

객체지향프로그래밍 4주차 과제 소스구현 설명

ICT융합학부 소프트웨어학과
202284002 고범주

1.문제 정의:

이 프로그램은 사용자로부터 두 개의 정수와 하나의 연산자를 입력받아, 주어진 정수들에 대해 해당 연산을 수행한 결과를 출력하는 기능을 구현하는 것입니다. 이 과정에서 더하기(+), 빼기(-), 곱하기(*), 나누기(/) 사칙연산을 수행하는 4개의 클래스(Add, Sub, Mul, Div)를 만들어야 하며 각 클래스마다 int타입의 변수인 피연산자, 매개 변수 x,y를 a,b에 복사하는 void setValue() 함수, 연산을 실행하고 결과값을 내보내는 int calculate() 함수가 공통적으로 있어야 하며 main함수에는 각 클래스 타입의 객체를 생성하여 그 객체에 대한 setValue() 함수를 호출하고 calculate()함수를 호출하여 결과를 출력해야 하며 프로그램은 무한루프를 돌아야 합니다.

2.문제 해결 방법:

(1)클래스 만들기

각 연산을 처리하기 위해 클래스(Add, Sub, Mul, Div)를 만들고 사용자로부터 입력받은 x,y 두 정수에 대해 연산을 수행하도록 합니다.

각 클래스는 두 정수(a, b)를 멤버 변수로 가지며, setValue() 함수로 이 값들을 설정합니다.

calculate() 함수는 각 클래스에서 특정 연산(+, -, *, /)을 수행하여 결과를 반환합니다.

(2)연산자(+,-,*,/)에 따른 계산

사용자가 연산자를 입력하면 if, else if 문을 사용하여 해당 연산에 맞는 클래스를 호출합니다.

덧셈은 Add 클래스, 뺄셈은 Sub 클래스, 곱셈은 Mul 클래스, 나눗셈은 Div 클래스를 사용하여 연산을 수행합니다.

(3)나눗셈 예외처리

나눗셈(Div)에서 나누는 값(b)이 0일 경우, 0으로 나눌 수 없다는 메시지를 출력하고 연산을 중단하도록 예외 처리를 추가합니다.

(4)무한 루프와 반복 입력

프로그램은 무한 루프(while(true))로 계속해서 사용자 입력을 받고, 연산을 반복 수행합니다.

3.아이디어 평가

(1)클래스 분리

각 연산(+, -, *, /)을 개별 클래스로 분리하여, 각 연산은 독립적인 클래스로 분리되었기 때문에, 새로운 연산을 추가하거나 기존 연산을 수정하기 쉽습니다.

(2)입력 처리

사용자 입력을 받고 연산자를 조건문으로 처리하는 방식으로 사용자가 원하는 계산이 가능합니다.

(3)예외 처리

나눗셈에서 0으로 나누는 상황을 if문을 사용하여, 오류를 방지하였습니다.

4.문제를 해결한 키 아이디어 또는 알고리즘 설명

(1)독립적인 클래스들의 연산처리

각 연산을 독립적인 클래스가 계산하도록 구현하였습니다. 각 클래스는 모두 비슷한 구조로 설계되었으며, 두 정수를 받아 각기 다른 연산을 수행하는 setValue()와 calculate() 함수를 사용합니다.

(2)조건문을 이용한 연산처리

사용자가 입력한 연산자를 기준으로 해당 클래스의 메서드를 호출하여 연산을 수행합니다. 조건문(if, else if)을 사용하여 연산자를 확인하고, 그에 맞는 클래스를 호출합니다.

(3)0으로 나누기 예외 처리

나눗셈의 경우, 0으로 나누는 상황에서 예외 처리를 도입하여 프로그램의 안정성을 높였습니다. Div 클래스에서 calculate() 메서드를 호출할 때, b 값이 0이면 연산을 수행하지 않고 경고 메시지를 출력합니다.