**Отчет по практической работе №6**

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

Выполнил: студент

группы 319

Яковлева Анастасия Сергеевна

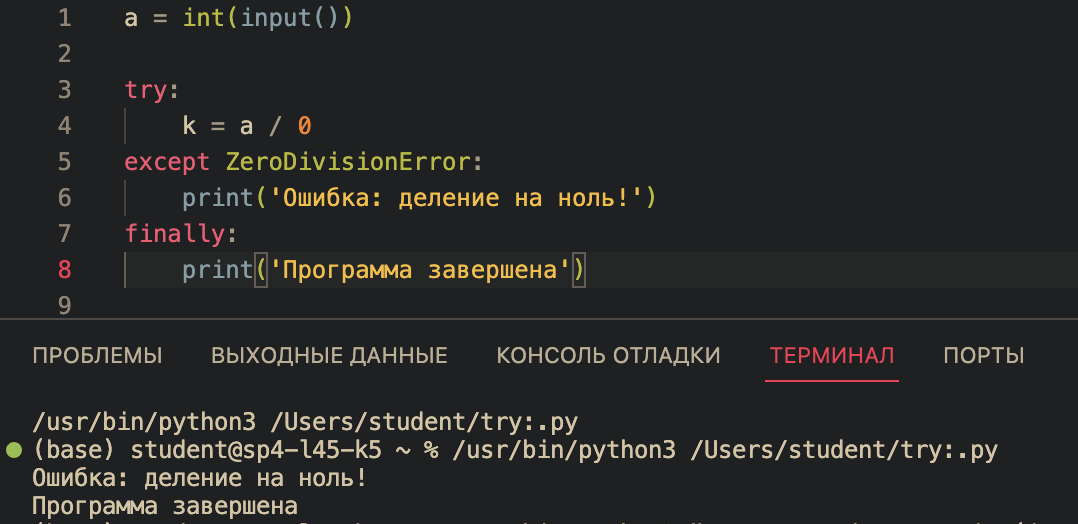
29.11.2024

**Цель работы:**

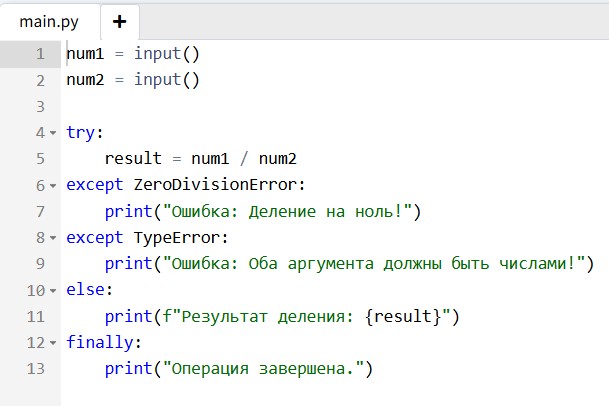
Цель данной работы – изучить методы и подходы к обработке исключений, научиться применять конструкции try-catch, обрабатывать и регистрировать исключения для повышения стабильности программного обеспечения.

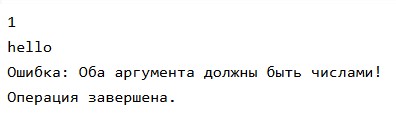
Задание 1: Базовая обработка исключений

1. Создала простой программный модуль, в котором возникает ошибка ZeroDivisionError - деление на ноль.

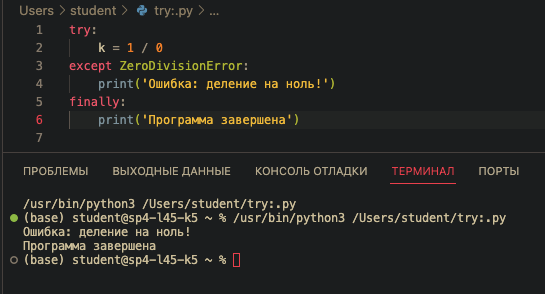


2. Реализовала обработку исключения с помощью конструкции try-catch, чтобы предотвратить аварийное завершение программы.





3. Вывела сообщение об ошибке и продолжила выполнение программы с помощью finally .

