**Отчет по самостоятельной работе №6**

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

Выполнил: студент

группы 319

Юсупова Камола Акрамовна

Дата 26.11.2024

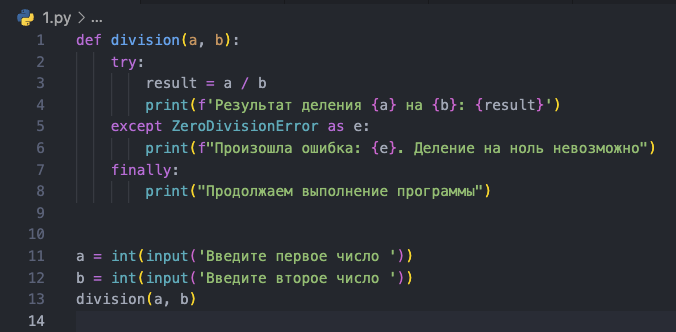
**Цель работы**

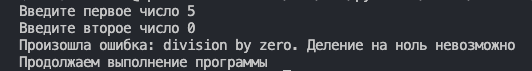
Цель данной работы – изучить методы и подходы к обработке исключений, научиться применять конструкции try-catch, обрабатывать и регистрировать исключения для повышения стабильности программного обеспечения.

**Основная структура задания**

**Задание 1: Базовая обработка исключений**

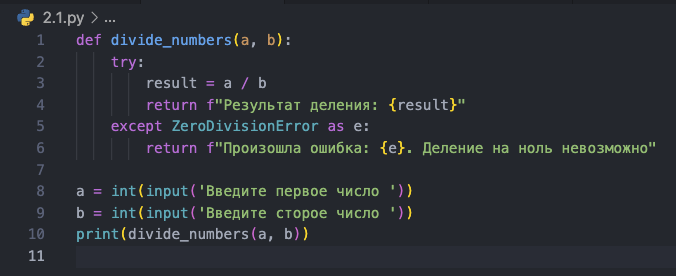
1. Создала простой программный модуль, в котором возникает ошибка (например, деление на ноль).
2. Реализовала обработку исключения с помощью конструкции try-catch, чтобы предотвратить аварийное завершение программы.
3. Выведите сообщение об ошибке и продолжите выполнение программы.



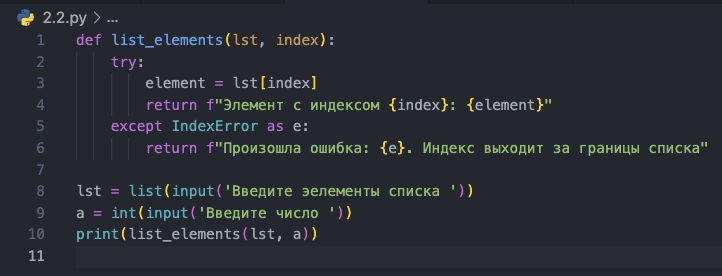


**Задание 2: Иерархия исключений**

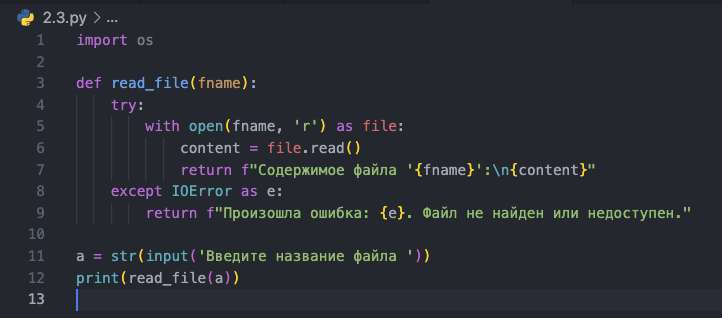
1. Изучила, как работают различные типы исключений в вашем языке программирования.
2. Создала несколько примеров обработки различных исключений (например, деление на ноль, выход за границы массива, ошибки ввода-вывода).













