**Отчет по самостоятельной работе №**

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

Выполнил: студент

группы 319

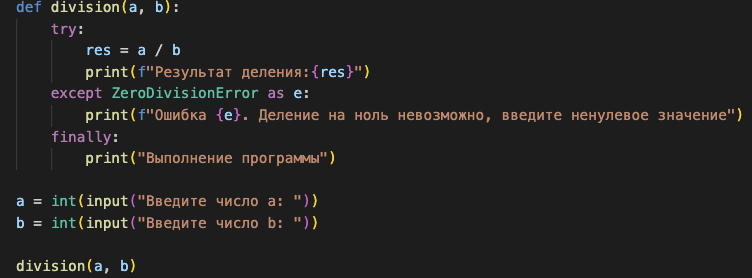
Губайдуллина Камилла Рустемовна

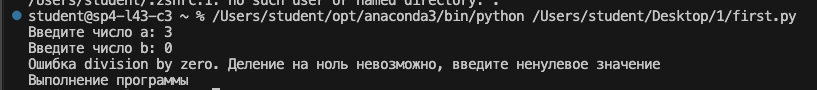
30.11.2024

# 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:

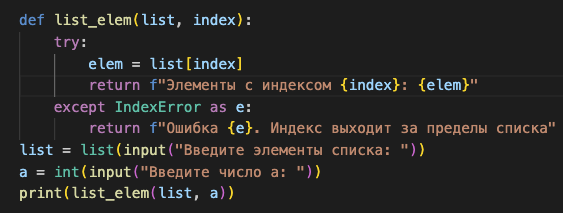
Задание 1: Базовая обработка исключений

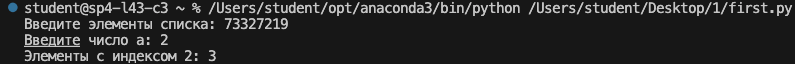
Создала простой программный модуль, в котором возникает ошибка “деление на ноль”, затем реализовала обработку исключения с помощью конструкции try-catch, чтобы предотвратить аварийное завершение программы. И в конце вывела сообщение об ошибке и продолжила выполнение программы.

Рисунок 1 – программный модуль, в котором возникает ошибка

## Задание 2: Иерархия исключений

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





1. Задокументировала различия между ними и предложите стратегию обработки каждой категории:

ZeroDivisionError: возникает при попытке деления на ноль.

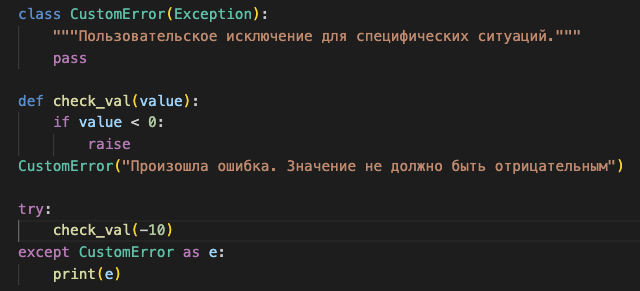
Стратегия обработки: уведомить пользователя о некорректном делении, предлагая повторить ввод данных.

IndexError: возникает при доступе к элементу массива по несуществующему индексу.

Стратегия обработки: информировать о выходе за пределы массива, предупреждать пользователя о неверном индексе, предлагая выбрать другой вариант.

## 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.

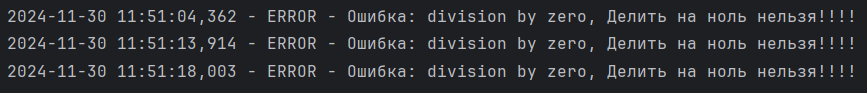


1. Пользовательские исключения полезны тем, что их можно вызвать с неправильными или неожиданными входными данными, тем самым лучше прояснив ситуацию с кодом, который падает или неправильно работает. Они позволяют вам определять собственные условия возникновения ошибок и обрабатывать их более конкретным и осмысленным образом.

## Задание 4: Логирование исключений

1. Реализовала механизм логирования исключений в файл для их дальнейшего анализа.
2. Проверила корректность логирования и убедитесь, что каждая ошибка записывается с указанием времени, типа ошибки и контекста.



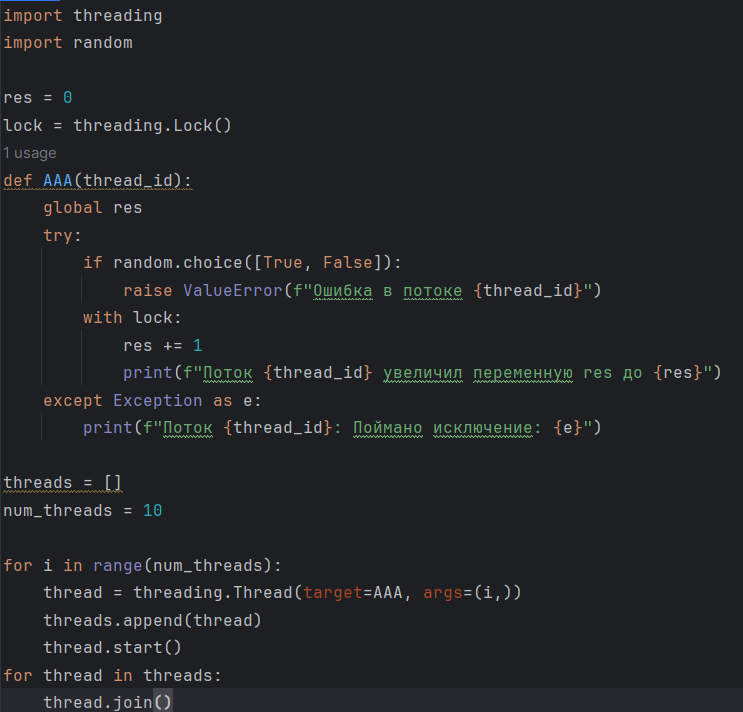
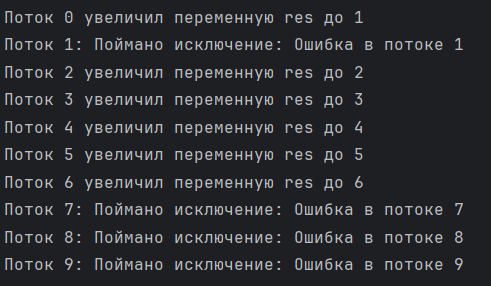


1. Задокументировала подходы к анализу и устранению исключений на основе логов.

* Использование нескольких уровней логирования. Это позволяет писать много информации постоянно только при необходимости отладки.
* Трассировка кода. В начало каждой процедуры и функции вставляется вызов логирования, который запоминает название этой процедуры и входящие параметры.
* Отладка через логирование. Программа запускается, и по логам можно посмотреть, что в них выписано. Если какой-то информации не хватает, её добавляют и решают проблемы, изучая эти данные.
* Запись объекта исключения. В лог записывают как сообщение разработчика, так и весь объект исключения. Это даёт полный вид исключения, включая вызванные методы и трассировку стека.
* Запись типа исключения в лог. Указание в логе типа исключения помогает определить характер проблем.

## Задание 5: Обработка исключений в многопоточности

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



# Выводы работы: в ходе практической работы я изучила механизмы обработки исключений в Python, включая создание пользовательских исключений и логирование ошибок. Я освоила навыки обработки исключений в многопоточной среде, научилась использовать блоки try-except для обработки ошибок в каждом потоке, также реализовала примеры, включающие в себя различные исключение и применила синхронизацию для предотвращения конфликтов.