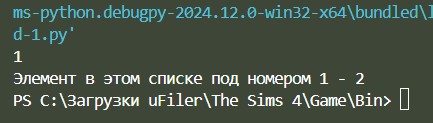
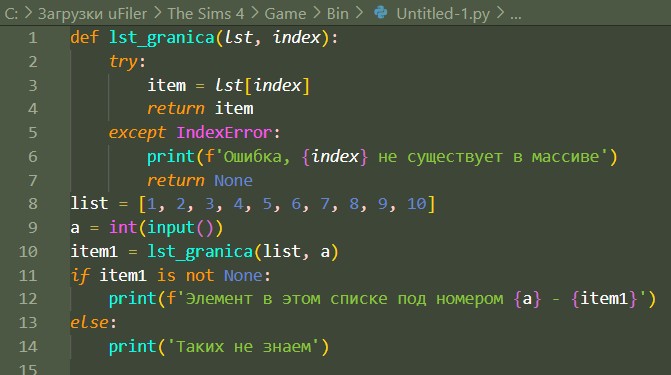


Задание 2: Иерархия исключений

1. Изучила, как работают различные типы исключений в языке программирования python (например, стандартные исключения и их наследники).

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

Выход за границы массива:



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

1. Деление на ноль: это исключение возникает, когда происходит попытка деления числа на ноль. Это может быть результатом неправильного ввода данных или логической ошибки в коде.
2. Выход за границы массива: это исключение возникает, когда программа пытается получить доступ к элементу списка (или другого индексируемого объекта) по индексу, который выходит за пределы допустимого диапазона.
3. Ошибка ввода-вывода: эти исключения возникают при ошибках ввода-вывода, например, при попытке открыть файл, который не существует, или при проблемах с доступом к файлу.

Стратегии обработки:

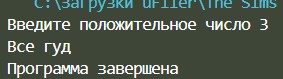
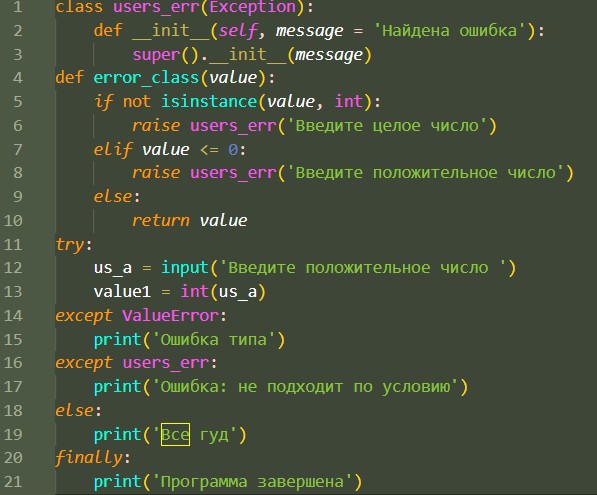
1. Деление на ноль: использовать конструкцию try/except для перехвата ZeroDivisionError, вывести пользователю понятное сообщение об ошибке и проверить значение перед выполнением деления, чтобы избежать этой ошибки.
2. Выход за границы массива: использовать конструкцию try/except для перехвата IndexError, вывести пользователю сообщение об ошибке и проверить значение перед выполнением вывода элемента массива, чтобы избежать этой ошибки.
3. Ошибка ввода-вывода: использовать конструкцию try/except для перехвата IOError или OSError, вывести пользователю сообщение об ошибке и проверить значение перед выполнением вывода-ввода данных и проверить.

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.



3. Необходимость создания пользовательского исключения заключается в том, что таким образом можно более точно и эффективно обрабатывать специфические ошибки, которые могут не попадать под условия стандартных исключений. Это влияет на удобство и понятность кода, ведь разработчики, работающие над проектом, смогут быстро понять, какие виды ошибок возможны в той или иной части программы.

Задание 4: Логирование исключений

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

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

3. Подходы к анализу и устранению исключений на основе логов:

1. Нужно найти и устранить конкретные ошибки в коде, которые привели к исключению. Это может включать проверку корректности входных данных, добавление обработчиков исключений и т.д.
2. Иногда исключение возникает из-за плохой архитектуры или недостаточной проверки условий. Рефакторинг кода может улучшить его структуру и предотвратить возникновение подобных ошибок в будущем.
3. Стоит написать тесты, которые проверяют работу кода в условиях, приводящих к исключениям. Это позволит убедиться, что проблема действительно решена и избежать регрессий.
4. Настроить систему мониторинга, которая будет отслеживать состояние приложения и уведомлять разработчиков о возникновении новых исключений. Это поможет оперативно реагировать на новые проблемы.

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. Использовала синхронизацию для корректной работы с исключениями в многопоточных приложениях.

**Вывод работы:**

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