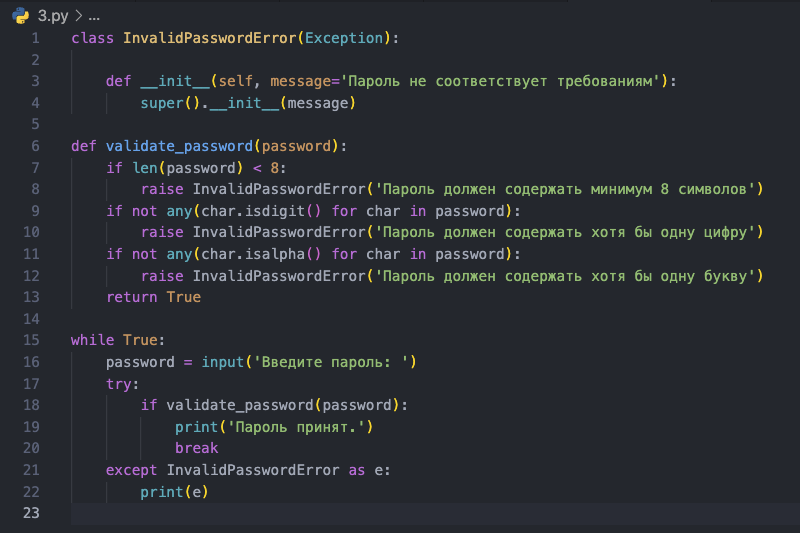
1. Задокументировала различия между ними и предложила стратегию обработки каждой категории.

Различные исключения возникают в ответ на разные ситуации, поэтому подход к их обработке может варьироваться в зависимости от контекста.

* Деление на ноль: сообщает пользователю о невозможности операции и предлагает ввести другие данные. Пример стратегии: перехватывает исключение и выводит понятное сообщение об ошибке, предлагая повторить ввод.
* Выход за границы массива: проверяет индексы перед доступом к массивам или обрабатывает исключение, если оно всё же возникло. Пример стратегии: предупреждает пользователя о неверном индексе и предлагает выбрать другой вариант.
* Ошибки ввода-вывода: проверяет существование и доступность файлов до попытки чтения/записи. Пример стратегии: сообщает пользователю, что файл не найден, и предлагает проверить правильность пути или наличие файла.

**Задание 3: Генерация пользовательских исключений**

1. Создала пользовательский класс исключений для обработки специфических ситуаций в вашем программном проекте.
2. Написала пример кода, который генерирует и обрабатывает это исключение.

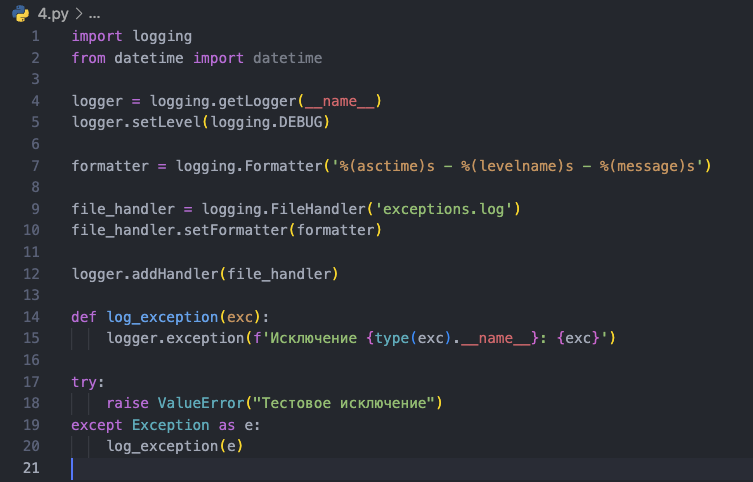


1. Обосновала необходимость создания пользовательского исключения.

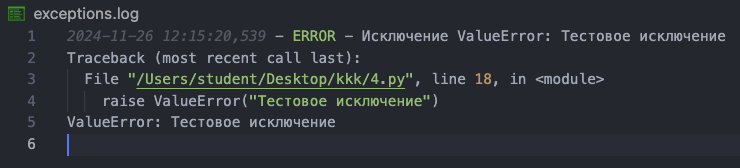
* Пользовательское исключение делает код более читаемым и самодокументированным.
* С помощью пользовательских исключений можно предоставлять более точные и информативные сообщения об ошибках, что облегчает поиск и устранение неисправностей.
* Вы можете точно контролировать, какие исключения должны быть пойманы и как они будут обработаны.
* Пользовательские исключения легко расширять и адаптировать под новые требования проекта.

