

# 2021 Spring OOP Assignment Report

과제 번호 : Programming Assignment #2\_problem 1

학번 : 20200437

이름 : 김채현

Povis ID : kch3481

## 명예서약 (Honor Code)

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

## 문제 1번

### 1. 프로그램 개요

본 프로그램을 간략히 설명하면 다음과 같다.

- 분수를 표현하는 Fraction class를 생성한다.
- Fraction에 대해 분자에 분모를 더하는 연산, 대입 연산, 이항 증가 연산, 덧셈 연산, 뺄셈 연산, 곱셈 연산, 나눗셈 연산을 지원한다.
- Fraction에 대해 가능한 기약분수 형태로 출력한다.

### 2. 프로그램 구조 및 알고리즘

#### 1) Fraction class member

##### (1) long long int

분자, 분모를 받는 member numerator와 denominator에 대해 int형으로 선언하려고 했지만 int형의 숫자 범위가  $-2,147,483,648 \sim 2,147,483,647$ 인 것을 고려하였을 때  $2,147,483,647 / 2 * 2 / 2,147,483,647$ 의 연산도 값이 1로 도출되어야 하지만 구현한 곱하기 연산과정 알고리즘 순서 상  $2,147,483,647 * 2 / 2 * 2,147,483,647$ 로 연산하여 Fraction class에 `int numerator = 4,294,967,294, denominator = 4,294,967,294`을 저장한 후에 약분을 진행하기 때문에 int형을 overflow한다. 이를 해결하기 위해  $2,147,483,647 / 2,147,483,646 * 2,147,483,646 / 2,147,483,647$  계산과정에서 발생하는 값  $2,147,483,647 * 2,147,483,646$ 까지 저장할 수 있는 long long int형으로 받아주었다.

#### 2) Fraction class method

##### (1) Fraction(long long int numer, long long int denom)

Fraction class에서 numer과 denom을 저장해줄 member의 data type이 long long int이기 때문에 parameter의 data type도 long long int형이다. numerator, denominator에 저장 후 약분 함수 abbreviate를 호출하여 numerator, denominator의 pointer형태로 넘겨준 후 최종적으로 약분된 형태로 numerator, denominator를 저장한다. store 함수도 비슷한 과정이다.

(2) Fraction(long long int numer, long long int denom)

Fraction class에서 numer과 denom을 저장해줄 member의 data type이 long long int이기 때문에 parameter의 data type도 long long int형이다. numerator, denominator에 저장 후 약분 함수 abbreviate를 호출하여 numerator, denominator의 pointer형태로 넘겨준 후 최종적으로 약분된 형태로 numerator, denominator를 저장한다. store 함수도 비슷한 과정이다.

(3) 연산자 오버로딩

전위 연산자의 경우 class 내의 numerator에 denominator을 더한 후 this를 return하면 되지만 후위 연산자의 경우 const fraction fr을 생성한 후에 numerator에 denominator을 더하고 fr을 return한다. 다른 연산자의 경우 별 다른 특이점 없이 구현해주었다. 메모리 복사를 아끼기 위해 가능한 reference로 받아왔다.

### 3) 토론 및 개선

본 프로그램은 class의 개념을 익히는 것을 목표로 하고 있다. class의 member를 생성하고 method를 구현하고 getter, setter 함수를 구현함으로써 class내의 private에 대해서는 외부에서 접근할 수 없다는 것을 익힐 수 있었다. 특이점은 assign 설명서에서는 member변수를 int형으로 선언하라고 하였지만 계산과정에서 overflow가 발생할 수 있기 때문에 long long int형으로 선언해주었다. 하지만 Fraction 생성자에서 numer과 denom을 임의로 받아 놓고 약분을 해준 후에 numerator과 denominator에 저장해주었다면 int형으로 받았어도 된다는 생각이 든다.