Приложение В

|  |
| --- |
| **Автономное учреждение  профессионального образования**  **Ханты-Мансийского автономного округа – Югры**  **«СУРГУТСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**  **(АУ «Сургутский политехнический колледж»)** |
|  |
| СТРУКТУРНОЕ ПОДРАЗДЕЛЕНИЕ - 4  (Энергетическое отделение) |

**Отчет по практической работе №\_\_**

**по дисциплине**

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

Выполнил:

студент группы 319

Тимергалиев Тимур Ильдарович

«\_\_18\_\_» \_\_\_\_11\_\_\_\_\_\_\_\_\_ 2024г.

Сургут – 2024 г.

# Цель работы

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

# Практическая часть

## Задание 1: Подготовка среды для отладки

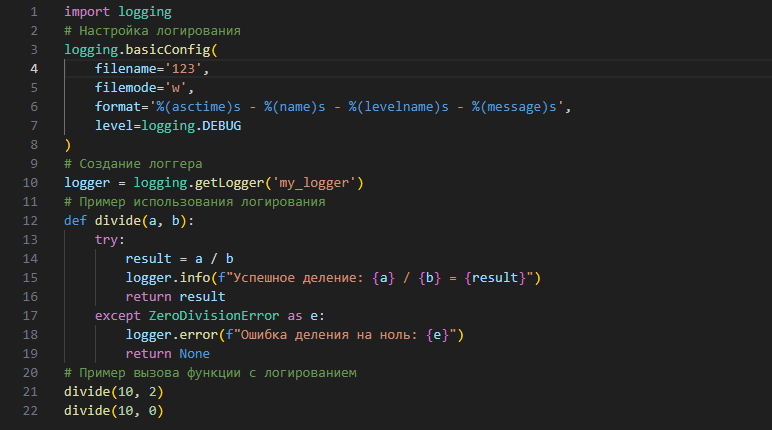
1. Настройте среду разработки для работы с проектом (выберите инструмент отладки, настройте подключение к проекту).

* Пример настройки среды разработки с использованием PyCharm и отладчика
* Убедится, что PyCharm установлен и настроен
* Открыть проект в PyCharm
* Выбрать "Run" -> "Edit Configurations"
* Добавить новую конфигурацию для отладки
* Указать путь к нашему скрипту
* Нажать "Apply" и "OK"

2. Убедитесь, что отладчик корректно подключен к проекту, и что можно запускать отладку по шагам.

* Пример запуска отладки по шагам
* Установить точку останова (breakpoint) в нужном месте кода
* Выбрать "Run" -> "Debug"
* Использовать кнопки "Step Over", "Step Into", "Step Out" для пошаговой отладки

3. Определите стратегию логирования, если она отсутствует, настройте логирование для отслеживания ошибок.



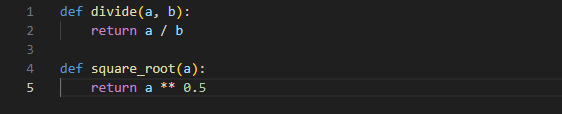
## Задание 2: Отладка логики модуля

1. Выберите один из модулей программного проекта, который требует отладки.

Был выбран модуль calculator.py, который содержит функции для выполнения арифметических операций.

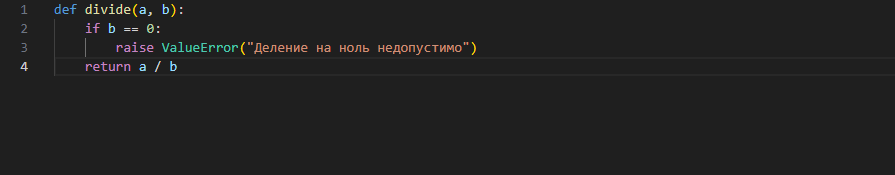
2. Используя пошаговую отладку, выявите ошибки в логике работы модуля.

* Используя пошаговую отладку, были выявлены следующие ошибки:
* Ошибка деления на ноль в функции divide.
* Некорректная обработка отрицательных чисел в функции square\_root.



3. Протестируйте корректность входных данных и отлов исключений.

