**Отчет по самостоятельной работе №**6

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

Выполнил: студент

Группы 319

Суворова Анастасия Викторовна

Дата\_\_.\_\_.\_\_\_\_

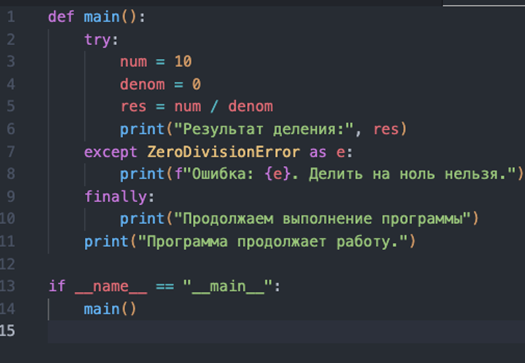
**Цель работы**

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**

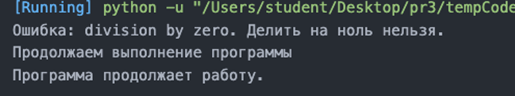
Задание 1: Базовая обработка исключений

1. Создайте простой программный модуль, в котором возникает ошибка (например, деление на ноль).



2. Реализуйте 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. Изучите, как работают различные типы исключений в вашем языке программирования (например, стандартные исключения и их наследники).

ZeroDivisionError — деление на ноль.

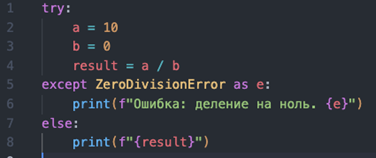
IndexError — выход за пределы индекса в списке или массиве.

ValueError — некорректное значение (например, попытка преобразовать строку в число, но строка не может быть интерпретирована как число).

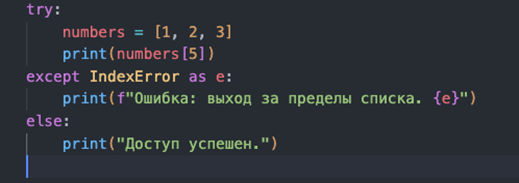
KeyError — попытка доступа к несуществующему ключу в словаре.

2. Создайте несколько примеров обработки различных исключений (например, деление на ноль, выход за границы массива, ошибки ввода-вывода).

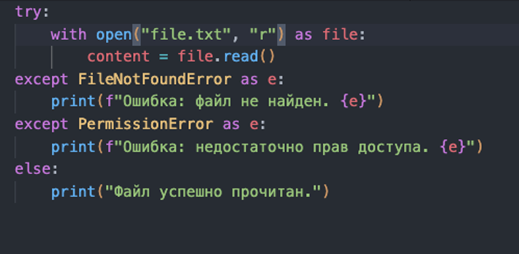
Деление на ноль



Выход за пределы массива



Ошибка ввода/вывода



3. Задокументируйте различия между ними и предложите стратегию обработки каждой категории.

1. Деление на ноль

Исключение должно быть поймано и обработано. Вместо аварийного завершения стоит вывести сообщение или предоставить пользователю альтернативные варианты.

2. Выход за пределы массива

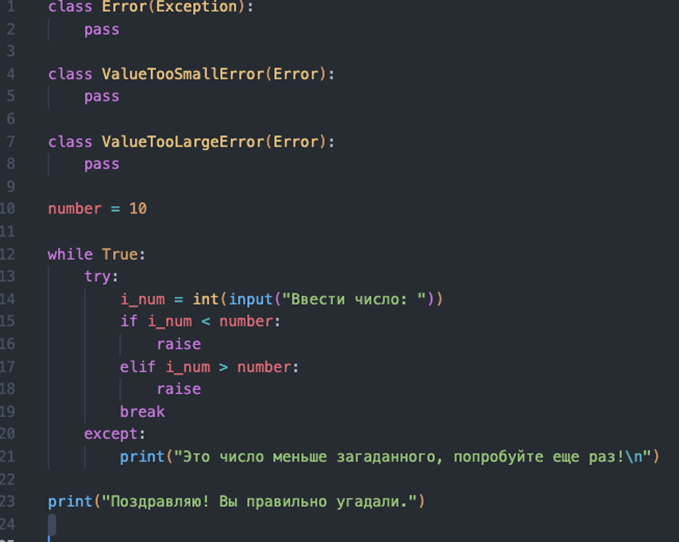
Ошибка может быть обработана путем проверки допустимости индекса перед доступом к элементу массива.

3. Ошибки ввода/вывода

Ошибки с файлами должны быть обработаны, чтобы избежать потери данных. В случае ошибок прав доступа можно предложить пользователю перезапустить программу с необходимыми правами или попросить изменить настройки доступа.

Задание 3: Генерация пользовательских исключений

1. Создайте пользовательский класс исключений для обработки специфических ситуаций в вашем программном проекте.



2. Напишите пример кода, который генерирует и обрабатывает это исключение.

3. Обоснуйте необходимость создания пользовательского исключения.

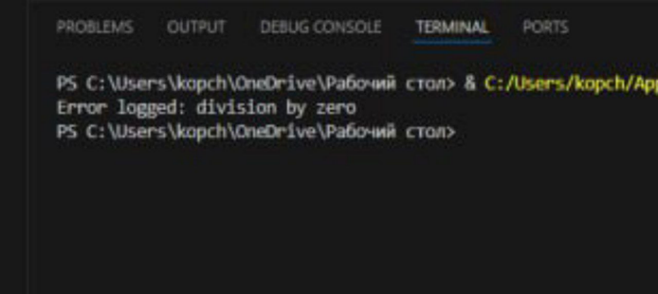
Пользовательские исключения полезны тем, что их можно вызвать с неправильными или неожиданными входными данными, тем самым лучше прояснив ситуацию с кодом, который падает или неправильно работает.

Задание 4: Логирование исключений

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



2. Проверьте корректность логирования и убедитесь, что каждая ошибка записывается с указанием времени, типа ошибки и контекста.



3. Задокументируйте подходы к анализу и устранению исключений на основе логов.

**Выводы работы**

В ходе выполнения практической работы я изучила методы и подходы к обработке исключений, научилась применять конструкции try-catch, обрабатывать и регистрировать исключения для повышения стабильности программного обеспечения.