    printArray(b);
    System.out.println("Масив c: ");
    printArray(c);
    System.out.println("Сортовані масиви");
    sortArray(a);
    sortArray(b);
    sortArray(c);
    System.out.println("Масив a: ");
    printArray(a);
    System.out.println("Масив b: ");
    printArray(b);
    System.out.println("Масив c: ");
    printArray(c);
    System.out.println("Кількість значень з інтервалу (-1;1)");
    System.out.println("Масив a: " + searchInArray(a));
    System.out.println("Масив b: " + searchInArray(b));
    System.out.println("Масив c: " + searchInArray(c));
}

```