**LAB 01**

**OOP Reviews and Array.**

Nguyễn Minh Đạt – ITDSIU22166

Problem 1: Simple Application

1. **Array to Number Converter:**

The public class “ArrayToNumberConverter” class contains 2 methods for convert the array to number, which are “convertArrayToNumber” and “convertHexArrayToNumber”. The first one will convert the array of digits to number, while the second one will convert the array arrays representing hexadecimal numbers.

**Validation**: Verify that every element is a valid digit (0–9). If not, handle the error appropriately (e.g., throw an exception or return an error code)

* convertArrayToNumber: There are two part in this class: first one to checking for the first one digit is negative and the second one to make sure that the other digits is from 0 – 9. If not, it will throw a error for invalid digit.
* convertHexArrayToNumber: In this class, I will check that the code have to be concluded in 0 – 9, A-F. If not, it will throw an error about invalid hexadecimal character
* Besides, when input of the array is 0 or empty, I will also throw an error of Input array cannot be null or empty

**Negative Numbers**:

* convertArrayToNumber: this class has been made to be sure that it will only accept the first one digit is negative or positive. When we have the first one digit is negative, the line digit = Math.abs(digit) will make it positive for processing

**Optional Extension**: convertHexArrayToNumber is the class supports conversion of arrays representing hexadecimal numbers (digits 0–9 and letters A–F).

**Time Complexity**: The complexity of convertArrayToNumber and convertHexArrayToNumber is O(1). Therefore, the space complexity of the main method is O(decimalSize + hexSize), or O(n) in the worst case.

**Output**:

A computer screen with white text

AI-generated content may be incorrect.

1. **Median Calculator:**

The MedianCalculator class computes the median of a list of numbers. It validates input, sorts the list, and returns the middle value (or average of two middle values for even-sized lists).

**Data Types:** By taking the List<Double> doubleNumbers = new ArrayList<>(), we can handle both int and float numbers.

**Sorting & Even-Sized Lists:** I will use Collections.sort(doubleNumbers) function for sorting the list of double numbers.

**Error Handling**: when the input is not a number, it will throw an error of "Invalid input. Please enter a number or 'done'."

**Time Complexity**: The overall time complexity of the program is dominated by the calculateMedian method, which is O(n log n), where 'n' is the number of input numbers.

**Output**:

A black background with white text

AI-generated content may be incorrect.

1. **Min-gap Finder:**

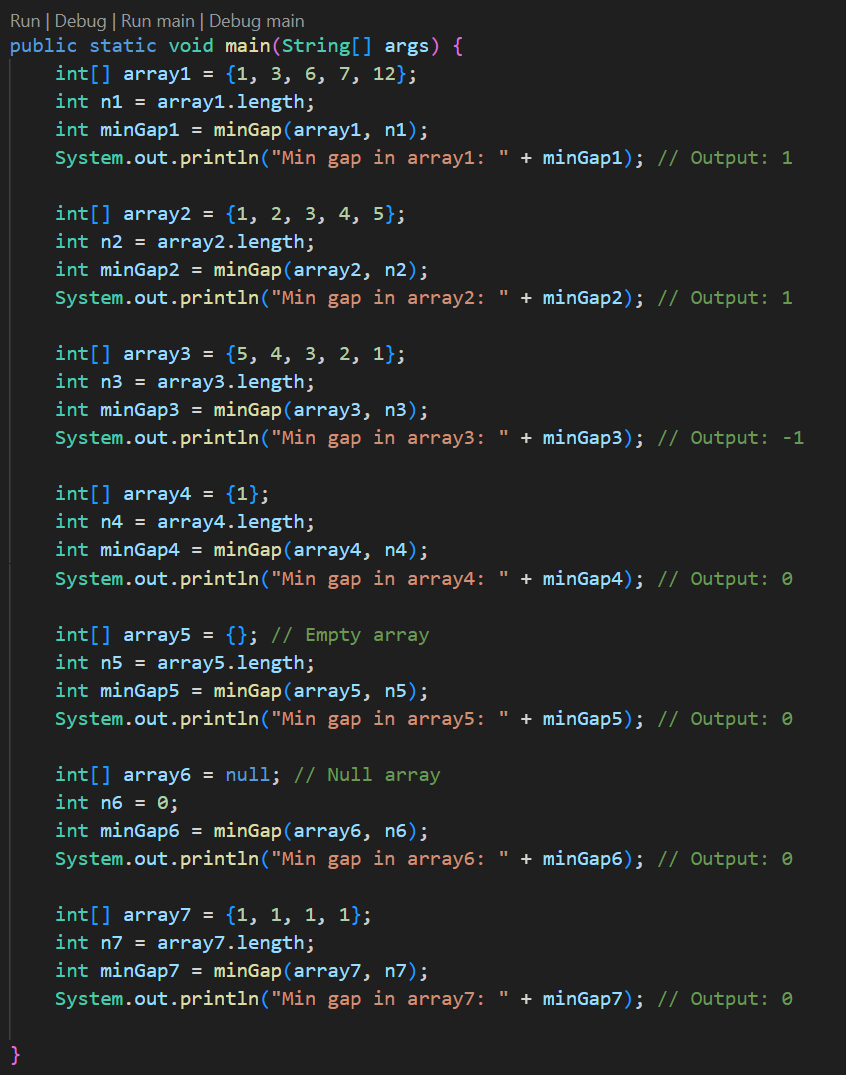
The MinGap class contains a method minGap, which calculates the minimum difference between consecutive elements in an integer array. It handles cases where the array has fewer than two elements by returning 0.

