**Отчет по практической работе № 5**

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

Выполнили: студенты

Группы 319

Биглова Яна Рамилевна

Дата 29.11.2024

# 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: выбрала VSCode в качестве инструмента отладки, открыла проект. VSCode имеет встроенный отладчик, поэтому было достаточно открыть проект (представлен на рисунке 1):



Рисунок 1 – Изначальный проект

1. Убедилась, что отладчик корректно подключен к проекту, и что можно запускать отладку по шагам. Для запуска отладчика по шагам установила точку breakpoint и нажала F5. Запуск отладчика представлен на рисунке 2.

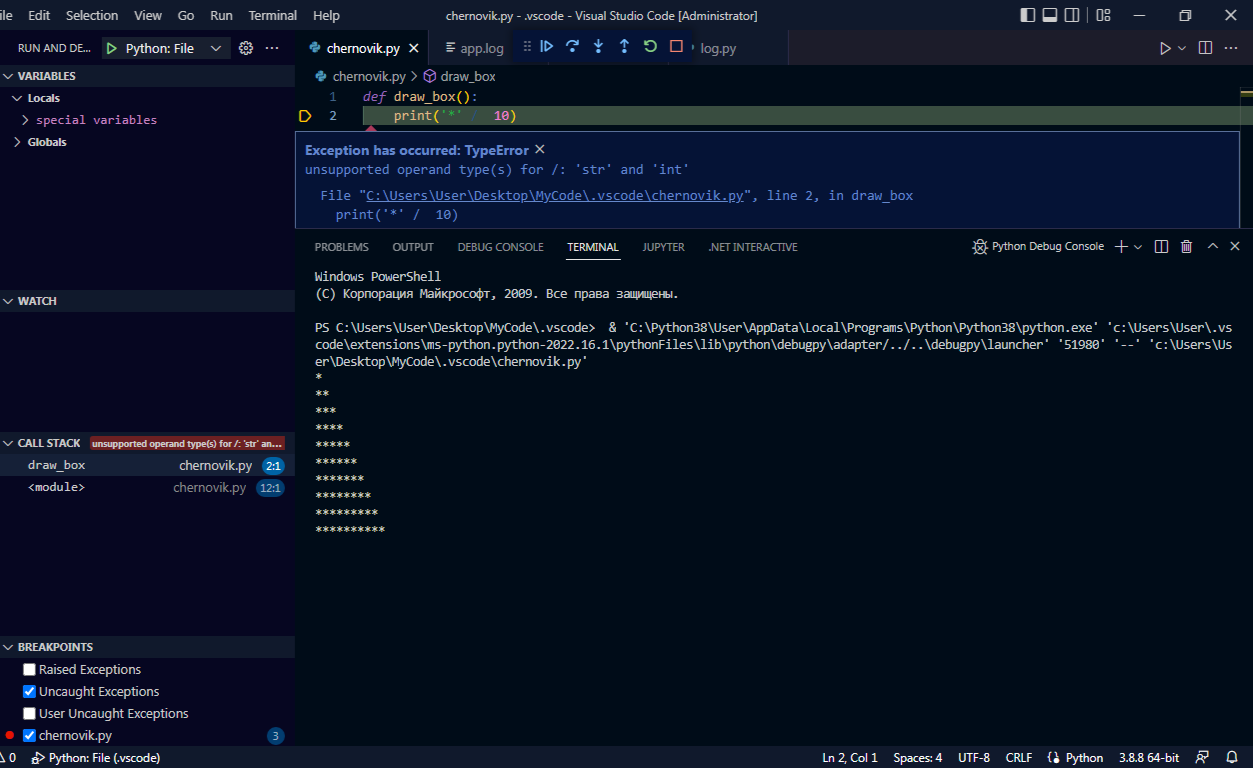


Рисунок 2 – Запуск отладчика

1. Настроила логирование для отслеживания ошибок при помощи модуля logging. Код с использованием библиотеки изображен на рисунке 3.

