### ****Практическая часть (анализ кода)****

## ****1. Спецификация****

### Предусловия для функции calculate\_positive\_average:

Функция ожидает на вход список чисел. То есть, переменная numbers должна быть итерируемой — например, списком, кортежем и т.д., и содержать в себе числа (можно и отрицательные, и ноль, и положительные). Если передать что-то другое, например строку или число вместо списка, то всё сломается.

### Постусловие:

Функция возвращает среднее арифметическое всех положительных чисел из списка. Если таких чисел нет (или список пуст), то вернётся 0.

### Инвариант цикла:

Да, можно сказать, что инвариант есть. На каждой итерации цикла:

Переменная total хранит сумму всех положительных чисел, которые мы уже прошли.

Переменная count показывает, сколько таких чисел мы нашли до текущего момента.

И при переходе к следующей итерации это свойство сохраняется.

**2. Ручное тестирование**

Сделал 3 теста: один обычный случай, один граничный, и один с ошибочными данными.

| ВХОД (NUMBERS) | ОЖИДАЕМЫЙ РЕЗУЛЬТАТ | ФАКТИЧЕСКИЙ РЕЗУЛЬТАТ | ПРОШЁЛ? |
| --- | --- | --- | --- |
| [2, 4, 6, -1, 3] | 3.75 | 3.75 | ✅ |
| [-5, -2, 0] | 0 | 0 | ✅ |
| not a list | TypeError | TypeError | ✅ |

## ****3. Анализ корректности и ошибок****

### Почему в строке A не будет деления на ноль?

Потому что перед этим стоит проверка:

1: if count > 0:

То есть, если у нас был хотя бы один положительный элемент, только тогда мы считаем среднее. Иначе просто возвращаем 0. Так что деление происходит только когда count > 0.

### Как обрабатывается пустой список?

Если передать пустой список [], то count остаётся равным 0, значит условие if count > 0: не выполняется, и функция возвращает 0. Это соответствует постусловию — в списке нет положительных чисел, значит возвращаем 0.

### Что может сломаться, если нарушить предусловие?

Например, если передать строку вместо списка, то цикл for num in numbers: вызовет TypeError. Потому что строка тоже итерируемая, но её элементы — символы, и если они не числа, то при попытке сравнения num > 0 тоже возникнет ошибка.

#### Как улучшить функцию?

Можно добавить проверку :

1: if not isinstance(numbers, list):

2: raise TypeError("Ожидается список")

Или использовать блок try-except, чтобы ловить возможные ошибки и выводить понятное сообщение.

**4. Выводы**

### Считаешь ли ты функцию корректной относительно спецификации?

Да, считаю. Она правильно считает среднее положительных чисел и возвращает 0, если их нет. Тесты подтверждают это. Даже с пустым списком работает нормально.

### Какой метод лучше всего помог бы найти ошибку, если бы не было проверки count > 0?

В этом случае лучше всего подошёл бы **статический анализ** или **формальная верификация.** Они могут заранее заметить, что есть риск деления на ноль. Тестирование могло бы не поймать это, если не попадём в нужный случай.