# Національний технічний університет України

*Додаток 1*

# «Київський політехнічний інститут імені Ігоря Сікорського»

**Навчально-науковий інститут атомної і теплової енергетики**

**Кафедра цифрових технологій в енергетиці**

**ЗВІТ**

# з контрольної роботи

**з дисципліни «Програмування на мові Java»**

Варіант № 25

Виконав: Студент групи ТР-12

Руденко Владислав

Дата здачі: 27.12

Київ – 2023

**Завдання згідно до варіанту**

**Завдання 1**

Написати програму на Java для обчислення:

, ,

**Завдання 2**

Розробити програму для визначення максимального та мінімального елементів {0.1, 1.6, 2.7, 4.1, 7.7} та {1.1, -2.2, 3.8, -4.1, -9.1}

**Вимоги:**

- створити метод, який використовує varargs для прийому змінної кількості дійсних чисел;

- метод повинен визначати та повертати максимальне та мінімальне значення серед переданих чисел;

- вивести результати на консоль, використовуючи різні набори дійсних чисел.

**Виконання завдання 1**

Для виконання поставленого завдання було створено програму, яка в залежності від введеного значення Х, що сканується з консолі, виконує обрахунок по формулі, згідно до завдання. Оскільки умовою передбачено що параметр а – має конкретне значення, то було вирішено зробити його незмінним. Було реалізовано функціонал умови за допомогою перевірки умови сканера.

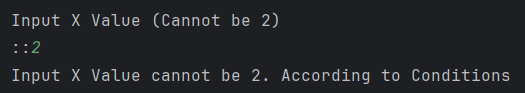
****

Рис. 1 Виконання завдання 1 з використанням хибного значення

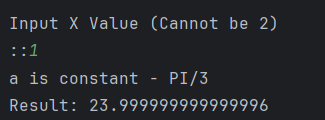
****

Рис. 2 Виконання завдання 1 з використанням вірного значння

**Висновок:**

В результаті виконання поставленого завдання була створена програма, яка здійснює обрахунок значення функції за введеним користувачем значенням X. Програма забезпечує точний обчислення виразу при дотриманні вказаних умов, використовуючи введені користувачем дані.

**Виконання завдання 2**

Для виконання завдання 2 необхідно реалізувати код, що містить функціонал пошуку мінімального та максимального значення динамічно введеного масив значень за допомогою varargs – аргументів змінної довжини. Отриманий результат виводиться до консолі та дозволяє користувачеві провести операцію знов. Для функції пошуку максимально-мінімального значення було використано окрему функцію з аргументами змінної довжини. Пошук відбувається за допомогою проходження по масиву та звірення елементів один з одним

