

## Исключения

Во всех задачах обязательно нужно использовать

“Множественная обработка исключений”

Синтаксис множественной обработки исключений

try:

# Код, который может вызвать исключение

except:

# первое исключение

except:

# второе исключение если есть

else:

# Блок, который выполняется, если не было исключений

finally:

# Блок, который выполняется всегда (независимо от того, было исключение или нет)

1. Создайте программу для расписания поездов. Пользователь вводит время отправления, время прибытия и день недели. Программа должна определить, является ли это утренним, дневным или вечерним поездом.

2. Деление на ноль: Пользователь вводит два числа. Попробуйте выполнить деление первого числа на второе число. Обработайте исключение `ZeroDivisionError`, если второе число равно нулю.

3. Ввод неправильного типа данных: Пользователь вводит число, но вместо числа случайно вводит строку. Попробуйте выполнить операцию с этим числом. Обработайте исключение `ValueError`.

4. Ошибка преобразования типов: Пользователь вводит строку, которая должна быть преобразована в число. Попробуйте преобразовать введенную строку в число. Если введенная строка не является числом, выведите сообщение "Ошибка преобразования типов". Обработайте исключение `ValueError`.

5. Некорректное использование операторов: Попробуйте выполнить некорректную операцию, например, деление строки на строку. Если происходит некорректное использование операторов, выведите сообщение "Некорректное использование операторов". Обработайте исключение `TypeError`.

6. Доступ к несуществующему индексу: Создайте список из нескольких элементов и попробуйте обратиться к элементу списка с индексом, который выходит за пределы списка. Обработайте исключение `IndexError`.

7. Деление строки на число: Пользователь вводит строку и число. Попробуйте разделить введенную строку на это число. Если введенная строка не может быть преобразована в число, выведите сообщение "Ошибка преобразования". Обработайте исключения `ValueError` и `TypeError`.

### **Вложенные блоки try-except:**

8. Деление на ноль и проверка на число: Пользователь вводит два числа. Попробуйте выполнить деление первого числа на второе. Если пользователь ввел не число, обработайте это исключение.

9. Возведение в степень и квадратный корень: Пользователь вводит число и степень. Попробуйте выполнить возведение числа в указанную степень. Если пользователь ввел отрицательную степень или не число, обработайте это исключение.

10. Сложение и деление с остатком: Пользователь вводит два числа. Попробуйте выполнить сложение и деление первого числа на второе с остатком. Обработайте исключения деления на ноль и ввода не числа.