**Отчет по самостоятельной работе №5**

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

Выполнила: студентка

группы 319

Попова Маргарита Вячеславовна

Дата 03.01.2024

**Цель работы**

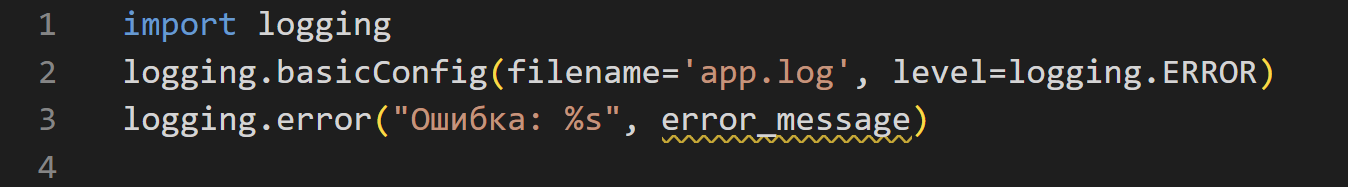
Изучить процесс отладки отдельных модулей программного проекта, выявить типичные ошибки и научиться эффективно использовать инструменты отладки для их устранения.

**Основная структура задания**

Задание 1: Подготовка среды для отладки

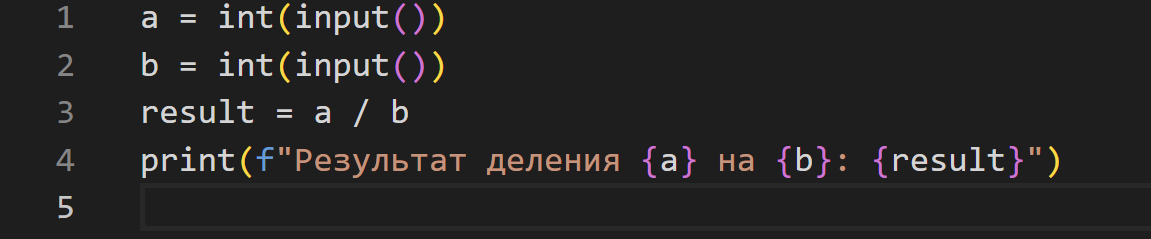
1. Настроили среду разработки для работы с проектом (встроенный отладчик в Visual Studio Code).

2. Убедились, что отладчик корректно подключен к проекту, и что можно запускать отладку по шагам, использовав breakpoint.

3. Определили стратегию логирования – используем библиотеку logging.

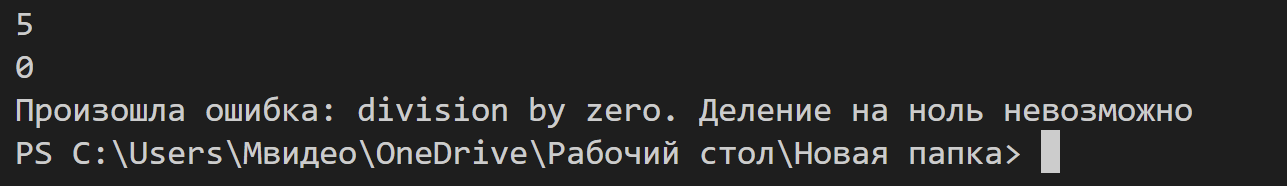
Задание 2: Отладка логики модуля

1. Выбрали модуль программного проекта, который требует отладки.

2. Используя пошаговую отладку, выявили ошибки в логике работы модуля.

3. Протестировали корректность входных данных и отлов исключений. Выявили, что отсутствует проверка деления на ноль.

