**Отчет по самостоятельной работе №5**

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

Выполнил: студент

группы 319

Визнер Дарья Александровна

Дата 02.12.2024

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

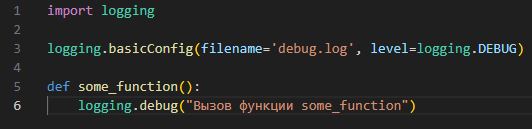
# Основная структура задания:

Задание №1: Подготовка среды для отладки

1. Настроила среду разработки для работы с проектом, инструментом отладки стал - Visual Studio code.

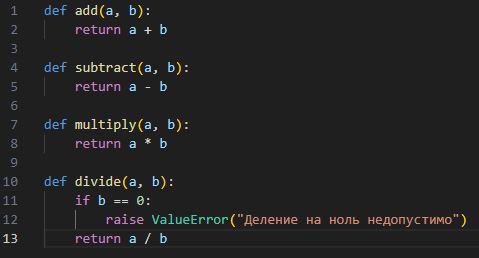
2. Убедилась, что отладчик корректно подключен к проекту, запустив сам отладчик через кнопку "Debug".

3. Использовала библиотеку Python logging для логирования ошибок и событий в проекте.



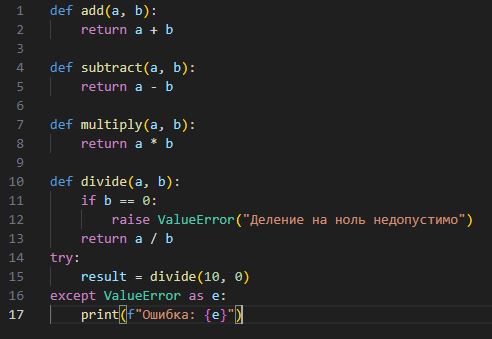
Задание №2: Отладка логики модуля

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



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

3. Выявила ошибку — деление на нуль.



4. Документирование ошибок и способы их исправления:

В Python деление на нуль вызывает исключение ZeroDivisionError. Для исправления этой ошибки можно использовать следующие методы:

* Обработка исключений с помощью try и except.
* Проверка перед делением.
* Использование функции, возвращающей None или другое значение при делении на ноль.
* Логирование ошибки.

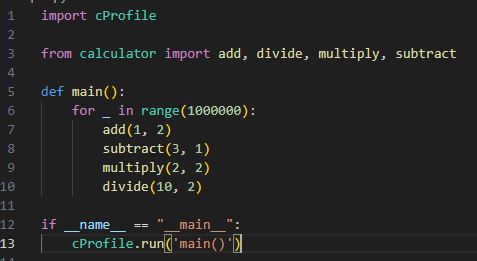
Задание №3: Интеграционная отладка

Тестирование после интеграции:

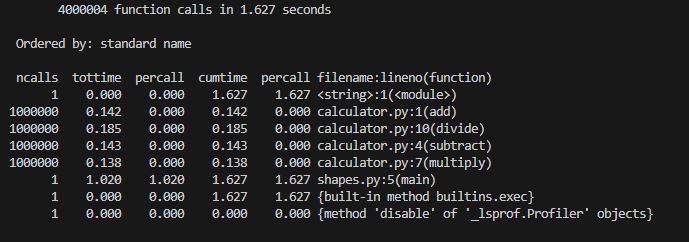
1. Провела тестирование на корректную работу модуля после интеграции с другими частями проекта.
2. Используя отладчик, проследила работу модуля в рамках всего проекта и убедилась в отсутствии конфликтов с другими модулями.
3. Выяснила, что модуль выполняет свои функции верно, исправление не требуется.

Задание №4: Отладка производительности и использования памяти

1. Использовала cProfile для анализа временных затрат и производительности кода.

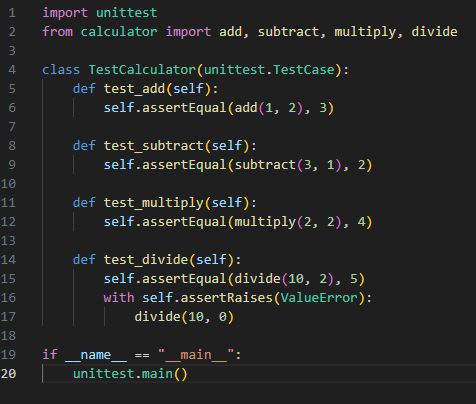


2. В ходе проведения тестов выяснила, что модуль не требует оптимизации, так как он эффективно использует память для своих расчетов и операций.



Задание №5: Автоматизированное тестирование и отладка

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



2. Исправлять ничего не нужно, так как ошибок в модуле не наблюдается.



Задание №6: Отладка проблем с параллелизмом

# В проекте не применяется многопоточность или асинхронное программирование, поэтому отладка на предмет ошибок с параллелизмом не требуется.

# Для решения проблем, связанных с параллелизмом, можно использовать следующие инструменты и методы:

# Использование механизмов блокировок (Lock, RLock, Semaphore и др.) для управления доступом к общим ресурсам.

# Применение инструментов профилирования и отладки, таких как py-spy и objgraph, для анализа и устранения проблем с параллелизмом.

# Использование библиотек, таких как concurrent.futures, которая упрощает работу с потоками и процессами, предоставляя интерфейсы ThreadPoolExecutor и ProcessPoolExecutor для асинхронного выполнения задач, а также joblib, которая подходит для параллельного выполнения задач и работы с массивами данных, что особенно полезно в научных расчетах и машинном обучении.

# Вывод: в ходе этой работы, я изучила процесс отладки отдельных модулей программного проекта, научилась выявлять типичные ошибки и научиться эффективно использовать инструменты отладки для их устранения.