­

**Infinite Int Calculator Report**

**CAU 23-2**

**OOP Project02**

|  |  |
| --- | --- |
|  |  |
| Subject | Object Oriented Programming |
| Professor | Bong-Soo Sohn |
| Major | Computer Science and Engineering |
| Team | TEAM 07 |
| Members | **20204946 이규성 – Speaker**  **20203436 이병구**  **20203458 조영호** |

목차

[(b) Build and Run - 3 -](#_Toc150090427)

[Build - 3 -](#_Toc150090428)

[Run - 3 -](#_Toc150090429)

[Test environment - 3 -](#_Toc150090430)

[(c) Important Functionality - 4 -](#_Toc150090431)

[(d) how you implemented (important implementation issues) - 4 -](#_Toc150090432)

[(e) the result of SW system design - 4 -](#_Toc150090433)

[(f) execution results - 4 -](#_Toc150090434)

[(g) Appling object oriented concepts to the development for your project. also explain what you felt and learned from the project. - 4 -](#_Toc150090435)

[(h) Conclusion - 4 -](#_Toc150090436)

## Summary

Project github Link : <https://github.com/Tastypotato245/OOP_Project02>

구현된 기능: 본 라이브러리는 다음과 같은 기능을 제공한다.

* 정수, 문자열, 그리고 다른 무한 정수 객체로부터의 생성자 오버로딩
* 대입, 복사 생성자, 소멸자
* 무한 정수 간의 비교 연산(==, !=, >, <)
* 무한 정수 간의 사칙연산(+, -, \*)
* 절댓값 계산
* 무한 정수의 출력 스트림 처리

## (b) Build and Run

### Build

$ make

### Run

$ ./inf\_int 9999999999999999999 + 999999

$ ./inf\_int 12345678901234567890 \\* 123456789

$ ./inf\_int

Input: 54 \* 123456788888

Output: 6666666599952

Input: 2345234523525345 + 22233322

Output: 2345234545758667

Input: ^C (terminating)

Input: 0 (terminating)

### Test environment

* macOS 14.0 arm64 (clang 15.0.0)
* Debian GNU/Linux 11 aarch64 (g++ 10.2.1)
* if you want to execute on macOS(using Intel CPU),
* delete "-fsanitize=address -g3" of (Makefile - CFLAG option)

## (c) Important Functionality

동적 할당된 문자 배열을 이용하여 숫자를 역순으로 저장하는 방식을 통해, 숫자의 크기에 제한 없이 정수를 저장하고 처리한다.

연산 시 발생할 수 있는 자리 올림(carry)과 자리 내림(borrow)을 효율적으로 관리하기 위한 알고리즘을 구현하였다.

부호 처리를 통해 양수와 음수를 정확하게 구분하고, 연산 결과의 부호를 결정한다.

여러 숫자와 부호에 대해서도 연산할 수 있다.

## (d) how you implemented (important implementation issues)

## (e) the result of SW system design

UML

## (f) execution results

예시 사진과 결과

## (g) Appling object oriented concepts and Learning things

객체 지향 개념의 적용:

생성자와 소멸자, 연산자 오버로딩을 통해 캡슐화와 추상화를 구현하였다.

복사 생성자와 대입 연산자를 통해 객체의 올바른 복사를 보장한다.

클래스 멤버 함수를 통해 데이터에 대한 접근과 조작의 제어를 향상시킨다.

프로젝트를 통해 느낀 점 및 배운 점:

## (h) Conclusion