**Design Choices & Explanation:**

* **Handling Error**: When the size of the array is fewer than 0, it will throw a warning "Array size must be positive."
* **Less Elements Array**: When the array have fewer than 2 elements or null, we will return the min-gap with 0.

**Time Complexity**: The overall time complexity of the program is O(n), where n is the size of the input array.

**Output**:

A black background with white text

AI-generated content may be incorrect.

1. **GasMileage.java (input, the use of Scanner class to read numeric data):**

The GasMileage program calculates miles per gallon (MPG) for multiple cars. Users input the car name, miles driven, and fuel used. The program validates inputs, calculates MPG, and provides overall statistics, including the average MPG for all cars entered.

**Multiple Cars:** The code use a while loop for reading data for multiple cars, where each car’s information is provided on a separate line.

**Input Validation:** When the input is negative, it will throw an error of Invalid input. Beside, the miles and gallons have made sure to be double values to handling with integer or float.

**Time Complexity**: The dominant time complexity is from the input loop and the average MPG calculation loop, both of which are O(n), where n is the number of cars entered by the user. Therefore, the overall time complexity of the GasMileage program is O(n).

**Output**:  
A screen shot of a computer

AI-generated content may be incorrect.

1. **Student.java, Students.java, students.txt (text files, loops, decision making, access modifiers):**

The program reads student records from a file, processes their grades,categorizes them as "EXCELLENT," "OK," or "FAILURE," and sorts them by lastn name. It also computes statistics for each category, including the total count and average grade.

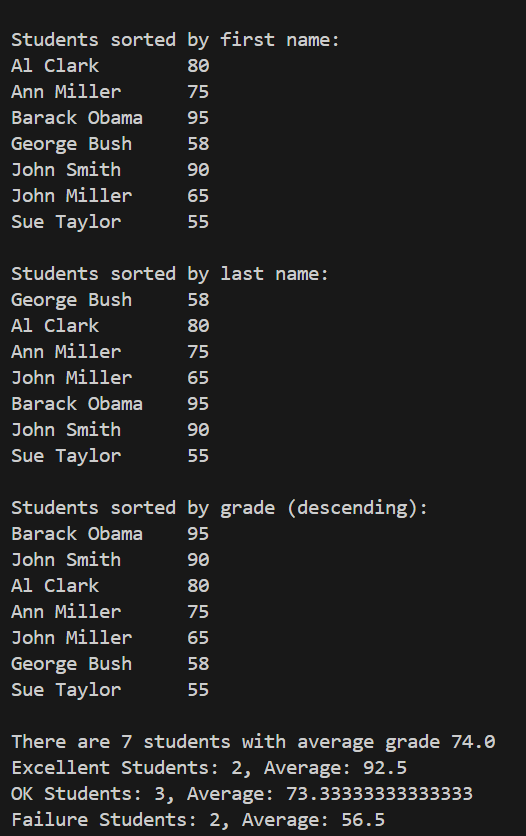
**Loop Transformation:** Pros and Cons of for vs. while:

* for Loops:
  + Pros:
    - Generally preferred when the number of iterations is known or can be determined before the loop starts.
    - The initialization, condition, and increment/decrement parts are all in one place, making the loop structure clearer.
    - Can be more concise in some cases.
  + Cons:
    - Less flexible than while loops when the number of iterations is not known in advance or when the loop termination condition is more complex.
* while Loops:
  + Pros:
    - More flexible and suitable for situations where the number of iterations is unknown or the termination condition is complex.
    - Can be used for event-driven loops or loops that continue until a specific condition is met.
  + Cons:
    - The loop structure can be less clear, especially if the initialization or increment/decrement parts are far from the loop condition.
    - Can be more prone to errors if the loop condition is not properly managed, leading to infinite loops.

**Sorting:** By usingthe function Comparator for comparing, we can sort the ouput by last name, first name or by grade

**Error Handling:** Whenthe file student.txt is not found, the code will throw an error of "Error: students.txt not found." and stop the code.

**Output**:



Problem 2: Arrays

1. **HighArrayApp.java:**

A screenshot of a computer

AI-generated content may be incorrect.

1. **OrderApp.java:**

A screenshot of a computer

AI-generated content may be incorrect.