

Задание 1. Проверка корректности вычисления скидок

Ситуация: в магазине действует программа лояльности, где покупателям предоставляются скидки в зависимости от их уровня в этой программе: базовый, серебряный и золотой. Чтобы избежать ошибок в расчётах, руководство решило опробовать систему с использованием модульного тестирования.

Задача — создать функцию, которая рассчитает скидку в зависимости от уровня покупателя в программе лояльности и суммы покупки, и написать тесты с использованием модуля unittest, чтобы проверить корректность работы функции.

Задание 2. Количество знаков препинания

Ситуация: коллега написал часть программы, которая работает с текстом. Задача функции, которую нам передали, — вычислять количество знаков препинания в полученной строке. Ниже представлена её **реализация**:

```
def count_punct_marks(string: str) -> int:
    total_count = 0
    for sym in ",.:!":
        total_count += string.count(sym)
    return total_count
```

Задача — проверить, верно ли работает функция, и в случае найденных ошибок исправить их. Для поиска ошибок использовать pytest.

Задание 3

Задача: создать функцию add_numbers(a, b), которая принимает два числа и возвращает их сумму. Написать тесты для проверки:

- корректности сложения двух положительных чисел;
- корректности сложения отрицательных чисел;
- корректности сложения нуля с любым числом.

Задание 4

Задача: создать функцию `is_palindrome(s)`, которая проверит, является ли строка палиндромом (читается одинаково слева направо и справа налево). Функция должна игнорировать регистр символов. Написать тесты для:

- палиндрома, состоящего из букв;
- палиндрома, состоящего из цифр;
- не палиндрома.
-

Задание 5

Задача: создать функцию `word_count(s)`, которая принимает строку и возвращает словарь, где ключи — слова из строки, а значения — количество их вхождений. Тесты должны проверять:

- корректность подсчёта слов;
- игнорирование регистра при подсчёте;
- работу функции с пустой строкой.