

## ① 수행 과업

이 과제는 파이썬의 기본 문법(변수, 연산자, 조건문, 반복문)을 실습하며, 사용자가 입력한 데이터를 기반으로 계산 작업을 수행하는 프로그램을 작성하는 데 있습니다. 계산기의 기본 동작을 통해 프로그래밍 문제 해결 역량과 예외 처리를 학습합니다.

## ② 요구사항

## 1. 기능 정의

- 사용자로부터 두 숫자와 연산자(+, -, \*, /)를 입력받는다.
- 입력받은 연산자에 따라 덧셈, 뺄셈, 곱셈, 나눗셈 결과를 출력한다.
- 나눗셈 연산의 경우, 0으로 나누는 예외를 처리한다.

## 2. 확장 가능성

- 프로그램이 종료되지 않고 반복적으로 동작하도록 구현한다.
- 사용자가 "exit"을 입력하면 프로그램이 종료된다.

## 3. 실무 활용

- 입력된 데이터가 유효한 숫자인지 확인하여, 잘못된 입력에 대해 사용자에게 안내 메시지를 출력한다.

## ③ 스스로 학습

## 1. 파이썬의 데이터 타입

- 키워드: int, float, str, bool
- 학습 방향: 숫자, 문자열, 논리값의 차이와 활용 사례를 알아보고, 변수 선언 및 기본 연산을 연습합니다.

## 2. 입출력

- 키워드: input(), print()
- 학습 방향: 사용자 입력을 받아 처리하는 방법과, 데이터를 출력하는 방법을 이해합니다.

## 3. 조건문

- 키워드: if, elif, else
- 학습 방향: 조건에 따라 다른 결과를 실행하는 방법을 이해합니다.

## 4. 반복문

- 키워드: while, for
- 학습 방향: 반복 작업을 수행하는 두 가지 방식의 차이를 이해하고, 적합한 상황을 찾아봅니다.

## 5. 예외 처리

- 키워드: try, except, raise
- 학습 방향: 프로그램 실행 중 발생할 수 있는 오류를 다루고, 안정적인 코드를 작성하는 방법을 배웁니다.

④ 코드 구현 예제를 이해하여 설명하시오.

```
def calculator():
    print("=== 간단한 계산기 프로그램 ===")
    print("사용법: 숫자1 연산자 숫자2 (예: 5 + 3)")
    print("프로그램 종료를 원하면 'exit'를 입력하세요.\n")

    while True:
        user_input = input("계산식을 입력하세요: ").strip()

        # 종료 조건
        if user_input.lower() == "exit":
            print("프로그램을 종료합니다.")
            break

        # 입력 파싱
        try:
            num1, operator, num2 = user_input.split()
            num1 = float(num1)
            num2 = float(num2)

            # 연산 처리
            if operator == "+":
                result = num1 + num2
            elif operator == "-":
                result = num1 - num2
            elif operator == "*":
                result = num1 * num2
            elif operator == "/":
                if num2 == 0:
                    raise ZeroDivisionError("0으로 나눌 수 없습니다.")
                result = num1 / num2
            else:
                raise ValueError("지원하지 않는 연산자입니다.")

            print(f"결과: {result}\n")

        # 예외 처리
        except ZeroDivisionError as zde:
            print(f"오류: {zde}\n")
        except ValueError as ve:
            print(f"오류: {ve}\n")
        except Exception as e:
            print(f"입력 오류: {e}. 형식에 맞게 입력해주세요.\n")

# 프로그램 실행
calculator()
```

## 【Peer 평가 Check List】 (파이썬 기초) 간단한 계산기 프로그램 작성

설명자 Explainer		평가자 Evaluator		날짜 date	
<input type="checkbox"/> 전혀 설명하지 못함 <input type="checkbox"/> 기본적인 설명은 했으나 부족함 <input type="checkbox"/> 완벽하게 이해하고 명확히 설명함					

## 1. 기본 문법 이해

- ☐ 변수 선언과 자료형 변환(float)의 개념을 정확히 설명했는가?
- ☐ input()과 print() 함수의 역할을 이해하고 활용 방법을 설명했는가?
- ☐ input() 함수로 읽어온 사용자 입력을 strip()한 이유를 이해하고 있는가?

## 2. 코드 구조와 흐름 이해

- ☐ while 반복문이 코드에서 어떤 역할을 하는지 설명했는가?
- ☐ 종료 조건(if user\_input.lower() == "exit")이 프로그램 흐름에 어떤 영향을 미치는지 설명했는가?
- ☐ 사용자 입력(num1, operator, num2)을 split()으로 분리하는 이유를 명확히 이해했는가?

### 3. 조건문 및 연산 처리

- ☐ 조건문(if, elif, else)을 사용하여 연산자를 처리하는 방식을 정확히 설명했는가?
- ☐ 각 연산(+, -, \*, /)의 동작과 결과가 코드에서 어떻게 구현되는지 설명했는가?

#### 4. 예외 처리

- ☐ try, except 구문이 오류를 처리하는 원리를 이해하고 설명했는가?
- ☐ 0으로 나누기(ZeroDivisionError)와 잘못된 입력(ValueError)을 처리하는 방식을 정확히 설명했는가?

## 5. 확장 가능성과 실무 응용

- ☐ 프로그램이 종료되지 않고 반복 동작하는 구조가 실무에서 어떤 응용 가능성이 있는지 논의했는가?
- ☐ 이 프로그램을 실무(예: 데이터 처리 도구)에서 어떻게 활용할 수 있을지 아이디어를 제시했는가?

## 6. 상호 이해와 협력

- ☐ 상대방이 이해하지 못하는 부분에 대해 추가적으로 설명하려는 노력을 보였는가?
- ☐ 상대방의 설명을 경청하고, 피드백 또는 질문을 통해 협력적으로 학습했는가?

## 7. 발표자의 강점과 약점 (feedback)

--

## ⑤ [도전과제] 계산기 프로그램 기능 확장 및 개선

### - 요구사항:

#### 1. 새로운 연산 기능 추가

- 사용자가 % 연산자를 입력하면 두 숫자의 **나머지(Modulus)**를 계산하여 출력하도록 기능을 추가합니다.

#### 2. 연산 기록 기능

- 사용자가 수행한 모든 연산을 리스트에 저장하고, "history" 명령어를 입력하면 지금까지의 연산 기록을 출력하도록 수정합니다.
- 출력 예

```
5 + 3 = 8
10 / 2 = 5.0
```

#### 3. 프로그램 종료 확인

- 사용자가 "exit"를 입력하기 전에, 정말 종료할 것인지 묻는 확인 메시지를 추가합니다.

```
정말 종료하시겠습니까? (y/n): n
```

#### 【Hint】

① 연산 기록 저장 자료 구조

```
history = []
```

② 연산 기록 추가

```
history.append(f"{num1} {operator} {num2} = {result}")
```