

### 10 - (2)번

**문제 정의 :** 클래스 선언부와 클래스 구현부를 헤더파일과 cpp 파일로 나누어 프로그램을 정확히 작성한다.

#### 문제 해결 방법 및 알고리즘 설명:

나는 코드를 클래스 선언부(헤더파일), 클래스 구현부(Class\_Implementation.cpp), 메인 함수(Main.cpp)로 나누었다.

#### 클래스 선언부 헤더파일 ( Add.h 파일)

```
#ifndef ADD_H
#define ADD_H

class Add {
    int a;
    int b;
public:
    void setValue(int x, int y);
    int calculate();
};

# endif
```

클래스 4개의 구현부를 각각의 헤더파일로 나누었다.

public으로 멤버에 대해 외부에서 접근이 가능하게 한다.

class 안에서는 멤버함수를 선언만 해준다. (문제가 선언부와 구현부 분리였기 때문이다.)

조건 컴파일 문으로 다른 조건 컴파일 상수와 충돌을 피하기 위해 클래스의 이름으로 한다.

같은 방식으로 다른 Sub, Mul, Div 클래스의 선언부 헤더파일을 만들어준다.

이러면 헤더파일이 총 4개가 된다.

#### 클래스 구현부 cpp 파일 (Class\_Implementation.cpp)

```
#include "Add.h"
#include "Sub.h"
#include "Mul.h"
#include "Div.h"

void Add::setValue(int x, int y)
{
    a = x;
    b = y;
```

```

}
int Add::calculate()
{
    return a + b;
}
void Sub::setValue(int x, int y)
{
    a = x;
    b = y;
}
int Sub::calculate()
{
    return a - b;
}
void Mul::setValue(int x, int y)
{
    a = x;
    b = y;
}
int Mul::calculate()
{
    return a * b;
}
void Div::setValue(int x, int y)
{
    a = x;
    b = y;
}
int Div::calculate()
{
    return a / b;
}

```

가장 위에 방금 만든 클래스 선언부 헤더파일 4개를 #include 해준다.

함수의 리턴타입(void, int), 클래스 이름(Add), 범위지정연산자(::), 멤버 함수명과 매개변수를 써준다. 이때, calculate은 인자를 받아오지 않기 때문에 매개변수 자리를 비워둔다.

이때, Add의 Calculate 함수는 a+b를, Sub의 Calculate 함수는 a-b를, Mul의 Calculate 함수는 a\*b를, Div의 Calculate 함수는 a/b를 반환해준다.

메인함수 cpp (Main.cpp)

```
#include "Add.h"
#include "Sub.h"
#include "Mul.h"
#include "Div.h"
```

메인함수의 코드는 10-(1)번의 메인함수 코드와 동일하다.

동일하게 객체를 생성하고, 연산자와 피연산자 변수를 생성하고, 무한루프 안에서 피연산자와 연산자를 받아와서 원하는 연산을 하는 걸 반복한다.

다만, 10-(2)번의 Main.cpp에서는 메인함수 위에 우리가 만든 클래스 선언부 4개의 헤더파일을 #include 해주어야 한다.

## 아이디어 평가

이 방식으로 코드를 만들어서 실행하면 이와 같이 올바른 결과를 얻을 수 있다.

