소스 구현 설명

202304178 구서연

10 - (1)번

문제 정의 : 클래스 선언부와 클래스 구현부를 분리하는 법을 정확히 알고 한 파일에 구현하는 것.

문제 해결 방법 및 알고리즘 설명:

클래스 선언부

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

public으로 멤버에 대해 외부에서 접근 가능하게 함.
class 안에서는 멤버함수를 선언만 해줌. (문제가 선언부와 구현부 분리였기 때문)

클래스 구현부

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

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

또, Add 클래스의 calculate 함수는 int값을 리턴해줘야 한다.

같은 방식으로 클래스 Sub, Mul, Div도 만든다.
이때, Sub의 Calculate 함수는 a-b를, Mul의 Calculate 함수는 a*b를, Div의 Calculate 함수는 a/b를 반환해준다.

그 밑에 int main(void) 메인함수를 써준다.

```
Add a;
Sub s;
Mul m;
Div d;
```
로 각각의 클래스 타입의 객체를 생성해준다.

```
int x, y;
char op;
```
키보드로부터 받을 피연산자와 연산자를 넣을 변수를 만들어주고,
문제에서 프로그램은 무한루프를 돈다고 했기 때문에, 우리가 원하는 액션들을 무한루프 안에서 해준다.

```
while (1)
{
	cout << "두 정수와 연산자를 입력하세요 >> ";
	cin >> x >> y >> op;
	switch (op)
	{
		case('+'):
		{
			a.setValue(x, y);
			cout << a.calculate() << endl;
			break;
		}
		case('-'):
		{
			s.setValue(x, y);
			cout << s.calculate() << endl;
			break;
		}
		case('*'):
		{
			m.setValue(x, y);
			cout << m.calculate() << endl;
			break;
		}
		case('/'):
		{
			d.setValue(x, y);
			cout << d.calculate() << endl;
			break;
		}
	}
```

}

무한루프 안에서 연산자와 피연산자를 입력받고 연산자에 해당하는 객체의 setValue함수를 호출하고, 그 객체의 calculate()의 결과를 출력해준다. 그리고 무한루프로 이를 반복한다.

아이디어 평가

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