

Лабораторна робота №4

Використання інтерфейсів. Об'єкти.

Введення інформації з текстових файлів

Мета роботи: навчитися використовувати інтерфейси у мові Java. Ознайомитись з основами файлового введення-виведення.

1. У середовищі IntelliJ IDEA створити новий проект, що містить один головний клас Main.
2. Створити інтерфейс, що містить описання методів для введення масивів з файлів за зразком:

```
package lk2.io;

import java.io.File;

public interface DoubleArrayReader {
    double[] readOneDimensionalArray(File file);
    double[] readOneDimensionalArray(String fileName);

    double[][] readTwoDimensionalArray(File file);
    double[][] readTwoDimensionalArray(String fileName);
}
```

3. Створити інтерфейс, що містить описання методів опрацювання масивів згідно пунктів завдання.

```
package lk2.logic;

public interface ArrayProcessor {
    double calculate(double[] array);
    double calculate(double[][] array);
    void processArray(double[] array);
    void processArray(double[][] array);
}
```

4. Додати до проекту два нових класи, що реалізують описані інтерфейси, виконуючи завдання 4.1 та 4.2.
5. Створити тестові класи з методами для тестування методів класів з п.4.

Короткі теоретичні відомості

Інтерфейси - це спеціальні конструкції, які тільки оголошують набір певних дій без коду, який описує, що саме треба робити в оголошених методах (див. п.2 та п.3)

Але навіть турбуватися про створення інтерфейсів без тексту програми? Причина в тому, що інтерфейс, зроблений одного разу, може використовуватися в багатьох класах. Наприклад, коли інші класи (або сама віртуальна машина JVM) бачать, що клас `MyDoubleArrayReader` реалізує інтерфейс `DoubleArrayReader`, вони знають, що в цьому класі точно є два методи `readOneDimensionalArray(...)` з різними типами параметрів та два методи `readTwoDimensionalArray()` також з різними типами параметрів.

Це може бути зроблено за допомогою оголошення таких класів, що реалізують вказані інтерфейси:

```
public class Lab4ArrayReader
    implements DoubleArrayReader {
    .....
    .....
}
```

Кожен клас може реалізовувати довільну кількість інтерфейсів. Якщо клас реалізує деякий інтерфейс, він повинен мати реалізовані методи, відповідні до тих, що оголошені у інтерфейсі або бути оголошеним як абстрактний клас.

Стандартний клас `Scanner` може використовуватись для зчитування даних не лише зі стандартного введення, а й з файлу. Якщо об'єкт класу `Scanner` створюється конструктором `new Scanner(System.in)`, він буде виконувати зчитування зі стандартного введення. Для введення з файлу треба скористатись конструктором класу `Scanner`, якому передається об'єкт класу `File`, наприклад, так:

```
new Scanner(new File(filename));
```

Оскільки операції файлового введення-виведення можуть завершитись аварійно, мова програмування Java вимагає, аби всі вони були розміщені у «захищеному» блоці `try-catch`, як у наступному прикладі:

```
public double[] readOneDimensionalArray(String fileName) {
    try (Scanner in = new Scanner(new File(fileName))) {
        int n = in.nextInt();
        double[] arr = new double[n];
        for (int i = 0; i < arr.length; i++) {
            arr[i] = in.nextDouble();
        }
        return arr;
    } catch (IOException ex) {
        System.err.println("Error reading file");
        return null;
    }
}
```

Завдання 4.1. У створеному класі розробити метод, що зчитує дані з текстового файлу, та метод, що опрацьовує масив згідно завдання варіанту.

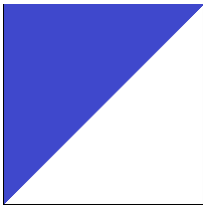
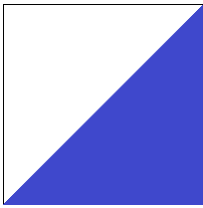
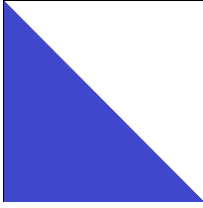
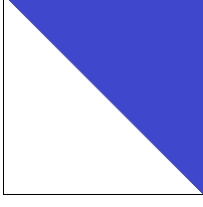
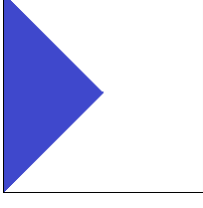
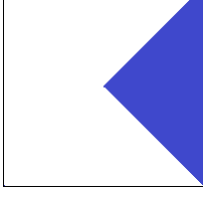
У всіх варіантах дано текстовий файл **lab41.txt**, який містить дані для розміщення у масиві, та складається з двох рядків. Перший рядок містить одне число – кількість чисел у другому рядку. Другий рядок містить дійсні числа відокремлені пропусками – елементи масиву.

(Файл підготувати самостійно. Кількість елементів у другому рядку файлу не менше 30).