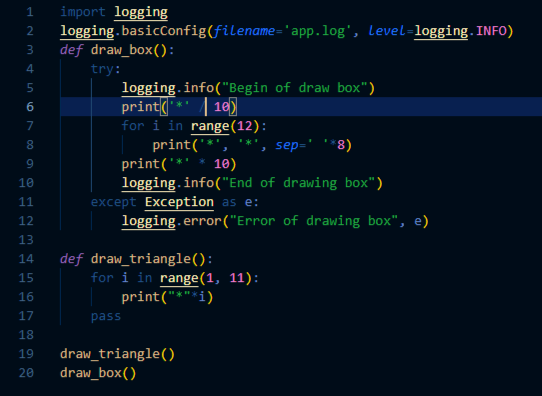


Рисунок 3 – Настройка логирования

Задание №2: Отладка логики модуля

1. Выбрала модуль «draw\_box» программного проекта, который требует отладки.
2. Используя пошаговую отладку, выявила ошибки в логике работы модуля: деление строки на число вместо умножения.
3. Протестировала корректность входных данных и отлов исключений, результат на рисунке 4.

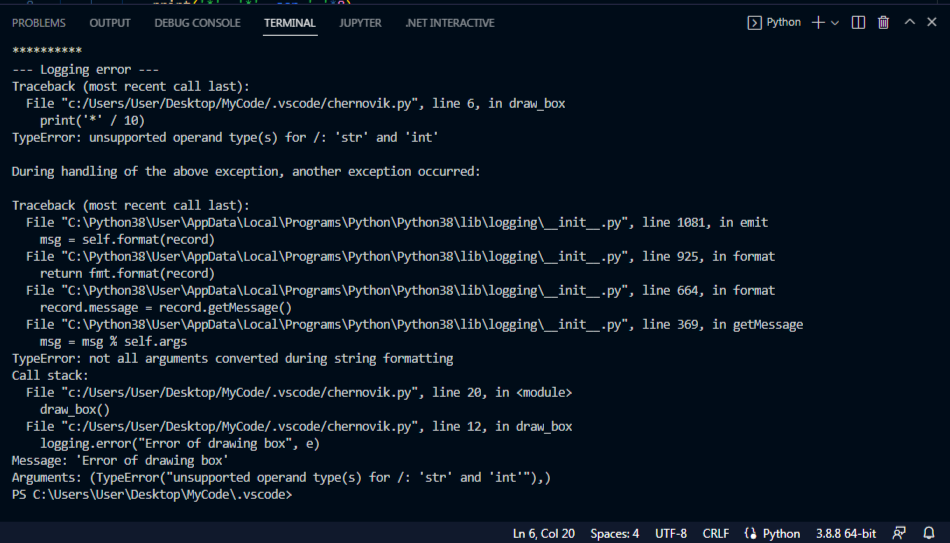


Рисунок 4 – Тест входных данных и отлов исключений

1. Задокументировала выявленную ошибку и предложила способ ее исправления. Документ с ошибкой и способом ее решения на рисунке 5.

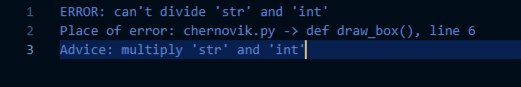


Рисунок 5 – Документ с ошибкой

Задание №3: Интеграционная отладка

1. Провела тестирование на корректную работу модуля после интеграции с другими частями проекта. Модуль работает корректно, не вызывая ошибок. Код и вывод показаны на рисунках 6 и 7.

