**Тема:** *Корректность программ. Эталоны и методы проверки корректности*

**Цель задания:**

1. Понять суть корректности программы и ее отличия от надежности.
2. Узнать основные эталоны (спецификации) корректности.
3. Познакомиться с базовыми методами проверки корректности (тестирование, статический анализ, доказательство инвариантов).
4. Научиться применять простейшие методы (ручное тестирование, анализ предусловий/постусловий) для оценки кода.

Теоретическая часть

1. Корректность

Определение корректности программы:

Программа считается корректной, если она выполняет свои функции в соответствии с заданной спецификацией (предусловиями, постусловиями и инвариантами). Корректность означает, что программа производит ожидаемые результаты для всех допустимых входных данных.

Отличие корректности от надежности:

Корректность относится к соответствию программы спецификации, а надежность — к способности программы работать без сбоев в различных условиях, включая нештатные ситуации.

Пример надежной, но некорректной программы:

Программа, которая всегда возвращает 0, даже если должна вычислять среднее значение. Она надежна (не падает), но некорректна (не соответствует спецификации).

2. Эталоны (спецификации) корректности

Спецификация программы:

Это формальное описание ожидаемого поведения программы, включая входные и выходные данные, условия и ограничения. Спецификация необходима для определения корректности, так как без нее невозможно проверить, соответствует ли программа требованиям.

Виды спецификаций:

Функциональные требования: Описывают, что должна делать программа.

Предусловия: Условия, которые должны быть выполнены перед вызовом функции (например, входной список не должен быть пустым).

Постусловия: Гарантии, которые выполняются после работы функции (например, возврат среднего значения положительных чисел).

Инварианты: Условия, которые сохраняются на протяжении выполнения цикла или функции (например, total и count всегда неотрицательны).

Спецификация для calculate\_positive\_average(numbers):

Предусловие:

Список numbers должен содержать хотя бы один положительный элемент.

Постусловие:

Функция возвращает среднее арифметическое всех положительных элементов списка numbers. Если таких элементов нет, возвращается 0.

3. Методы проверки корректности

Тестирование:

Суть: Проверка программы на конкретных входных данных.

Сильные стороны: Простота, возможность выявления ошибок в реальных сценариях.

Слабые стороны: Не покрывает все возможные случаи.

Тестовый сценарий: Набор входных данных, ожидаемых результатов и условий выполнения теста.

Статический анализ:

Суть: Анализ кода без его выполнения для выявления потенциальных ошибок.

Пример: Инструмент pylint может обнаружить деление на ноль или неиспользуемые переменные.

Формальные методы (доказательство корректности):

Суть: Математическое доказательство соответствия программы спецификации.

Инвариант цикла: Условие, которое сохраняется перед каждой итерацией цикла. Например, для calculate\_positive\_average инвариантом может быть:

total равно сумме всех положительных элементов, обработанных на текущий момент, а count — их количество.

Сравнение методов:

Для небольшой функции подходит ручное тестирование. Для большой системы эффективнее комбинация статического анализа и формальных методов.