

# HW2 report

22100233 김현중

## Time taken

과제를 시작하고 2일간 하루 약 3시간을 소비하여 총 6시간에 걸쳐 과제를 완료했습니다.

## Calculator class의 line16에 있는 args[0].toUpperCase()의 역할

args[0].toUpperCase()는 프로그램을 실행할 때 명령줄 인수로 전달되는 첫 번째 인수, 즉 Engine의 이름을 소문자로만 입력하거나 소문자를 포함해서 입력을해도 대문자로 변환해주는 역할을 해서 engine의 이름을 명령줄에 입력만 하더라도 그 이름에 맞는 engine을 실행시킬 수 있게 만듭니다.

## Difficulties

1. 처음에 InputChecker class에 있는 메소드들을 Engine들마다 어떻게 적용을 시켜야될지에 대해 많은 고민을 하였습니다. 하지만 처음에 제공된 메소드를 이해하는데 시간을 썼고, 그 결과 각 engine 클래스에 InputChecker class의 메소드들을 적용시킬 수 있었습니다.

2. MinEngine.java 파일에 java.util.Arrays가 초기에 import 되어 있었는데 왜 이것이 import 되어있는지 몰랐었습니다. 그래서 무시하고 최솟값 구하는 계산 로직을 구현했는데 java.util.Arrays를 쓰지 않아서 노란줄이 있었기 때문에, 이 부분을 구글링한 결과 최솟값을 바로 계산할 수 있는 메소드가 있는 것을 알게되었습니다.


3. 배열에 관련해서 항상 첫 번째 명령줄 인수, 즉 `args[0]`은 value 값이 아닌 engine의 이름이었기 때문에 `setInput` 메소드와 같이 input 값을 설정해주는 메소드에서의 명령줄 인수의 인덱스 값을 지정해주는 부분이 헛갈렸습니다.

### chatGPT의 도움을 받으며...

1. 인터페이스에 메소드를 선언할 때 메소드의 body를 정의해야만 되는지 알았는데 인터페이스는 메소드의 선언만을 제공하는 역할을 한다는 것을 알게 되었습니다.

H

그럼 인터페이스는 body를 정의하지 않고 메소드의 선언만을 제공하는 역할을해?



맞습니다, 인터페이스는 메소드의 선언(declaration)만을 제공하고 메소드의 구현(implementation)을 제공하지 않습니다. 인터페이스는 어떤 클래스가 특정 메소드를 가져야 한다는 계약(Contract)을 정의하는데 사용됩니다. 클래스가 인터페이스를 구현할 때, 해당 클래스는 인터페이스에 선언된 모든 메소드를 정의(구현)해야 합니다.

인터페이스의 역할은 다음과 같습니다:

1. 메소드 선언: 인터페이스는 메소드의 선언을 제공하여 어떤 메소드가 어떤 매개변수와 반환 형식을 가져야 하는지 정의합니다.


2. 계약 정의: 인터페이스는 클래스 간의 계약을 정의하는 역할을 합니다. 클래스가 특정 인터페이스를 구현한다면, 그 클래스는 인터페이스에 정의된 메소드를 반드시 제공해야 합니다.


3. 다형성 지원: 인터페이스를 구현한 여러 클래스의 객체는 동일한 인터페이스를 통해 다룰 수 있으므로, 다형성을 지원합니다. 이는 객체 지향 프로그래밍의 중요한 개념 중 하나입니다.

4. 클래스 간의 결합도 감소: 인터페이스를 사용하면 클래스 간의 결합도를 줄일 수 있습니다. 클래스는 인터페이스를 구현할 때 필요한 메소드만 구현하면 되므로, 클래스 간의 의존성이 낮아집니다.

메소드의 본문(body)을 제공하지 않는 이유는, 인터페이스가 어떤 동작을 가져야 하는지 정의하고, 이 동작을 실제로 어떻게 수행할지에 대한 구체적인 구현은 클래스에서 제공하기 때문입니다. 클래스마다 구현이 다를 수 있으며, 이것이 다형성을 지원하고, 클래스 간의 유연한 연결을 가능하게 합니다.


2. C++에서와 다르게 명령줄 인수의 첫 번째 인덱스 값은 파일이름이 아니라는 것을 알게 되었습니다.


자바에서 args값의 첫 번째 요소는 c++언어에서는 파일이름인데 자바는 첫 번째 입력값이야?

자바에서 `main` 메소드의 매개변수인 `args` 배열의 첫 번째 요소는 일반적으로 프로그램이 실행될 때 명령줄 인수(command-line arguments) 중에서 프로그램 이름이 아닌 첫 번째 입력 값입니다. 이 배열은 프로그램이 실행될 때 사용자가 명령줄에서 전달한 인수들을 저장하는 데 사용됩니다.

따라서 `args` 배열의 첫 번째 요소는 프로그램을 실행한 명령어 자체가 아니라, 프로그램을 실행할 때 사용자가 제공한 첫 번째 입력 값입니다. C++에서는 `argc`와 `argv`를 사용하여 명령줄 인수를 처리하는 데, 이 중 `argv` 배열의 첫 번째 요소가 프로그램의 이름을 가리킵니다. 자바에서는 이와 달리 `args` 배열의 첫 번째 요소가 프로그램의 이름이 아니라 첫 번째 입력 값입니다.

3. java.util.Arrays 클래스에 max 메소드는 없지만 min 메소드가 있다는 사실을 알게 되었습니다.

Arrays클래스에 max 메소드나 min 메소드가 있어?

`java.util.Arrays` 클래스에는 `max` 메소드는 없지만 `min` 메소드가 있습니다. `min` 메소드는 배열의 요소 중에서 가장 작은 값을 반환합니다.

예를 들어, 다음과 같이 `min` 메소드를 사용하여 배열에서 최솟값을 찾을 수 있습니다:

```
java
import java.util.Arrays;

public class Example {
    public static void main(String[] args) {
        double[] numbers = {5.5, 2.0, 9.8, 1.2, 7.3};

        double min = Arrays.stream(numbers).min().orElse(Double.NaN);

        System.out.println("Minimum value: " + min);
    }
}
```

위 예제에서 `min` 메소드는 `numbers` 배열에서 최솟값을 찾아서 `min` 변수에 저장합니다. 결과로는 `Minimum value: 1.2`와 같이 출력됩니다.