프로젝트 리뷰 보고서

작성자: DeepSeek API
작성일자: 2025-01-20

1. 리뷰 데이터 요약

PR ID	제목	평균 등급	리뷰 작성일자
1	test	Pending	2025-01-19 16:33:53
2	test	Pending	2025-01-20 06:27:27
3	test	Pending	2025-01-20 06:29:03
4	test	Pending	2025-01-20 06:29:40
5	test	Pending	2025-01-20 06:29:54

2. 분석 내용

2-1. 리뷰 결과 통계

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

2-2. 주요 취약점 및 개선 우선순위

취약한 유형 통계 및 개선 방향:

1. **문제점**: 코드 중복

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

- **관련 코드 예시**:

```python

# Before
```

def calculate_area_of_circle(radius):

return 3.14 * radius * radius

def calculate_area_of_square(side):

return side * side

After

def calculate_area(shape, *args):

if shape == "circle":

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

elif shape == "square":

return args[0] * args[0]

```

2. **문제점**: 불필요한 주석

- **개선 방향**: 코드 자체가 설명이 될 수 있도록 명확한 변수명과 함수명을 사용하세요.

- **관련 코드 예시**:

```python

# Before

# This function adds two numbers

def add(a, b):

return a + b

# After

def add_numbers(first_number, second_number):

return first_number + second_number

```

3. **문제점**: 긴 함수

- **개선 방향**: 함수를 더 작은 단위로 분리하여 가독성을 높이세요.
```

- **관련 코드 예시**:

```python

# Before

def process_data(data):

# Step 1: Validate data

if not data:

return None

# Step 2: Transform data

transformed_data = [item * 2 for item in data]

# Step 3: Save data

save_to_database(transformed_data)

# After

def validate_data(data):

return bool(data)

def transform_data(data):

return [item * 2 for item in data]

def process_data(data):

if not validate_data(data):

return None

transformed_data = transform_data(data)

save_to_database(transformed_data)

```

2-3. 개인화된 피드백 및 권장사항

사용자 맞춤 개선 방향:

- **가장 낮은 점수를 받은 평가 기준**: 코드 중복

- **개선 방향**: 중복된 코드를 함수로 추출하여 재사용성을 높이세요. 이를 통해 코드의 유지보수성이 향상되고, 버그 발생 가능성을 줄일 수 있습니다.

- **구체적인 개선 방향 및 코드**:

```python

# Before

def calculate_area_of_circle(radius):

return 3.14 * radius * radius


def calculate_area_of_square(side):

return side * side


# After

def calculate_area(shape, *args):

if shape == "circle":

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

elif shape == "square":

return args[0] * args[0]

```

2-4. 종합 결론

- **프로젝트 평가**:

- **강점**: 코드의 기본적인 구조는 잘 잡혀 있으며, 기능 구현이 명확합니다.

- **개선이 필요한 영역**: 코드 중복, 불필요한 주석, 긴 함수 등이 주요 개선 대상입니다.

- **향후 권장 사항**:

- **클린 코드 모드**: 코드 중복을 줄이고, 함수를 더 작은 단위로 분리하여 가독성을 높이는 데 클린 코드 모드를 활용하세요.

- **최적화 모드**: 성능 최적화가 필요한 부분에 대해 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

# Example of clean code

def calculate_area(shape, *args):

if shape == "circle":

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

elif shape == "square":

return args[0] * args[0]

```

---

3. 결론

강점:
- 기능이 잘 동작한다.
- 구조가 단순하다.

약점:
- 코드 중복이 많다.
- 성능 최적화가 필요하다.

권장 사항:
- 클린 코드 적용
- 성능 개선