

## <객체지향프로그래밍\_4주차과제\_소스구현설명>

7조(202104140김경록, 202104276이선우)

### ----문제 정의----

프로그램을 작성할 때 가장 먼저 필요한 부분은 조건에 따라 하는 것이라고 생각했기 때문에 프로그램의 기틀을 어떻게 해야할 지에 대해 먼저 고민했습니다.

그래서 사칙계산기라는 큰 틀에서 +,-,/,\*의 클래스를 각각 만들고 그 클래스의 부품들인 setValue 함수, calculate 함수, a,b변수들을 어떻게 다룰지에 대해 고민했습니다. 그때 생각났던 것은 지정자를 이용해 각 클래스 별로 함수들을 정의하는 것이었습니다. 그 후 while 무한루프 구문을 통해 if를 사용하여 알맞은 calculate 함수를 불러오는 식으로 코드를 작성했습니다. 코드에 대해 설명하겠습니다.

### ----문제 해결 방법, 아이디어들(코드 옆에 설명)----

#include <iostream> 입출력을 위한 라이브러리를 불러왔습니다.

using namespace std; 표준 네임스페이스 사용을 선언하여 std:: 생략이 가능하도록 만들었습니다.

다음은 덧셈을 위한 클래스 정의입니다.

```
class Add {
```

private를 int형 변수 두개에 적용함으로써 변수에 대한 보안을 더 철저하게 하는 모습입니다. 이들을 public에 넣었을 때도 코드를 실행했을 때 같은 결과값이 나왔습니다.

```
private:
```

```
    int a, b;
```

ADD 클래스에 있는 setValue 함수를 구현했습니다. 이 함수의 의미는 멤버 변수 a,b에 입력 받은 x값과 y값을 설정하는 것입니다. 이때 입력은 추후 main 함수에서 나타내겠지만 입력

은 main 함수에서 받습니다. calculate 함수는 ADD 클래스에서는 더하기를 적용합니다.

public:

```
void setValue(int x, int y){
```

```
    a = x;
```

```
    b = y;
```

```
}
```

```
int calculate();
```

```
};
```

```
void Add::setValue(int x, int y) {
```

```
    a = x;
```

```
    b = y;
```

```
}
```

```
int Add::calculate() {
```

```
    return a + b;
```

```
}
```

뺄셈을 위한 클래스 정의입니다.

```
class Sub {
```

ADD 클래스에서와 마찬가지로 int형 변수인 a,b는 값의 보존을 위해 private에 포함시킵니다.

private:

```
    int a, b;
```

Sub 클래스에 있는 setValue 함수를 구현했습니다. 이 함수의 의미는 멤버 변수 a,b에 입력 받은 x값과 y값을 설정하는 것입니다. 마찬가지로 입력은 main 함수에서 받습니다. calculate

함수는 Sub 클래스에서는 빼기를 적용합니다.

public:

```
void setValue(int x, int y) {
```

```
    a = x;
```

```
    b = y;
```

```
}
```

```
int calculate();
```

```
};
```

```
void Sub::setValue(int x, int y) {
```

```
    a = x;
```

```
    b = y;
```

```
}
```

```
int Sub::calculate() {
```

```
    return a - b;
```

```
}
```

ADD 클래스에서와 마찬가지로 int형 변수인 a,b는 값의 보존을 위해 private에 포함시킵니다.

```
class Mul {
```

```
private:
```

```
    int a, b;
```

Mul 클래스에 있는 setValue 함수를 구현했습니다. 이 함수의 의미는 멤버 변수 a,b에 입력 받은 x값과 y값을 설정하는 것입니다. 마찬가지로 입력은 main 함수에서 받습니다. calculate

함수는 Mul 클래스에서는 곱하기를 적용합니다.

public:

```
void setValue(int x, int y) {
```

```
    a = x;
```

```
    b = y;
```

```
}
```

```
int calculate();
```

```
};
```

```
void Mul::setValue(int x, int y) {
```

```
    a = x;
```

```
    b = y;
```

```
}
```

```
int Mul::calculate() {
```

```
    return a * b;
```

```
}
```

```
class Div {
```

ADD 클래스에서와 마찬가지로 int형 변수인 a,b는 값의 보존을 위해 private에 포함시킵니다.

private:

```
    int a, b;
```

Mul 클래스에 있는 setValue 함수를 구현했습니다. 이 함수의 의미는 멤버 변수 a,b에 입력 받은 x값과 y값을 설정하는 것입니다. 마찬가지로 입력은 main 함수에서 받습니다. calculate 함수는 Mul 클래스에서는 곱하기를 적용합니다.

public:

```
void setValue(int x, int y) {  
    a = x;  
    b = y;  
}  
int calculate();  
};
```

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

```
int Div::calculate() {  
    return a / b;  
}
```

문제에서 주어진 것처럼 각 클래스별 객체들을 생성하고 입력을 받을 수 있는 정수형 변수 x,y 그리고 부호를 판단해줄 문자형 변수 c를 선언하였습니다.

```
int main() {  
    Add a;  
    Sub s;
```

```
Mul m;
```

```
Div d;
```

```
int x, y;
```

```
char c;
```

While(true)를 통해 무한루프 조건을 만족시켰으며 정수 부호의 순으로 입력 받을 수 있도록 코드를 작성했습니다. +는 ADD, -는 Sub, \*는 Mul /는 Div 클래스를 적용합니다. 그 후 각 클래스에 정의되어 있는 calculate 함수를 불러와 계산한 값을 불러온 후 그 값을 출력하도록 코드를 짰습니다.

```
while (true) {  
    cin >> x >> y >> c;  
  
    if (c == '+') {  
        a.setValue(x, y);  
        cout << a.calculate() << endl;  
    }  
  
    else if (c == '-') {  
        s.setValue(x, y);  
        cout << s.calculate() << endl;  
    }  
  
    else if (c == '*') {  
        m.setValue(x, y);  
        cout << m.calculate() << endl;  
    }  
  
    else if (c == '/') {  
        d.setValue(x, y);
```

```

        cout << d.calculate() << endl;

    }

}

}

```

코드의 설명은 위와 같습니다.

----아이디어 평가----

이 프로그램의 핵심이자 아이디어는 Class에 포함된 멤버들을 어떻게 활용할 것인지에 대해 초점을 두는 것과 if문을 사용하는 것이었습니다. 먼저 의미가 없어 보이더라도 setValue 함수를 정의해두었는데 이를 정의하지 않고 하나의 코드로 통일하더라도 같은 결과값이 나오기도 했습니다. 하지만 과제를 진행하는 과정에서 이는 보안상의 문제로 지양해야 하는 점이라는 것을 알게되어 각 클래스별로 함수를 정의하게 되었습니다. 또한 if문을 통해 +,-,\*,/를 나누어 활용하는 것이 중요했습니다.

----문제를 해결한 키 아이디어 또는 알고리즘 설명----

이 프로그램은 정수 2개, 문자를 통해 계산을 진행하게 하는 간단한 프로그램입니다. 하지만 C++ 객체지향 프로그래밍의 기초인 ::을 활용하지 못하면 풀지 못하는 문제였습니다. 그리고 또한 if문이라는 키 아이디어를 사용하면 잘 풀리는 문제였습니다. 결국 교제를 통해 조원과 고민해본 결과 가장 간단하게 작성되는 코드는 이것이었습니다.