

# 요구 추출 & 분석 및 정의

## <요구 추출 질문>

1. 계산의 실수 정확도는 소수 몇번째 자리까지 확보되어야 하나요?
2. 계산 내역 저장은 프로그램 실행 중에만 필요한가요? 파일로도 저장하길 원하나요?
3. 에러메시지 형식 중 사전에 정해두신게 있나요?
4. 계산기에 현재 시간 표시가 필요한가요?
5. 계산 화면에 들어와야 하는 정보들이 추가로 있나요?

## 💬 고객 답변

### 1. 소수점 정확도

- 기본적으로 **소수점 아래 두 자리**까지만 표시되면 충분합니다.
- 다만, 나눗셈 결과가 긴 경우에는 **최대 4자리까지 반올림**되는 형태를 원합니다.
- 예: `10 / 3 → 3.3333`

### 2. 계산 내역 저장 방식

- 프로그램을 종료하면 기록은 초기화되어도 괜찮습니다.
- 다만, 실행 중에는 **최근 5개 계산 결과**를 조회할 수 있으면 좋겠어요.

### 3. 에러 메시지 형식

- 특별히 정해진 건 없지만,  
단순하게 `"Error: 잘못된 입력입니다."`  
이런 형태면 충분합니다.
- 단, 프로그램이 멈추지 않고 다시 입력받을 수 있어야 합니다.

### 4. 현재 시간 표시 여부

- 필요 없습니다.
- 계산 기능만 간결하게 유지해 주세요.

### 5. 계산 화면 정보

- 계산식 입력란과 결과 출력만 있으면 됩니다.
- 추가로, "이전 계산 보기" 명령을 입력하면  
최근 5개의 계산 내역이 출력되면 좋겠어요.

이제 호진 씨가 다음 단계로 해야 할 것은 **요구 분석 및 정의 단계**입니다.

즉, 위 답변을 정리해서 다음처럼 구성하면 됩니다 📌

- 기능적 요구와 비기능적 요구를 구분하고,
- 품질 속성(명확성, 완전성, 검증가능성 등)을 충족하는 형태로 **요구 명세 테이블**을 작성.

## <기능요구>

ID	요구명	요구 설명	우선순위	검증 가능성	비고
FR-1	기본 사칙연산 기능	덧셈, 뺄셈, 곱셈, 나눗셈을 수행한다.	★★★★★	단위 테스트로 검증 가능	
FR-2	퍼센트 계산 기능	입력값에 대해 퍼센트 연산을 수행한다.	★★★★☆	테스트 입력값 50% of 200 등으로 확인	
FR-3	계산 내역 저장 기능	최근 5개의 계산 결과를 메모리 상에 저장하고 조회한다.	★★★★☆	콘솔에서 "history" 명령으로 확인	
FR-4	오류 처리 기능	잘못된 입력 발생 시 "Error: 잘못된 입력입니다." 출력 후 재입력 루프 복귀	★★★★☆	예외 입력 테스트로 확인	
FR-5	결과 표시 형식	결과는 기본적으로 소수점 2자리, 최대 4자리까지 반올림 표시	★★★★☆	출력 형식 검사로 확인	
FR-6	계산 화면 정보	화면은 '계산식 입력란과 출력란 2개로만 구성'	★★★★☆	화면 출력 검사로 확인	

## <비기능 요구>

ID	항목	요구 설명	측정 기준	검증 방법
NFR-1	성능	계산 결과는 0.5초 이내에 출력되어야 한다.	0.5초 이하	실행 시간 측정
NFR-2	품질	예외 발생 시에도 프로그램이 종료되지 않아야 한다.	지속 실행	예외 처리 테스트
NFR-3	사용성	초보자도 사용법을 쉽게 이해할 수 있어야 한다.	도움말 제공 여부	프로그램 실행 시 안내문 확인

ID	항목	요구 설명	측정 기준	검증 방법
NFR-4	이식성	Python 3.12 이상에서 OS에 관계없이 실행 가능해야 한다.	다양한 OS 테스트	macOS, Windows에서 실행 테스트
NFR-5	보안	외부 입력은 산술식으로만 제한되어야 한다.	외부 코드 실행 차단	악의적 입력 테스트
NFR-6	품질	결과는 기본적으로 소수점 2자리, 최대 4자리까지 반올림 표시	출력값 소수점 2자리 ~ 4자리 표시	출력 형식 검사로 확인

## 최종

ID	유형	요구사항 설명	우선순위	검증 방법
FR-1	기능	덧셈·뺄셈·곱셈·나눗셈을 수행한다.	★★★★★	단위 테스트
FR-2	기능	% 기호는 퍼센트(%) 연산자로 사용한다. 예: <code>200 * 12% → 24</code> , <code>50% → 0.5</code>	★★★★☆	테스트 입력값으로 검증
FR-3	기능	%% 기호는 나머지 연산으로 사용한다. 예: <code>7 % 3 → 1</code>	★★★★☆	단위 테스트
FR-4	기능	최근 5개의 계산 결과를 메모리에 저장하고, <code>history</code> 명령으로 조회할 수 있다.	★★★★☆	콘솔 출력 확인
FR-5	기능	잘못된 입력 시 <code>"Error: 잘못된 입력입니다."</code> 출력 후 재입력 대기 상태로 복귀한다.	★★★★☆	예외 입력 테스트
FR-6	기능	화면은 "입력란(프롬프트)"과 "출력란(결과)"으로만 구성된다.	★★★★☆	콘솔 출력 검증
NFR-1	비기능	결과는 기본적으로 소수점 2자리, 최대 4자리까지 반올림 표시한다.	★★★★☆	출력 형식 검사
NFR-2	비기능	계산 결과는 0.5초 이내에 출력되어야 한다.	★★★★★	실행 시간 측정
NFR-3	비기능	예외 발생 시에도 프로그램은 종료되지 않는다.	★★★★★	예외 테스트
NFR-4	비기능	프로그램 실행 시 기본 사용법 안내문을 표시해야 한다.	★★★★☆	콘솔 실행 확인
NFR-5	비기능	Python 3.12 이상에서 OS 관계없이 실행 가능해야 한다.	★★★★☆	OS별 테스트
NFR-6	비기능	외부 입력은 산술식으로만 제한하고, <code>eval</code> 등 코드 실행은 차단해야 한다.	★★★★★	악의적 입력 테스트