

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**:**

- **문제점**:** 코드 가독성 저하 (3/5 PR에서 지적)
 - ****개선 방향**:** 변수 및 함수명을 직관적으로 변경하고, 불필요한 주석을 제거하세요.
 - ****관련 코드 예시**:**

```
python
# 기존 코드
```

```
a = 10 # a는 나이
```

```
b = 20 # b는 점수
```

```
# 개선된 코드
```

```
age = 10
```

```
score = 20
```

```
...
```

2. ****문제점****: 중복 코드 (2/5 PR에서 지적)

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

- ****관련 코드 예시****:

```
```python
```

```
기존 코드
```

```
def calculate_area(radius):
```

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

```
def calculate_circumference(radius):
```

```
 return 2 * 3.14 * radius
```

```
개선된 코드
```

```
def calculate_area(radius):
```

```
 return math.pi * radius ** 2
```

```
def calculate_circumference(radius):
```

```
 return 2 * math.pi * radius
```

```
...
```

3. **\*\*문제점\*\***: 예외 처리 부재 (1/5 PR에서 지적)

- **\*\*개선 방향\*\***: 예외 상황을 고려하여 try-except 블록을 추가하세요.

- **\*\*관련 코드 예시\*\***:

```
```python
```

기존 코드

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

```
    return a / b
```

개선된 코드

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

```
    try:
```

```
        return a / b
```

```
    except ZeroDivisionError:
```

```
        return "Division by zero is not allowed"
```

```
    ...
```

```
    ---
```

****2-3. 개인화된 피드백 및 권장사항****

****사용자 맞춤 개선 방향**:**

- ****가장 낮은 점수를 받은 평가 기준**:** 코드 가독성

- ****개선 방향**:** 변수 및 함수명을 직관적으로 변경하고, 불필요한 주석을 제거하세요.

- ****구체적인 개선 방향**:**

- 변수명은 명사로, 함수명은 동사로 시작하도록 변경.

- 불필요한 주석은 제거하고, 필요한 경우 코드 자체를 설명할 수 있도록 리팩토링.

- ****코드 예시**:**

```
```python
```

# 기존 코드

```
def f(x): # x는 나이
```

```
 return x * 2 # 나이를 두 배로
```

# 개선된 코드

```
def double_age(age):
```

```
 return age * 2
```

...

---

## **\*\*2-4. 종합 결론\*\***

### **- \*\*프로젝트 평가\*\*:**

- **\*\*강점\*\***: 대부분의 코드가 기능적으로 잘 동작하며, 기본적인 구조는 잘 잡혀 있습니다.

- **\*\*개선\*\***이 필요한 영역: 코드 가독성과 중복 코드, 예외 처리 부분에서 개선이 필요합니다.

### **- \*\*향후 권장 사항\*\*:**

- **\*\*클린 코드 모드\*\***: 코드 가독성과 중복 코드를 개선하기 위해 클린 코드 모드를 적극 활용하세요.

- **\*\*최적화 모드\*\***: 성능 최적화가 필요한 경우, 최적화 모드를 활용하여 코드의 효율성을 높이세요.

---

## **\*\*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
```

```
# 예외 처리 예시
```

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

```
    try:
```

```
        return a / b
```

```
    except ZeroDivisionError:
```

```
        return "Cannot divide by zero"
```

```
    except TypeError:
```

```
        return "Invalid input types"
```

중복 코드 제거 예시

```
def calculate_area(radius):
```

```
    return math.pi * radius ** 2
```

```
def calculate_circumference(radius):
```

```
    return 2 * math.pi * radius
```

```
...
```

```
---
```

3. 결론

강점:

- 기능이 잘 동작한다.
- 구조가 단순하다.

약점:

- 코드 중복이 많다.
- 성능 최적화가 필요하다.

권장 사항:

- 클린 코드 적용
- 성능 개선