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

**Отчет по самостоятельной работе №6**

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

Выполнил: студент

группы 319

Климентов Иван Вячеславович

Дата 28.11.2024

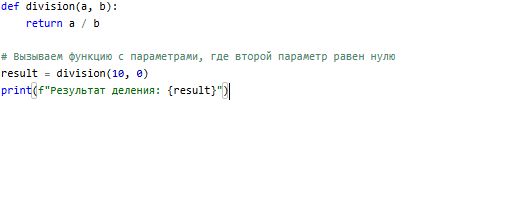
# Цель работы:

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

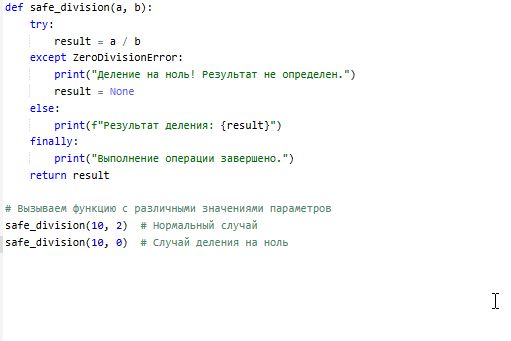
# Основная структура задания:

Задание №1: Базовая обработка исключений

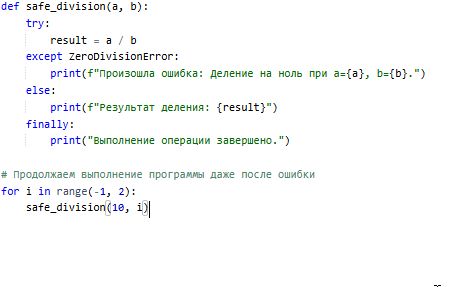
1. Был создан простой программный модуль, в котором возникает ошибка деления на нуль.



1. Была реализована обработка исключения с помощью конструкции try-except, с целью предотвращения аварийного завершения программы.

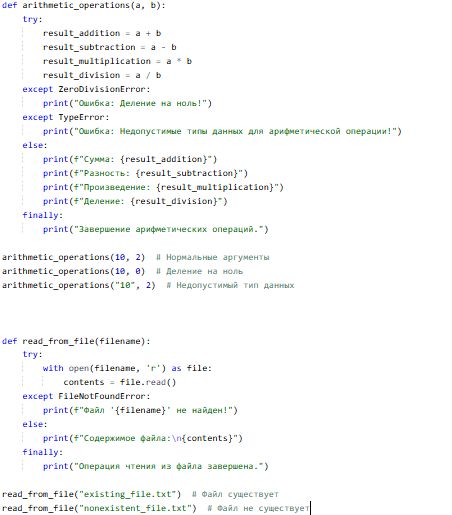


1. Был реализован вывод сообщения об ошибке, а также продолжение выполнения программы.



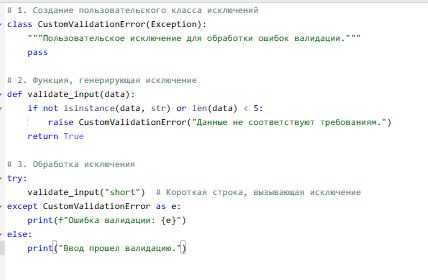
Задание №2: Иерархия исключений

1. После изучения различных типов исключений были выделены одни из самых распространённых.
   1. ZeroDivisionError (деление на ноль)
   2. IndexError (выход за границы массива)
   3. IOError (ошибки ввода-вывода)

