

# 프로젝트 리뷰 보고서

작성자: DeepSeek API  
작성일자: 2025-01-19

## 1. 리뷰 데이터 요약

PR ID	제목	평균 등급	리뷰 작성일자
6	Test PR Review 1	B	2025-01-19 11:56:07
7	Test PR Review 2	B	2025-01-19 11:56:07
8	Test PR Review 3	B	2025-01-19 11:56:07
9	Test PR Review 4	B	2025-01-19 11:56:07
10	Test PR Review 5	B	2025-01-19 11:56:07

## 2. 분석 내용

---

**\*\*2-1. 리뷰 결과 통계\*\***

- \*\*분석된 PR 수\*\*: 5
- \*\*Clean 모드\*\*: 5개의 리뷰
- \*\*Optimize 모드\*\*: 0개의 리뷰

---

**\*\*2-2. 주요 취약점 및 개선 우선순위\*\***

**\*\*취약한 유형 통계 및 개선 방향\*\*:**

**1. \*\*문제점\*\*: 코드 가독성 저하**

- \*\*개선 방향\*\*: 변수명과 함수명을 명확하게 작성하고, 불필요한 주석을 제거하여 코드의 가독성을 높이세요.
- \*\*관련 코드 예시\*\*:

```
```python
```

# 기존 코드

```
a = 10 # a는 카운트 변수
```

```
b = 20 # b는 최대값
```

# 개선된 코드

```
count = 10
```

```
max_value = 20
```

```
...
```

2. **\*\*문제점\*\***: 중복 코드

- **\*\*개선 방향\*\***: 중복된 코드를 함수로 추출하여 재사용성을 높이세요.

- **\*\*관련 코드 예시\*\***:

```
```python
```

# 기존 코드

```
def calculate_area_of_circle(radius):
```

```
    return 3.14 * radius * radius
```

```
def calculate_area_of_square(side):
```

```
    return side * side
```

# 개선된 코드

```
def calculate_area(shape, *args):
```

```
    if shape == "circle":
```

```
        return 3.14 * args[0] * args[0]
```

```
    elif shape == "square":
```

```
        return args[0] * args[0]
```

```
...
```

3. **\*\*문제점\*\***: 예외 처리 부재

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

- \*\*관련 코드 예시\*\*:

```
```python
```

```
# 기존 코드
```

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

```
    return a / b
```

```
# 개선된 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 기준\*\*:** 코드 가독성

- **\*\*개선 방향\*\*:** 변수명과 함수명을 명확하게 작성하고, 불필요한 주석을 제거하여 코드의 가독성을 높이세요.

- **\*\*구체적인 개선 방향과 코드\*\*:**

```
```python
```

```
# 기존 코드
```

```
x = 5 # x는 사용자 수
```

```
y = 10 # y는 최대 사용자 수
```

```
# 개선된 코드
```

```
user_count = 5
```

```
max_user_count = 10
```

```
```
```

---

## **\*\*2-4. 종합 결론\*\***

### **- \*\*프로젝트 평가\*\*:**

**- \*\*강점\*\*:** 코드의 기본 구조는 잘 작성되어 있으며, 기능 구현이 명확합니다.

**- \*\*개선이 필요한 영역\*\*:** 코드 가독성, 중복 코드 제거, 예외 처리 추가가 필요합니다.

### **- \*\*향후 권장 사항\*\*:**

**- \*\*클린 코드 모드\*\*:** 코드 가독성과 유지보수성을 높이기 위해 클린 코드 모드를 적극 활용하세요.

**- \*\*최적화 모드\*\*:** 성능 최적화가 필요한 부분에 대해 최적화 모드를 활용하여 성능을 개선하세요.

---

## **\*\*첨부 자료\*\***

### **- \*\*추천 학습 자료\*\*:**

- [Clean Code by Robert C. Martin](<https://www.amazon.com/Clean-Code-Handbook-Software-Craftsmanship/dp/0132350882>)

- [Refactoring: Improving the Design of Existing Code by Martin Fowler](<https://www.amazon.com/Refactoring-Improving-Design-Existing-Code/dp/0201485672>)

### **- \*\*관련 예시 코드\*\*:**

```
```python
```

```
# 클린 코드 예시
```

```
def calculate_area(shape, *args):
```

```
    if shape == "circle":
```

```
        return 3.14 * args[0] * args[0]
```

```
    elif shape == "square":
```

```
        return args[0] * args[0]
```

```
    else:
```

```
        raise ValueError("Unsupported shape")
```

```
```
```

---

### 3. 결론

강점:

- 기능이 잘 동작한다.
- 구조가 단순하다.

약점:

- 코드 중복이 많다.
- 성능 최적화가 필요하다.

권장 사항:

- 클린 코드 적용
- 성능 개선