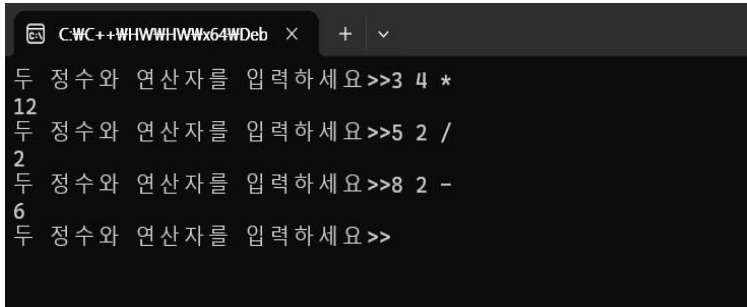


객체지향프로그래밍 과제#1

소프트웨어전공 202284012 김주원 / 소프트웨어전공 202304012 김도연

1. 소스 수행 결과 화면



```
C:\WC++\WHWWHWWx64WDeb x + v
두 정수와 연산자를 입력하세요>>3 4 *
12
두 정수와 연산자를 입력하세요>>5 2 /
2
두 정수와 연산자를 입력하세요>>8 2 -
6
두 정수와 연산자를 입력하세요>>
```

2. 문제 정의

이 문제는 덧셈(Add), 뺄셈(Sub), 곱셈(Mul), 나눗셈(Div)을 처리하는 4개의 클래스를 만들고 각 클래스에서 두 개의 정수와 연산자(+, -, *, /)를 입력받아 해당 연산을 처리하는 프로그램이다. 각 클래스는 int형 변수 a와 b를 가지고 setValue() 함수로 두 정수를 입력 받아 calculate() 함수로 연산 결과를 반환한다.

3. 문제 해결 방법

<선언부>

1. 기본 로직

클래스의 정의를 포함하여 각 클래스의 멤버 변수와 메서드의 구조를 제공하고 클래스와 관련된 모든 정의를 한 곳에 모아둠으로써 코드의 구조를 명확히 하고 관리하기 쉽게 만든다. 클래스 Calculator를 정의하여 정수 a, b와 값을 설정하는 setValue 메서드를 선언하고 이를 상속받아 각각의 연산을 수행하는 Add, Sub, Mul, Div 클래스를 구현한 후 각 클래스에서 calculate 메서드를 정의한다.

2. 헤더 파일

- 헤더 파일이 중복으로 포함되는 것을 방지하여 컴파일 오류를 예방하기 위해 #ifndef CALCULATOR_H와 #define CALCULATOR_H로 중복 포함 방지용 전처리 기 지시문 설정

3. Calculator 클래스 정의

사칙연산의 기본 클래스를 정의하여 피연산자를 저장하고 기본 기능을 제공한다.

- 자식 클래스에서 이 변수를 직접 접근할 수 있도록 하기 위해 protected 접근 권한 지정으로 멤버 변수 a, b를 저장
- 외부에서 Calculator 객체에 대해 값을 설정할 수 있도록 setValue(int x, int y) 함수를 public으로 선언
- 자식 클래스에서 구현해야 하는 추상 함수 calculate()를 선언
- 프로그램에서 객체의 실제 타입에 따라 적절한 calculate() 메서드가 호출되도록 하고 다형성을 구현하기 위해 virtual 사용

4. 멤버 함수

자식 클래스에서 피연산자를 설정하고 계산을 수행하는 기능을 제공한다.

- setValue 함수로 자식 클래스에서 피연산자를 설정하고 사용자로부터 입력받은 값을 멤버 변수에 저장
- Calculator 클래스에 정의된 순수 가상 함수 calculate는 자식 클래스에서 반드시 구현하고 각 클래스는 해당 연산에 대한 결과를 반환

5. 사칙연산 클래스 정의

각각의 사칙연산을 수행하는 클래스를 정의하여 코드의 재사용성을 높인다.

- 덧셈, 뺄셈, 곱셈, 나눗셈을 수행하는 Add, Sub, Mul, Div 클래스를 선언하고 각 클래스는 calculate() 메서드를 오버라이드하여 연산을 구현

6. 헤더 파일 보호

헤더 파일이 여러 번 포함되는 것을 방지하고 코드의 안정성과 가독성을 높이기 위해 #endif 사용

<구현부>

1. 기본 로직

사용자의 두 개의 정수와 연산자를 입력을 받아 연산자에 따라 적절한 객체를 선택한 후 setValue() 함수로 값을 설정하고 calculate() 함수로 결과를 출력한다.

2. 함수 구현

C++에서 입출력 스트림 기능을 제공하는 헤더 파일을 포함하기 위해 iostream 사용
Std 네임스페이스를 전역에서 사용하기 위해 using namespace std; 사용

사용자로부터 두 개의 정수를 입력받는 함수로 setValue(int x, int y) 설정

- 입력된 두 정수 x, y를 멤버 변수 a, b에 저장

구체적인 연산 클래스인 Add, Sub, Mul, Div를 Calculator 클래스를 상속받아 각 연산에 맞는 calculate() 메서드를 구현

- Add 클래스에서 a와 b의 합을 반환
- Sub 클래스에서 a에서 b를 뺀 결과를 반환
- Mul 클래스에서 a와 b의 곱을 반환
- Div 클래스에서 분모(b)가 0이 아닌 경우에만 a, b를 나눈 결과를 반환하며 분모(b)가 0인 경우 에러 메시지를 출력하고 0을 반환

main() 함수

- Add, Sub, Mul, Div 클래스 타입의 객체 a, s, m, d를 생성하고 덧셈, 뺄셈, 곱셈, 나눗셈을 이 객체로 수행
- while()문으로 프로그램이 사용자로부터 계속 입력받을 수 있도록 루프 사용
- while문에 정수 x, y와 연산자(문자형) op를 선언
- cin >> x >> y >> op;로 입력된 값을 변수 x, y, op에 저장

- **switch**문을 사용하여 입력된 연산자에 따라 수행할 연산 결정
 - 각 연산자 case에 대해 해당 클래스의 setValue 메소드를 호출하여 x, y를 설정
 - calculate 메소드를 호출하여 각 연산자에 적용되는 계산 방식으로 결과 출력
 - 잘못된 연산자가 입력된 경우 default 케이스가 실행되어 경고 메시지 출력
 - 사용자가 입력한 데이터를 읽어들이고 변수에 저장하는 cin과 출력 내용을 보여주는 cout를 사용하여 사용자와의 상호작용을 쉽게 처리
-
- main 함수 마지막에 return 0;을 사용하여 프로그램이 정상적으로 종료됨을 표시