2.2)Были созданы несколько примеров обработки различных исключений

Задание №3: Генерация пользовательских исключений

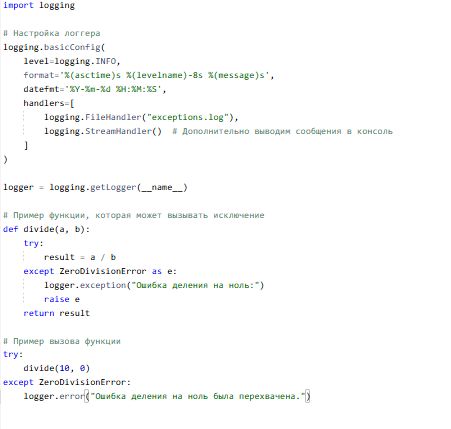
1. Был создан пользовательский класс исключений для обработки специфических ситуаций в программном проекте, а также генерацию и обработку этого исключения



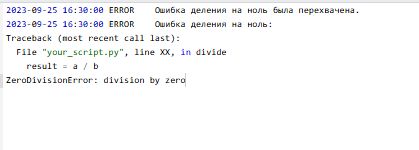
1. Обоснование необходимости создания пользовательского исключения.
   1. Пользовательские исключения позволяют более точно описывать специфические ошибки, возникающие в программе.
   2. Они упрощают отладку и обработку ошибок, так как позволяют легко идентифицировать их тип.

Задание №4: Логирование исключений

1. Был реализован механизм логирования исключений в файл для их дальнейшего анализа.



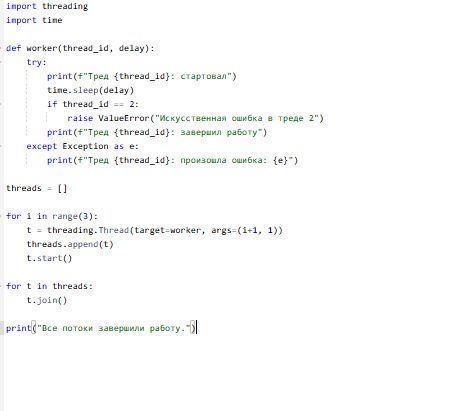
1. При проверке выяснилось, что все ошибки записываются с указанием времени, типа и контекста ошибки.



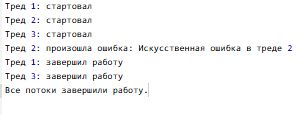
1. Документирование подходов к анализу и устранению исключений:
   1. Анализ логов: Проанализируй логи, чтобы определить частоту возникновения ошибок и их типы.
   2. Устранение исключений: Используй информацию из логов для устранения причин ошибок в коде.

Задание №5: Обработка исключений в многопоточности

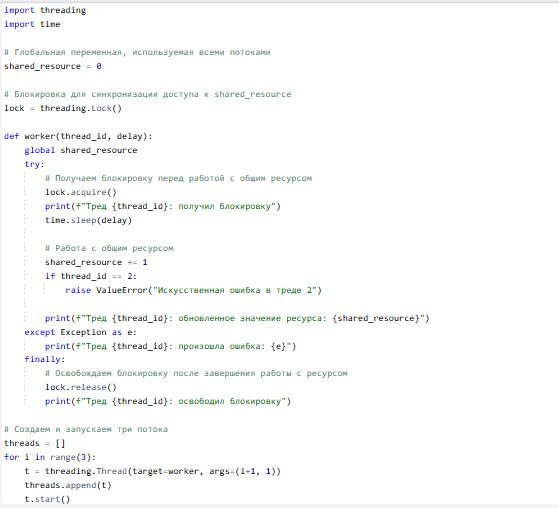
1. Была реализована обработка исключений в многопоточной среде.



1. После запуска программы стало ясно, что исключения в одном потоке не приводят к завершению всей программы.



1. Была использована синхронизация для корректной работы с исключениями в многопоточных приложениях.



**Вывод:**

Во время выполнения практической работы были исследованы и использованы методы обработки исключений в программном коде. Рассмотрены основные конструкции try-except, иерархия исключений, а также создание собственных исключений. Также был внедрен механизм логирования исключений для их последующей диагностики. В условиях многопоточного окружения обеспечено правильное управление исключениями посредством применения механизмов синхронизации.