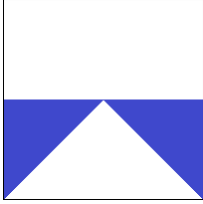
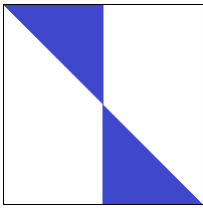
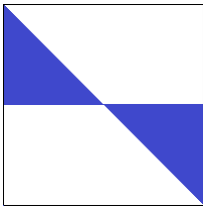
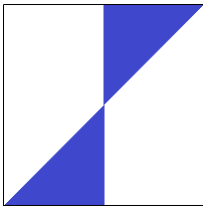
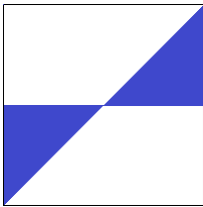
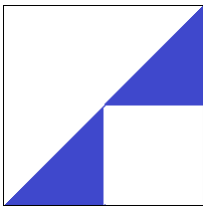
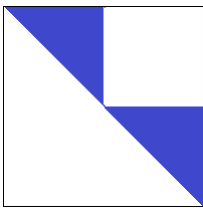
Варіант	Завдання
1	Знайти суму елементів масиву
2	Знайти добуток елементів масиву
3	Знайти суму квадратів елементів масиву
4	Знайти модуль суми та квадратний корінь добутку елементів масиву
5	Знайти останній додатний елемент масиву.
6	Знайти найменше значення серед елементів масиву
7	Знайти найменше значення серед елементів масиву з парними номерами
8	Знайти найбільший елемент масиву з непарним номером
9	Знайти суму найбільшого та найменшого елементів масиву
10	Знайти різницю першого та останнього додатних елементів масиву
11	Знайти суму елементів масиву з непарними номерами
12	Знайти суму від'ємних елементів масиву
13	Знайти модуль різниці найбільшого від'ємного елемента з непарним номером та найменшого додатного.
14	Знайти суму елементів, ціла частина яких кратна 3.
15	Знайти суму елементів масиву між першим та останнім від'ємними
16	Знайти середнє арифметичне від'ємних елементів масиву
17	Знайти суму першого від'ємного та останнього додатного елементів масиву
18	Знайти добуток всіх від'ємних елементів масиву
19	Знайти суму квадратів всіх елементів масиву
20	Знайти середнє арифметичне квадратів тих елементів масиву, які не дорівнюють нулю

Завдання 4.2. У створеному класі розробити метод, що зчитує з текстового файлу (використовувати файл **lab42.txt**) елементи двовимірного масиву цілих чисел розміром $N \times N$ елементів та метод, який опрацьовує масив згідно варіанту.

Примітка. Перший рядок файлу містить одне ціле число N . Наступні N рядків містять по N дійсних чисел – елементи заданого масиву. Файл можна отримати у викладача, або завантажити з сайту <http://www.berkut.mk.ua> у розділі «JAVA (гр.2151, 1157ст)» за [посиланням](#).

Вар	Завдання	Рисунок
1	Знайти суму елементів заштрихованої частини	
2	Знайти значення найбільшого елемента заштрихованої частини	
3	Знайти суму додатних елементів заштрихованої частини	
4	Знайти значення найменшого додатного елемента заштрихованої частини	
5	Знайти суму від'ємних елементів заштрихованої частини	
6	Знайти значення найбільшого від'ємного елемента заштрихованої частини	

Вар	Завдання	Рисунок
7	Знайти суму елементів заштрихованої частини, які за модулем менші 100	
8	Знайти значення найбільшого від'ємного елемента заштрихованої частини	
9	Знайти середнє арифметичне від'ємних елементів та заштрихованої частини	
10	Знайти суму елементів заштрихованої частини, ціла частина яких кратна 3	
11	Знайти суму найбільшого та найменшого елементів заштрихованої частини	
12	У заштрихованій частині знайти середнє арифметичне елементів, що відрізняються від найменшого елемента не більше ніж на 10%	
13	Знайти суму елементів заштрихованої частини, що відрізняються від найменшого елемента не більше ніж на 10%	

Вар	Завдання	Рисунок
14	У заштрихованій частині знайти середнє арифметичне елементів, що відрізняються від найбільшого елемента не більше ніж на 10%	
15	Знайти суму найбільшого та найменшого елементів у заштрихованій частині	
16	Знайти середнє арифметичне елементів заштрихованої частини	
17	Обчислити середнє арифметичне від'ємних елементів заштрихованої частини	
18	Обчислити середнє арифметичне додатних елементів заштрихованої частини	
19	Знайти суму від'ємних елементів заштрихованої частини	
20	Знайти суму квадратів елементів заштрихованої частини	

Примітка. Для відокремлення чисел у рядках файлу, що зчитується за допомогою `BufferedReader`, можна використовувати такий підхід:

```
public int[] readOneDimensionalArray(String fileName) {
    try (BufferedReader reader = new BufferedReader(new
        FileReader(fileName))) {
        reader.readLine();
        String[] split = reader.readLine().trim().split(" ");
        int[] arr = new int[split.length];
        for (int i = 0; i < split.length; i++) {
            arr[i] = Integer.parseInt(split[i]);
        }
        return arr;
    } catch (IOException e) {
        return new int[0];
    }
}
```