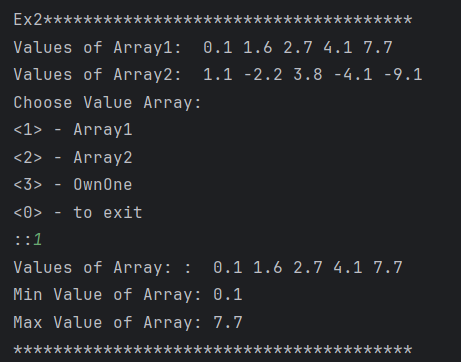
****

Рис. 3 Виконання завдання 2 з використанням 1го масиву з умови

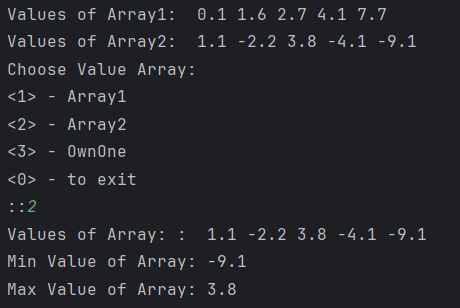


Рис. 4 Виконання завдання 2 з використанням 2го масиву з умови

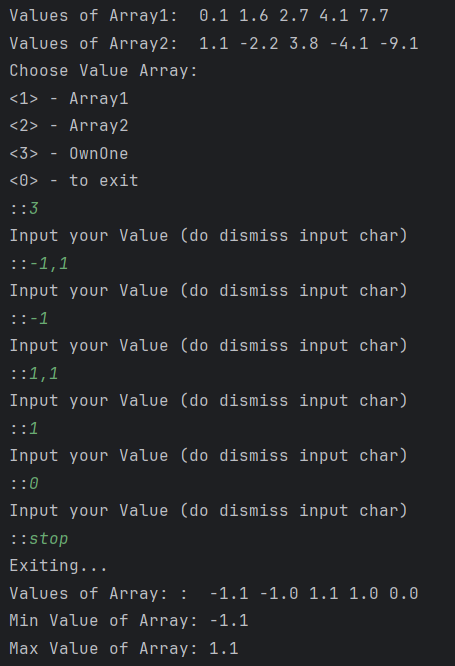


Рис. 5 Виконання завдання 2 з використанням власних значень масиву

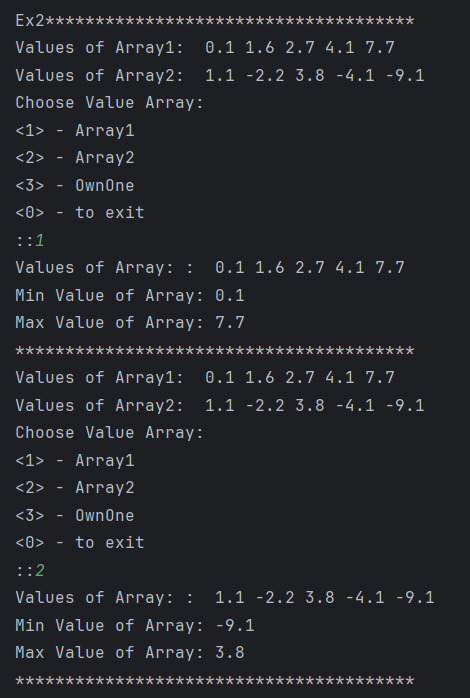


Рис. 6 Виконання завдання 2 з використанням виводу різних наборів чисел

**Висновок:**

Під час виконання завдання був реалізований код, який використовує метод, що використовує varargs для прийому змінної кількості дійсних чисел. Цей метод здійснює пошук максимального та мінімального значень серед переданих чисел. Програма була запрограмована для визначення максимальних та мінімальних значень у двох різних наборах дійсних чисел: {0.1, 1.6, 2.7, 4.1, 7.7} та {1.1, -2.2, 3.8, -4.1, -9.1}. Результати визначення максимального та мінімального значень виводяться на консоль для обох наборів чисел, дозволяючи користувачеві провести додаткові операції з цими даними.

*Додаток 1*

**Лістинг**

package MKP;

import java.util.ArrayList;

import java.util.InputMismatchException;

import java.util.List;

import java.util.Scanner;

import static java.lang.Math.PI;

import static java.lang.Math.cos;

public class MKP

{

public static void main(String[] args)

{

System.out.println("Ex1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//Ex1();

System.out.println("Ex2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Ex2();

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

public static long calculateFactorial(int n) {

if (n < 0) {

throw new IllegalArgumentException("Negative Input Value. Must be Positive");

}

long factorial = 1;

for (int i = 1; i <= n; i++) {

factorial \*= i;

}

return factorial;

}

private static void Ex1()

{

double a = PI/3;

int x = 2;

Scanner scan = new Scanner(System.in);

while(x==2)

{

System.out.printf("Input X Value (Cannot be 2)\n::");

x=scan.nextInt();

if(x==2)

{

System.out.println("Input X Value cannot be 2. According to Conditions\n");

}

}

System.out.println("a is constant - PI/3");

double l = (2\*(double)calculateFactorial(x+2))/(cos(a));

System.out.println("Result: "+l);

}

private static void Ex2()

{

Scanner scan = new Scanner(System.in);

double[] array1 = new double[] {0.1, 1.6, 2.7, 4.1, 7.7};

double[] array2 = new double[] {1.1, -2.2, 3.8, -4.1, -9.1};

double[] chosenArray = new double[] {1};

while(true)

{

Print("Array1",array1);

Print("Array2",array2);

System.out.println("Choose Value Array:");

System.out.println("<1> - Array1");

System.out.println("<2> - Array2");

System.out.println("<3> - OwnOne");

System.out.printf("<0> - to exit\n::");

switch (scan.nextInt())

{

case 1:

chosenArray = array1;

break;

case 2:

chosenArray = array2;

break;

case 3:

List<Double> tempList = new ArrayList<>();

while(true)

{

System.out.printf("Input your Value (do dismiss input char)\n::");

try

{

tempList.add(scan.nextDouble());

}

catch (InputMismatchException e)

{

System.out.println("Exiting...");

break;

}

chosenArray = new double[tempList.size()];

for (int i = 0; i < tempList.size(); i++) {

chosenArray[i] = tempList.get(i);

}

}

break;

default:

return;

}

double[] minmaxValue = Finder(chosenArray);

Print("Array: ", chosenArray);

System.out.println("Min Value of Array: " + minmaxValue[0]);

System.out.println("Max Value of Array: "+minmaxValue[1]);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

scan.nextLine();

}

}

private static void Print(String Name,double ... Value)

{

if(Value.length == 0)

{

throw new IllegalArgumentException("Finder Function Error. Need at least 1 argument");

}

System.out.printf("Values of "+Name+": ");

for(int i=0;i<Value.length;i++)

{

System.out.print(Value[i] + " ");

}

System.out.println();

}

private static double[] Finder(double ... Value)

{

if(Value.length == 0)

{

throw new IllegalArgumentException("Finder Function Error. Need at least 1 argument");

}

//result 0 - min, result 1 - max

double[] result = new double[]{Value[0],Value[0]};

for(double one : Value)

{

if(one<result[0])

{

result[0] = one;

}

else if(one > result[1])

{

result[1] = one;

}

}

return result;

}

}