**Анализ функции calculate\_positive\_average**

**1. Защита от деления на ноль (строка A)**

Деление на ноль исключено благодаря предусловию:

python

Copy

Download

assert len(numbers) > 0 *# указано в спецификации*

Это гарантирует, что:

* count = len(numbers) всегда больше нуля,
* знаменатель count никогда не будет равен нулю.

**2. Обработка пустого списка**

Если numbers = []:

* Нарушается предусловие len(numbers) > 0,
* Функция вызывает AssertionError,
* Не выполняется постусловие, предполагающее возврат числового значения.

**Проблема:** Спецификация требует возврата 0 для пустого списка, но текущая реализация этого не поддерживает.

**3. Возможные ошибки и пути улучшения**

**Типичные ошибки при нарушении контракта:**

* Передача строки вместо списка → TypeError при итерации,
* Наличие отрицательных чисел → некорректный результат (логическая ошибка).

**Рекомендации по улучшению:**

* **Проверка типа входных данных:**

python

Copy

Download

if not isinstance(numbers, list):

raise TypeError("Input must be a list")

* **Обработка крайних случаев:**

python

Copy

Download

if not numbers or not all(n > 0 for n in numbers):

return 0.0 *# соответствует постусловию*

**4. Выводы о корректности**

**Оценка соответствия спецификации:**  
❌ **Не полностью корректна** по следующим причинам:

* Не обрабатывает пустой список в соответствии с постусловием,
* Нет защиты от неверных типов данных,
* Молчаливо игнорирует отрицательные числа.

**Рекомендации по тестированию:**  
Для выявления потенциального деления на ноль:

* **Лучший метод:** Статический анализ (например, PyLint/SonarQube), так как он обнаруживает проблему без запуска кода.

**Дополнительные меры:**

* Тестирование выявило бы ошибку при пустом списке,
* Формальная верификация подтвердила бы нарушение контракта.

**Итог:** Функция требует доработки для полного соответствия спецификации и повышения надежности.