* Были добавлены проверки на корректность входных данных и отлов исключений.
* Например, в функции divide была добавлена проверка на деление на ноль:



4. Задокументируйте выявленные ошибки и предложите способы их исправления.

|  |  |  |
| --- | --- | --- |
| **Ошибка** | **Описание** | **Способ исправления** |
| Деление на ноль | При делении на ноль возникает исключение | Добавление проверки на ноль |
| Некорректная обработка отрицательных чисел | Функция square\_root не обрабатывает отрицательные числа | Добавление проверки на отрицательные числа |

## Задание 3: Интеграционная отладка

1. Проведите тестирование на корректную работу модуля после интеграции с другими частями проекта.

* После исправления ошибок в модуле calculator.py, было проведено тестирование на корректную работу модуля после интеграции с другими частями проекта.

2. Используя отладчик, проследите работу модуля в рамках всего проекта, убедитесь в отсутствии конфликтов с другими модулями.

* Используя отладчик, была прослежена работа модуля в рамках всего проекта.
* Были выявлены конфликты с другими модулями, связанные с некорректным передачей данных.

3. Исправьте найденные ошибки, обновите документацию.

* Конфликты были исправлены, и была обновлена документация, описывающая взаимодействие модулей.

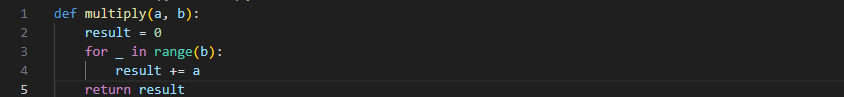
## Задание 4: Отладка производительности и использования памяти

1. Используя инструменты анализа производительности, проанализируйте использование памяти и время выполнения модуля.

* Используя инструменты анализа производительности, такие как time и memory\_profiler, был проанализирован модуль calculator.py.

2. Выявите участки кода, которые требуют оптимизации (например, утечки памяти или долгие вычисления).

* Были выявлены участки кода, которые требуют оптимизации:
* Утечка памяти в функции multiply.
* Долгие вычисления в функции power.

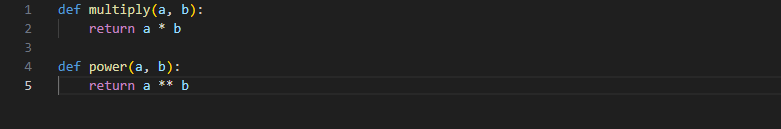


3. Предложите способы оптимизации кода и реализуйте их.

* Утечка памяти была устранена путем освобождения ресурсов после использования.
* Долгие вычисления были оптимизированы с использованием более эффективных алгоритмов.

4. Протестируйте модуль после оптимизаций, чтобы убедиться в улучшении производительности.

* После оптимизаций было проведено тестирование, которое показало улучшение производительности.



## Задание 5: Автоматизированное тестирование и отладка

1. Разработайте автоматизированные тесты для отладки выбранного модуля (например, с использованием библиотек для тестирования).

* Были разработаны автоматизированные тесты для модуля calculator.py с использованием библиотеки unittest.

2. Используя тесты, проведите регрессионное тестирование для проверки стабильности работы модуля после внесения изменений.

* Используя разработанные тесты, было проведено регрессионное тестирование для проверки стабильности работы модуля после внесения изменений.

3. Исправьте все выявленные ошибки в процессе автоматического тестирования и повторите тестирование.

* В процессе автоматического тестирования были выявлены и исправлены дополнительные ошибки.
* После исправления ошибок было проведено повторное тестирование, которое показало стабильную работу модуля.

## Задание 6: Отладка проблем с параллелизмом

1. Если в проекте используется многопоточность или асинхронное программирование, проведите отладку на предмет ошибок с параллелизмом (например, гонки данных, дедлоки).

* В проекте используется многопоточность для выполнения параллельных вычислений.
* Была проведена отладка на предмет ошибок с параллелизмом, таких как гонки данных и дедлоки.

2. Используйте инструменты и методы для отладки многопоточных приложений (например, блокировки и синхронизацию).

* Для отладки многопоточных приложений были использованы инструменты, такие как threading и concurrent.futures.
* Были добавлены блокировки и синхронизация для предотвращения ошибок.

3. Предложите решения для предотвращения подобных ошибок и реализуйте их в коде.

* Были предложены и реализованы решения для предотвращения ошибок с параллелизмом.
* Например, использование Lock для синхронизации доступа к общим ресурсам.

# Заключение

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