Задание №1

1. Цели ревью кода

1. Убедиться, что функция правильно выполняет сложение двух чисел

2. Убедиться, что функция корректно обрабатывает некорректные входные данные (например, нечисловые значения)

3. Убедиться, что код соответствует стандартам стиля и легко читается

1. Задачи для ревьюера

1. Убедиться, что функция возвращает правильный результат для различных комбинаций входных данных

2. Убедиться, что функция корректно обрабатывает различные типы данных, включая нечисловые значения

3. Убедиться, что код соответствует стандартам стиля

4. Убедиться, что функция корректно обрабатывает исключения, такие как переполнение или некорректные типы данных

1. Объекты проверки

1. Проверить, как функция обрабатывает входные данные, которые не являются числами (строки, списки)

2. Проверить, как функция обрабатывает очень большие числа

3. Проверить, как функция обрабатывает различные типы данных, такие как целые числа, числа с плавающей запятой

4. Убедиться, что функция возвращает правильный результат для различных комбинаций входных данных.

5. Убедиться, что код соответствует стандартам стиля (отступы, именование переменных и т.д)

Задание №2

|  |  |  |  |
| --- | --- | --- | --- |
| Пункты | Вопросы | | |
| 1.Работоспособность кода | Выполняет ли код свои обязанности? | Корректна ли логика выполнения задач |  |
| 2.Понятность кода | Прост ли код для понимания? | Не слишком ли длинные названия методов и переменных? | Используются ли понятные и описательные имена для переменных и функций? |
| 3.Соответствие кода стилю оформления | Правильно ли названы пространства имен, классы, методы и переменные? | Соответствуют ли имена стилю snake.case для переменных и функций, и CamelCase для классов |  |
| 4.Избыточность кода | Отсутствуют ли повторяющиеся части кода, которые можно вынести в отдельные функции? | Нет ли дублирования логики, которую можно унифицировать? |  |
| 5.Корректная обработка исключений | Используются ли исключения по назначению? | Сохраняется ли информация о произошедшей ошибке? | Есть ли обработка исключений для всех потенциально опасных участков кода? |
| 6.Безопасность | Проверяются ли все входные данные на корректный тип, длину, формат и диапазон? | Есть ли защита от потенциальных уязвимостей, таких как SQL-инъекции, XSS и т.д.? |  |
| 7.Документация | Есть ли комментарии, которые раскрывают смысл кода? | Прокомментированы ли все функции и их параметры? | Есть ли документация для модуля и его основных компонентов? |
| 8.Тестирование | Есть ли тесты для кода? | Достаточно ли тестов для покрытия основных сценариев использования? | Проверяют ли юнит-тесты, что код предоставляет требуемую функциональность? |
| 9.Соответствие PEP-8 / Google Java Style | Соответствует ли код стандарту PEP 8 для Python? | Если используется другой язык, соответствует ли код соответствующему стандарту стиля? |  |
| 10.Наличие unit-тестов для всех методов |  |  |  |
| 11.Проверка уязвимостей (SQL-инъекции) | Проведена ли проверка кода на наличие уязвимостей, таких как SQL-инъекции? | Используются ли параметризованные запросы для работы с базами данных? | Есть ли защита от других распространенных уязвимостей |