**Задание 4: Логирование исключений**

1. Реализовала механизм логирования исключений в файл или в базу данных для их дальнейшего анализа.



1. Проверила корректность логирования и убедилась, что каждая ошибка записывается с указанием времени, типа ошибки и контекста.

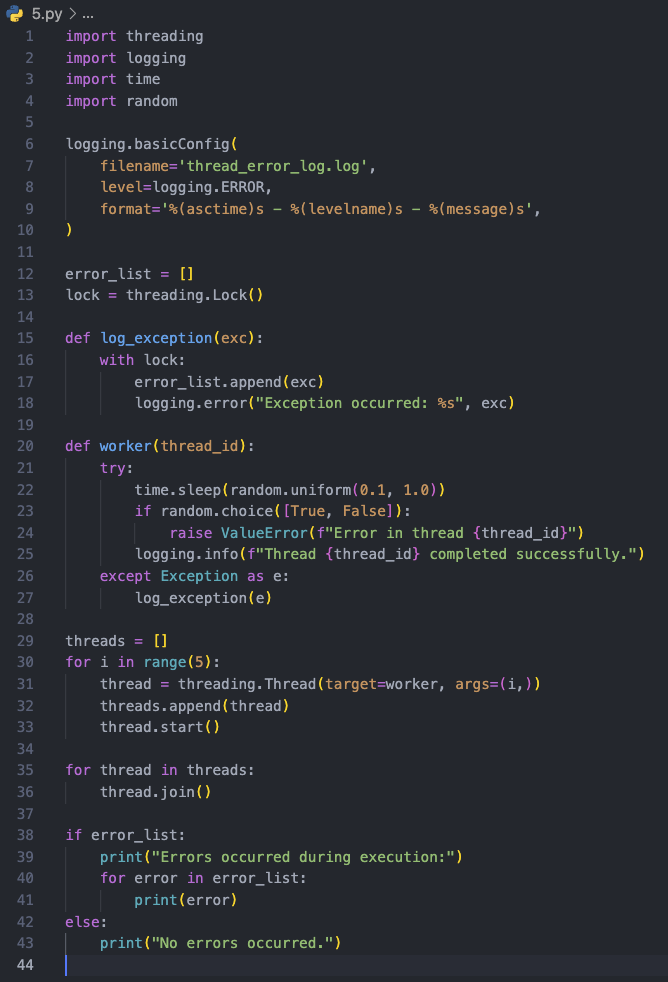


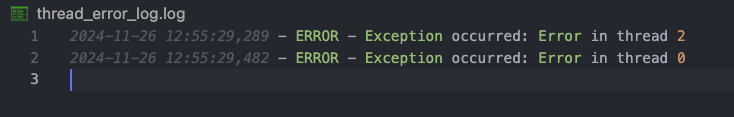
1. Задокументировала подходы к анализу и устранению исключений на основе логов.

* Подходы к анализу логов:
* Регулярный мониторинг логов: регулярно просматривайте логи, чтобы своевременно выявлять возникающие проблемы.
* Фильтрация по уровню серьезности: используйте фильтры для выделения ошибок определенного уровня, чтобы сосредоточиться на наиболее критичных проблемах.
* Анализ повторяющихся ошибок: обратите внимание на часто встречающиеся ошибки. Они могут указывать на системную проблему, которую необходимо устранить.
* Контекстный анализ: Изучайте контекст возникновения ошибок.
* Подходы к устранению исключений:
* Идентификация причины: определите причину возникновения исключения.
* Исправление кода: внесите изменения в код, чтобы предотвратить возникновение исключения.
* Добавление тестов: Напишите тесты, которые будут проверять исправленную функциональность и предотвращать повторное появление ошибки.
* Документирование решений: Зафиксируйте решения и внесенные изменения в документации проекта, чтобы другие разработчики могли ознакомиться с ними.

**Задание 5: Обработка исключений в многопоточности**

1. Реализовала обработку исключений в многопоточной среде.
2. Убедилась, что исключения в одном потоке не приводят к завершению программы или других потоков.
3. Используйте синхронизацию для корректной работы с исключениями в многопоточных приложениях.





**Выводы работы**

Изучила методы и подходы к обработке исключений, научилась применять конструкции try-catch, обрабатывать и регистрировать исключения для повышения стабильности программного обеспечения.