**Отчет по самостоятельной работе №5**

**по дисциплине МДК 01.02 “Инструментальные средства разработки программного обеспечения”.**

Выполнил: студент

группы 319

Губайдуллина Камилла Рустемовна

Дата 1.12.2024

Цель работы: изучить процесс отладки отдельных модулей программного проекта, выявить типичные ошибки и научиться эффективно использовать инструменты отладки для их устранения.

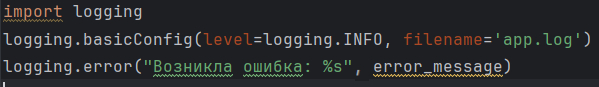
# Основная структура задания:

Задание №1: Подготовка среды для отладки

1. Настроила среду разработки для работы с проектом, инструментом отладки стал встроенный отладчик PyCharm.

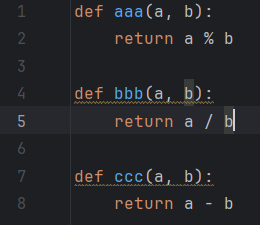
2. Убедилась, что отладчик корректно подключен к проекту, запустив сам отладчик через кнопку "Debug".

3. Определила стратегию логирования – библиотека Python logging.



Задание №2: Отладка логики модуля

1. Определила модуль, который содержит функции для математических операций и требует отладки.



2. Запустила отладчик, установила точки останова в ключевых местах, чтобы проверить значения переменных и выполнение условий.

3. Проверяя корректность входных данных, выявила ошибку – деление на нуль.

4. Задокументировала выявленные ошибки и рассмотрела способы их исправления. В Python деление на нуль вызывает исключение ZeroDivisionError. Есть несколько способов, как исправить это: обработка исключений с помощью try и except, проверка перед делением, использование функции, возвращающей None или другое значение при делении на ноль, логирование ошибки.

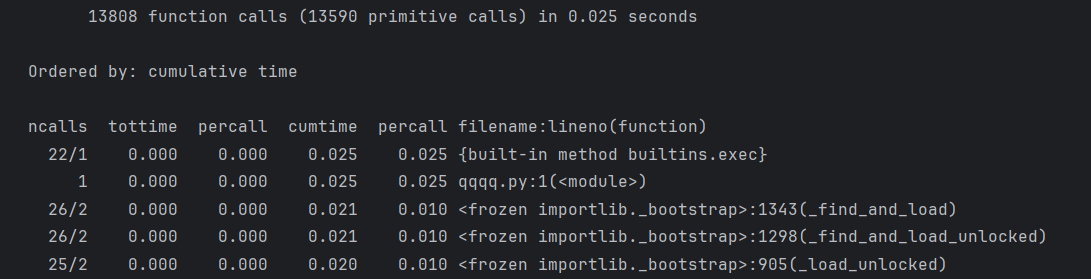
Задание №3: Интеграционная отладка

Провела тестирование на корректную работу модуля после интеграции с другими частями проекта, используя отладчик, проследила работу модуля в рамках всего проекта и убедилась в отсутствии конфликтов с другими модулями, затем выяснила, что модуль выполняет свои функции верно, исправление не требуется.

Задание №4: Отладка производительности и использования памяти

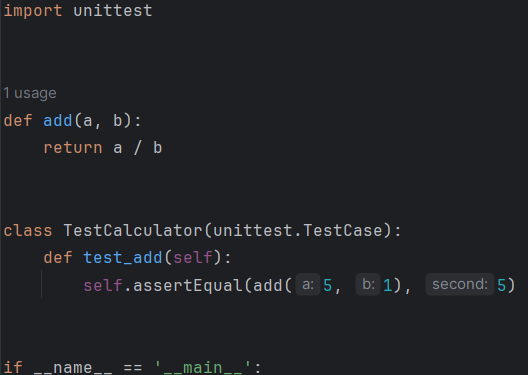
1. Использовала cProfile для анализа временных затрат

2. В ходе проведения тестов узнала, что модуль не подлежит оптимизации, ведь он использует достаточное количество памяти для его расчетов и действий.

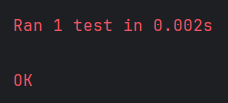


Задание №5: Автоматизированное тестирование и отладка

1. Были разработаны автоматизированные тесты для отладки выбранного модуля с использованием библиотеки unittest.



2. Исправлять ничего не нужно, ведь ошибок в модуле не наблюдается.



Задание №6: Отладка проблем с параллелизмом

В проекте не используется многопоточность или асинхронное программирование, поэтому провести отладку на предмет ошибок с параллелизмом не получится, но проверить его на корректность можно с помощью библиотеки threading. Также для решения проблем, связанных с параллелизмом, можно использовать блокировки и синхронизация (Lock, RLock, Semaphore и др.), профилирование и отладку (py-spy, objgraph), библиотеки (concurrent.futures – упрощает работу с потоками и процессами, предлагая интерфейсы ThreadPoolExecutor и ProcessPoolExecutor для выполнения задач асинхронно, joblib – подходит для параллельного выполнения задач и работы с массивами данных, что полезно в научных расчетах и машинном обучении).

# Выводы работы: в ходе практической работы, были изучены и применены основные методы отладки. Был настроен отладчик в PyCharm, установлены точки останова и проведена пошаговая отладка для выявления ошибок в модуле. Разработаны автоматизированные тесты для функций сложения, вычитания, умножения и деления, включая обработку исключений, таких как деление на ноль. Результаты работы подтвердили эффективность использования инструментов отладки и тестирования для обеспечения корректной работы программного кода.