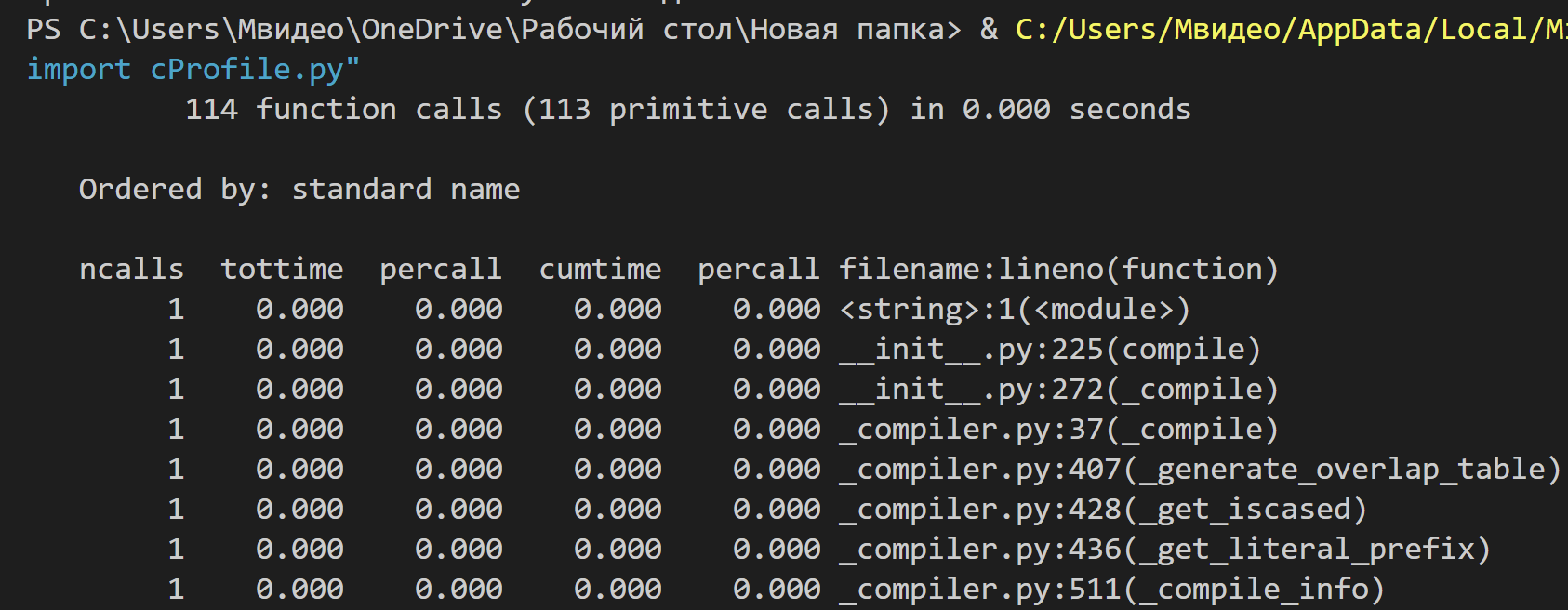
4. Задокументировали выявленные ошибки и предложили способы их исправления.

Ошибка деления на ноль: исключение ZeroDivisionError возникает, когда делитель равен нулю. Обработку стоит выполнять, если есть вероятность передачи нулевого значения в качестве делителя. проверьте значение делителя перед выполнением операции деления. Если делитель равен нулю, выведите соответствующее сообщение и либо повторите запрос у пользователя, либо завершите операцию.

