**Анализ функции calculate\_positive\_average**

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

Функция избегает деления на ноль благодаря предусловию:

python

Copy

Download

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

Это обеспечивает, что:

* count = len(numbers) всегда больше нуля,
* знаменатель count никогда не станет нулевым.

**2. Обработка пустого списка**

Если numbers = []:

* Нарушается предусловие len(numbers) > 0,
* Функция вызывает AssertionError,
* Постусловие (возврат числового значения) не выполняется.

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

**3. Возможные ошибки и улучшения**

**Типичные ошибки при нарушении контракта:**

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

**Рекомендации по доработке:**

* **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**

Функция **не полностью соответствует спецификации** из-за:

* Отсутствия обработки пустого списка согласно постусловию,
* Недостаточной защиты от неверных типов данных,
* Игнорирования отрицательных чисел без явного уведомления.

**Рекомендации по тестированию:**

* **Лучший метод:** Статический анализ (PyLint, SonarQube), так как он выявляет проблему без выполнения кода.
* **Дополнительные меры:**
  + Тестирование для обнаружения ошибок при пустом списке,
  + Формальная верификация для проверки соблюдения контракта.

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