**Отчёт по практической работе №6**

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

Выполнил: студент

группы 319

Абназыров Денис Ильгисович

Дата 20.11.2024

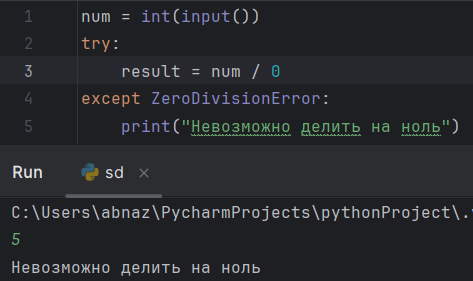
**Цель работы:**

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

Программный модуль в котором возникает ошибка. При возникновении ошибки выводит «Невозможно делить на ноль» и продолжает выполнять функцию.

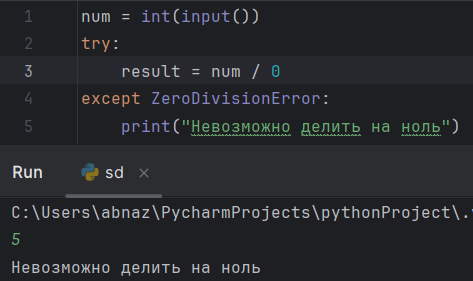


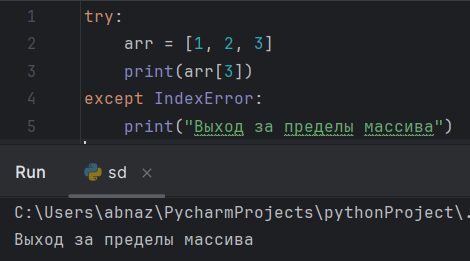
**Задание 2: Иерархия исключений**

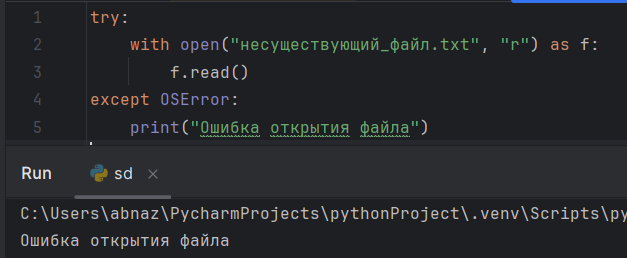
2.1 Типы исключений в Pythoon

В Python есть несколько стандартных исключений, которые наследуются от базового класса Exception. Наиболее распространенные из них:  
  
- ZeroDivisionError: возникает при попытке деления на ноль.  
- IndexError: возникает при выходе за пределы списка, кортежа или строки.  
- ValueError: возникает при передаче недопустимого значения функции или операции.  
- TypeError: возникает при попытке выполнить операцию с неподходящим типом данных.  
- OSError: возникает при возникновении ошибок ввода-вывода, таких как ошибки открытия файлов или чтения с устройств.  
- KeyboardInterrupt: возникает при нажатии пользователем комбинации клавиш прерывания (обычно Ctrl+C).

2.2 Примеры:







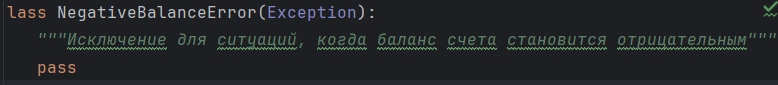
2.3 Стратегия обработки исключений

При обработке исключений важно учитывать следующие стратегии:  
  
- Обработка конкретных исключений: Обрабатывайте конкретные типы исключений (например, ZeroDivisionError), чтобы обеспечить соответствующую обработку.  
- Повторная попытка операции: В некоторых случаях можно повторить операцию после обработки исключения. Например, при временной ошибке сети можно повторить запрос через некоторое время.  
- Регистрация ошибок: Регистрируйте исключения в лог-файле или другом хранилище для последующего анализа и устранения неполадок.  
- Пользовательские сообщения об ошибках: Предоставляйте понятные и полезные сообщения об ошибках пользователям, чтобы помочь им понять, что произошло.  
- Передача исключений: В некоторых случаях необходимо передать исключение в вышестоящую функцию или класс для дальнейшей обработки.

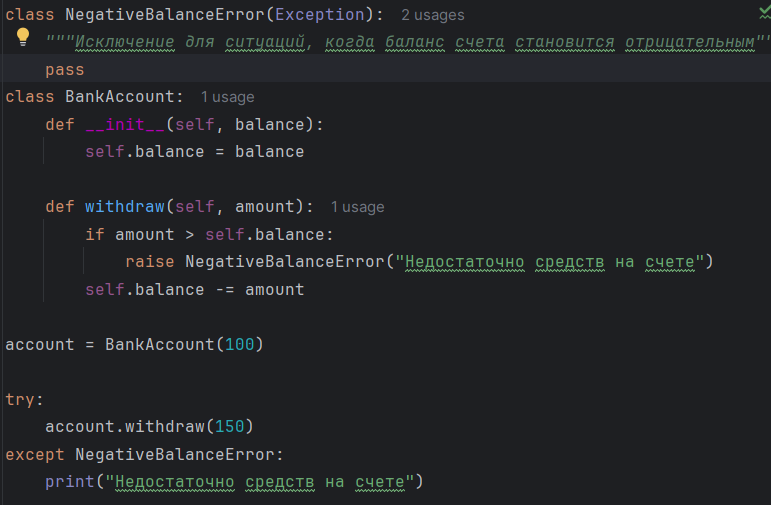
Задание 3: Генерация пользовательских исключений

3.1 Создание пользовательского класса исключений

Допустим, у нас есть программный проект для управления банковским счётом. Нам нужно создать пользовательское исключение для обработки случаев, когда баланс счета становится отрицательным.



3.2 Пример кода, который генерирует и обрабатывает это исключение



3.1 Обоснование необходимости создания пользовательского исключения

Создание пользовательского исключения оправдано в следующих случаях:  
  
- Уникальные ситуации: Если в вашем проекте есть специфические ситуации, которые не охватываются стандартными исключениями.  
- Понятные сообщения об ошибках: Пользовательские исключения позволяют создавать понятные и информативные сообщения об ошибках, которые могут быть легко поняты пользователями или разработчиками.  
- Инкапсуляция логики обработки: Пользовательские исключения позволяют инкапсулировать логику обработки исключений в отдельном классе, что делает код более читаемым и поддерживаемым.  
- Устранение дублирования кода: Если в вашем проекте есть несколько мест, где обрабатывается одна и та же ситуация с исключениями, создание пользовательского исключения позволяет избежать дублирования кода.  
  
В нашем примере создание пользовательского исключения NegativeBalanceError позволяет нам обрабатывать ситуации с отрицательным балансом счета более удобным и информативным способом.

**Выводы работы**

Обработка исключений является важной частью разработки надежного и устойчивого программного обеспечения на Python. Понимание различных типов исключений, стратегий обработки и использование пользовательских исключений позволяет разработчикам эффективно справляться с ошибками и обеспечивать бесперебойную работу приложений.