### Анализ функции calculate\_positive\_average

#### 1. Гарантия от деления на ноль в строке A

Деление на ноль невозможно благодаря **предусловию**:

python

Copy

Download

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

Это условие гарантирует, что:

* count = len(numbers) всегда > 0
* Знаменатель count никогда не будет нулевым

#### 2. Обработка пустого списка

При numbers = []:

* Нарушается предусловие len(numbers) > 0
* Функция выбросит исключение AssertionError
* **Не соответствует** постусловию, где предполагается возврат числа

Проблема: Спецификация требует обработки пустого списка (возврат 0), но реализация этого не обеспечивает.

#### 3. Потенциальные ошибки и улучшения

**Типичные сбои при нарушении контракта:**

* Передача строки вместо списка: TypeError при попытке итерации
* Отрицательные числа: некорректный результат (нарушение логики)

**Способы улучшения:**

1. Явная проверка типа:

python

Copy

Download

if not isinstance(numbers, list):

raise TypeError("Input must be a list")

1. Обработка edge-кейсов:

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 корректности:**

* ❌ Не полностью корректна относительно заявленной спецификации
* Причины:
  + Не обрабатывает пустой список согласно постусловию
  + Нет защиты от неверных типов данных
  + Молчаливое проглатывание отрицательных чисел

**Рекомендации по проверке:**  
Для обнаружения ошибки деления на ноль:

* **Наиболее эффективен:** Статический анализ
* **Причина:** Может детектировать потенциальное деление на ноль без выполнения кода
* **Пример:** PyLint/SonarQube покажут предупреждение "Possible division by zero"

**Дополнительно:**

* Тестирование выявило бы проблему с пустым списком
* Формальная верификация доказала бы нарушение контракта

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