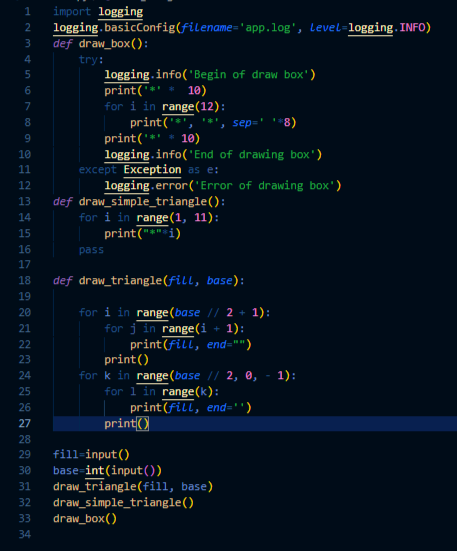


Рисунок 6 – Код с проверкой модуля после интеграции

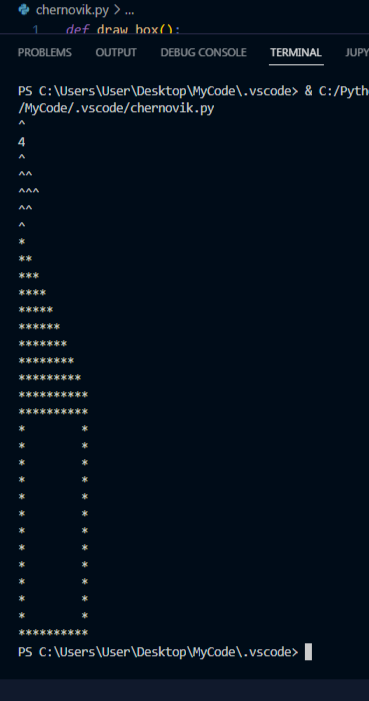


Рисунок 7 – Вывод программы после проверки

Задание №4: Отладка производительности и использования памяти

1. Используя инструменты анализа производительности: библиотеки – cProfile; memory\_profiler, проанализировала использование памяти и время выполнения модуля. Вычисление времени и памяти программы представлено на рисунке 8.

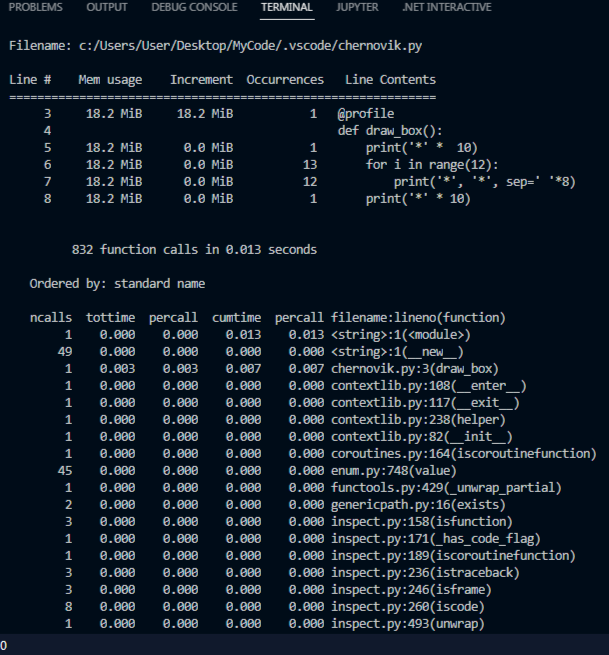


Рисунок 8 – Вывод времени вычисления и затраты памяти

1. Основываясь на предыдущем анализе, сделала вывод, что код не нуждается в оптимизации.

Задание №5: Автоматизированное тестирование и отладка

1. Разработала автоматизированные тесты для отладки выбранного модуля с использованием библиотеки unittest. Программа с автоматизированным тестированием изображена на рисунке 9.



Рисунок 9 – Программа с автоматизированным тестированием

1. Используя тесты, провела регрессионное тестирование для проверки стабильности работы модуля после внесения изменений. В результате тестирования модуль работал исправно. Работа тестирования представлена на рисунке 10.

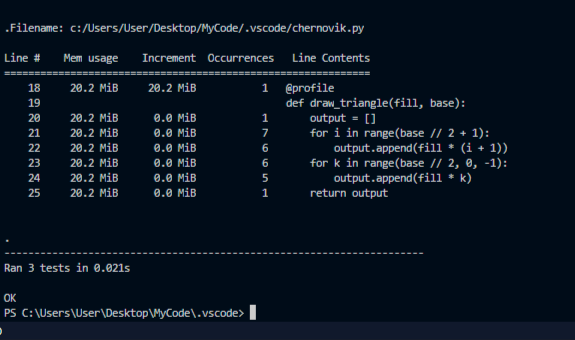


Рисунок 10 – Регрессивное тестирование

Задание №6: Отладка проблем с параллелизмом

1. В моем проекте не используется многопоточность или асинхронное программирование, а он представляет собой последовательный код, который выполняет функции одна за другой. Для проверки параллелизма можно использовать библиотеку threading, multiprocessing и asyncio.

**Выводы работы:**

В ходе выполнения работы изучила процесс отладки отдельных модулей программного проекта, выявила типичные ошибки и научилась эффективно использовать инструменты отладки для их устранения.