

# asdfasdf

작성자: DeepSeek API  
작성일자: 2025-01-25

PR ID	제목	평균 등급	작성일자
1	Revert "[feat/#57] 보고서 제작에 DEE...	A	2025-01-23
2	Revert "[feat/#57] 보고서 제작에 DEE...	A	2025-01-23
3	Revert "[feat/#57] 보고서 제작에 DEE...	A	2025-01-23
4	Revert "[feat/#57] 보고서 제작에 DEE...	A	2025-01-23
5	Revert "[feat/#57] 보고서 제작에 DEE...	A	2025-01-23

## 2. 분석 내용

### 2-1. 리뷰 결과 통계

- 분석된 PR 수 : 5
- Clean 모드 : 5개의 리뷰
- Optimize 모드 : 0개의 리뷰
- Study 모드 : 0개의 리뷰
- newbie 모드 : 0개의 리뷰
- basic 모드 : 0개의 리뷰

### 2-2. 주요 취약점 및 개선 우선순위

취약한 유형 통계 및 개선 방향 :

#### 1. 취약 문제 유형 : 코드 가독성

- 개선 방향 : 변수명과 함수명을 명확하게 작성하여 코드의 의도를 명확히 전달하세요.
- 개선 전 코드 :

```
def f(a, b):  
    return a + b
```

- 개선 후 코드 :

```
def add_numbers(num1, num2):
```

```
return num1 + num2
```

## 2. 취약 문제 유형 : 중복 코드

- 개선 방향 : 중복된 코드를 함수로 추출하여 재사용성을 높이세요.

- 개선 전 코드 :

```
def calculate_area(radius):  
    return 3.14 * radius * radius  
def calculate_circumference(radius):  
    return 2 * 3.14 * radius
```

- 개선 후 코드 :

```
def calculate_area(radius):  
    return math.pi * radius ** 2  
def calculate_circumference(radius):  
    return 2 * math.pi * radius
```

## 3. 취약 문제 유형 : 예외 처리 부재

- 개선 방향 : 예외 상황을 고려하여 적절한 예외 처리를 추가하세요.

- 개선 전 코드 :

```
def divide(a, b):  
    return a / b
```

- 개선 후 코드 :

```
def divide(a, b):  
    if b == 0:  
        raise ValueError("Cannot divide by zero")  
    return a / b
```

---

## 2-3. 개인화된 피드백 및 권장사항

### 사용자 맞춤 개선 방향 :

- 가장 낮은 점수를 받은 평가 기준은 **코드 가독성** 입니다. 코드의 가독성을 높이기 위해 변수명과 함수명을 명확하게 작성하고, 코드의 의도를 명확히 전달하는 데 집중하세요. 또한, 주석을 적절히 활용하여 코드의 목적과 동작 방식을 설명하는 것도 좋은 방법입니다.

---

## 2-4. 종합 결론

- 총평

- 프로젝트의 전체적 성향은 기본적인 기능 구현에 충실하지만, 코드의 가독성과 중복 코드, 예외 처리 부분에서 개선 여지가 큼니다. 이러한 부분을 보완한다면 더욱 견고하고 유지보수하기 쉬운 코드를 작성할 수

있을 것입니다.

- **강점**

1. 기본적인 기능 구현이 잘 되어 있습니다.
2. 코드의 구조가 간결하고 이해하기 쉽습니다.
3. 필요한 기능을 빠르게 구현할 수 있는 능력이 있습니다.

- **약점**

1. 코드의 가독성이 낮아 이해하기 어려운 부분이 있습니다.
2. 중복 코드가 많아 유지보수가 어렵습니다.
3. 예외 처리가 부족하여 안정성이 떨어집니다.

- **향후 권장 사항** : **Clean 모드** 를 사용하며 코드의 가독성과 중복 코드를 줄이는 데 집중하세요. 또한, 예외 처리를 추가하여 코드의 안정성을 높이는 연습을 해보세요.

---

**첨부 자료**

- **Clean Code: A Handbook of Agile Software Craftsmanship**

<https://www.amazon.com/Clean-Code-Handbook-Software-Craftsmanship/dp/0132350882>

- **Refactoring: Improving the Design of Existing Code**

<https://www.amazon.com/Refactoring-Improving-Design-Existing-Code/dp/0201485672>

- **Effective Python: 90 Specific Ways to Write Better Python**

<https://www.amazon.com/Effective-Python-Specific-Software-Development/dp/0134853989>

---