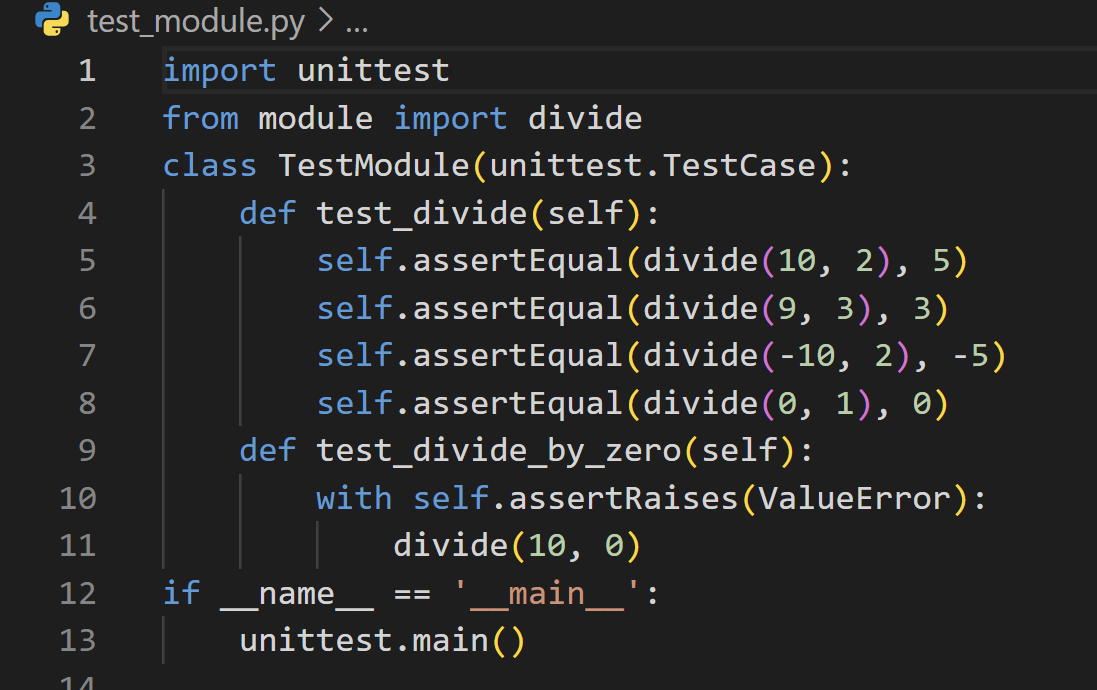
Задание 3: Интеграционная отладка

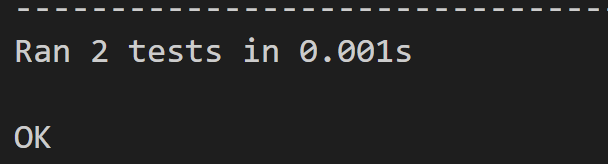
Провели тестирование на корректную работу модуля после интеграции с другими частями проекта, убедились в отсутствии конфликтов с другими модулями и что модуль не конфликтует и правильно выполняет функции.

Задание 4: Отладка производительности и использования памяти

Используя инструменты анализа производительности, проанализировали использование памяти и время выполнения модуля. Протестировали модуль, убедились, что все в порядке и он не подлежит оптимизации.

Задание 5: Автоматизированное тестирование и отладка

1. Разработали автоматизированные тесты для отладки выбранного модуля при помощи библиотеки unittest.

2. Используя тесты выяснили, что ошибок в модуле нет.

Задание 6: Отладка проблем с параллелизмом

1. В проекте не используется многопоточность или асинхронное программирование, следовательно, отладка на предмет ошибок с параллелизмом не требуется.

2. Среди инструментов многопоточных приложений выделяются дебаггеры, такие как GDB, которые позволяют отслеживать состояние всех потоков и устанавливать условия остановки на брейкпоинтах.

Потоковые профилировщики, например Thread Profiler в PyCharm и ThreadSanitizer в Clang, помогают анализировать действия каждого потока и выявлять гонки данных.

Статический анализ с помощью инструментов, таких как PVS-Studio или SonarQube, позволяет обнаружить потенциальные проблемы на этапе написания кода.

Методы отладки включают добавление логирования, что помогает отслеживать состояние приложения и выявлять проблемы, раздельную отладку, при которой каждый поток запускается на отдельном процессоре для облегчения контроля, и рефакторинг кода, который подразумевает разделение общих данных на отдельные переменные, предотвращая конфликты между потоками.

**Выводы работы**

В ходе выполнения практической работы мы изучили процесс отладки отдельных модулей программного проекта, выявили типичные ошибки и научились эффективно использовать инструменты отладки для их устранения.