과제#1 내용 보고서

과제 내용

교재 P154 - P155의 10번 문제를 프로그래밍하여 제출하시오.

1. 문제 정의

본 문제에서는 사칙연산(+, -, *, /)을 수행하는 네 개의 클래스를 생성하여, 두 개의 정수와 연산자를 입력받고 해당 연산을 수행하여 결과를 출력하는 프로그램을 작성합니다. 이를 위해 다음과 같은 요구사항을 설정했습니다:

- 각 클래스는 두 개의 정수를 멤버 변수로 가집니다 (`int a, b`).

- 각 클래스는 `void setValue(int x, int y)` 메서드를 통해 두 정수를 초기화하고, `int calculate()` 메서드를 통해 연산 결과를 반환합니다.

- 사용자의 입력을 받아 연산을 처리하고, 결과를 출력합니다.

- 프로그램은 무한 루프를 통해 계속 실행됩니다.

- 클래스의 선언부와 구현부는 헤더 파일과 CPP 파일로 분리하여 작성합니다.

2. 문제 해결 방법

1. 클래스 정의:

   - 각 연산을 담당하는 클래스를 `Add`, `Sub`, `Mul`, `Div`로 정의했습니다.

   - 각 클래스는 `setValue`와 `calculate` 메서드를 가지고 있으며, 이들 메서드는 각각의 클래스에 맞는 연산을 수행합니다.

구현>

```
class Add{ //Add라는 클래스 정의
        int a, b;
public:
        void setValue(int x, int y);

        int calculate();
};
```

+ 해당 클래스의 객체 생성 a,s,m,d:
Add a;
Sub s;
Mul m;
Div d;

2. 입력 처리:

- `cin`을 사용하여 사용자로부터 두 개의 정수와 연산자를 입력받습니다.

- 입력된 값을 변수 `x`, `y`, `op`에 저장합니다.

구현>
```
cout << "두 정수와 연산자를 입력하세요>>";
int x, y;
char op;
cin >> x >> y >> op;
```

3. 연산 수행:

- 연산자 구분: 사용자가 입력한 연산자를 if문으로 검사하여, 각각의 연산에 맞는 클래스의 함수를 불러옵니다.

- 입력된 연산자에 따라 적절한 클래스의 `setValue` 메서드를 호출하여 두 정수를 저장합니다.

- 이후 `calculate` 메서드를 호출하여 연산 결과를 얻고 값을 반환합니다.

구현>

연산자 구분하고 해당 setValue와 calculate 실행
```
if (op == '+') {    +-*/ 연산자 구분
        a.setValue(x, y);
        a.calculate();
        cout << a.calculate() << endl;
}
```

4. 무한 루프 구현:

- `while (true)` 문을 사용하여 프로그램이 계속해서 사용자 입력을 받을 수 있도록 설정했습니다.

5. 파일 분리:

   - `Calculator.h` 파일에서 클래스 선언을 하고, `Calculator.cpp` 파일에서 각 클래스의 메서드를 구현하였습니다.

구현> 보고서 페이지 4~6p(전체 코드)

3. 아이디어 평가

- 구조적 설계: 각 연산을 별도의 클래스로 구현함으로써 코드의 가독성과 유지보수성을 높였습니다.

- 유연성: 새로운 연산을 추가하거나 기존의 연산을 수정하는 것이 용이합니다. 필요한 경우, 새로운 클래스만 추가하면 됩니다.

- 에러 처리: 나눗셈 클래스에서는 0으로 나누는 경우를 처리하여 사용자에게 적절한 오류 메시지를 제공합니다.

- 반복적 실행: 무한 루프를 통해 사용자가 원하는 만큼 연산을 수행할 수 있게 하였습니다.

4. 키 아이디어 및 알고리즘 설명

본 프로그램의 핵심 아이디어는 <객체 지향 프로그래밍>을 활용하여 각 연산을 클래스 형태로 구현한 것입니다. 이를 통해 각 클래스는 독립적으로 자신의 연산을 수행하며, 코드의 중복을 줄이고 재사용성을 높였습니다.

- setValue 메서드는 입력된 두 정수를 멤버 변수에 저장하고,

- calculate 메서드는 저장된 두 정수에 대한 연산을 수행하여 결과를 반환합니다.

이와 같은 구조는 프로그램의 명확한 목적을 달성하며, 사용자의 입력에 따라 다양한 결과를 동적으로 제공할 수 있습니다.

이상으로 과제#1에 대한 보고서를 마치겠습니다.

Calculator.h 폴더

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

Calculator.cpp 폴더

```
#include <iostream>
#include "Calculator.h"

using namespace std;
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;
}
int main() {
	Add a;
	Sub s;
	Mul m;
```

```
	Div d;


	while (true) {
		cout << "두 정수와 연산자를 입력하세요>>";
		int x, y;
		char op;
		cin >> x >> y >> op;
		if (op == '+') {
			a.setValue(x, y);
			a.calculate();
			cout << a.calculate() << endl;
		}
		else if (op == '-') {
			s.setValue(x, y);
			s.calculate();
			cout << s.calculate() << endl;

		}
		else if (op == '*') {
			m.setValue(x, y);
			m.calculate();
			cout << m.calculate() << endl;
		}
		else if (op == '/') {
			d.setValue(x, y);
			d.calculate();
			cout << d.calculate() << endl;

		}
	}
}
```