



DS
Программирование
Python

Обработка ошибок и исключений



Введение



◆ *Что произойдет, если мы попытаемся целое число разделить на 0?*



◆ **Что произойдет, если мы попытаемся целое число разделить на 0?**
Увидим ошибку о делении на ноль - `ZeroDivisionError`



- ◆ **Что произойдет, если мы попытаемся целое число разделить на 0?**
Увидим ошибку о делении на ноль - `ZeroDivisionError`
- ◆ **Какую ошибку увидим на экране, если попытаемся обратиться к элементу списка по индексу, которого не существует?**



- ◆ **Что произойдет, если мы попытаемся целое число разделить на 0?**
Увидим ошибку о делении на ноль - `ZeroDivisionError`
- ◆ **Какую ошибку увидим на экране, если попытаемся обратиться к элементу списка по индексу, которого не существует?**
Увидим ошибку `IndexError`



- ◆ **Что произойдет, если мы попытаемся целое число разделить на 0?**
Увидим ошибку о делении на ноль - `ZeroDivisionError`
- ◆ **Какую ошибку увидим на экране, если попытаемся обратиться к элементу списка по индексу, которого не существует?**
Увидим ошибку `IndexError`
- ◆ **Функция `int()` может преобразовать любой объект в целое число?**



- ◆ **Что произойдет, если мы попытаемся целое число разделить на 0?**
Увидим ошибку о делении на ноль - `ZeroDivisionError`
- ◆ **Какую ошибку увидим на экране, если попытаемся обратиться к элементу списка по индексу, которого не существует?**
Увидим ошибку `IndexError`
- ◆ **Функция `int()` может преобразовать любой объект в целое число?**
Нет, только число и строку, состоящую из цифр



- ◆ **Что произойдет, если мы попытаемся целое число разделить на 0?**
Увидим ошибку о делении на ноль - `ZeroDivisionError`
- ◆ **Какую ошибку увидим на экране, если попытаемся обратиться к элементу списка по индексу, которого не существует?**
Увидим ошибку `IndexError`
- ◆ **Функция `int()` может преобразовать любой объект в целое число?**
Нет, только число и строку, состоящую из цифр
- ◆ **Можно ли использовать условный оператор `if` без оператора `else`?**



- ◆ **Что произойдет, если мы попытаемся целое число разделить на 0?**
Увидим ошибку о делении на ноль - `ZeroDivisionError`
- ◆ **Какую ошибку увидим на экране, если попытаемся обратиться к элементу списка по индексу, которого не существует?**
Увидим ошибку `IndexError`
- ◆ **Функция `int()` может преобразовать любой объект в целое число?**
Нет, только число и строку, состоящую из цифр
- ◆ **Можно ли использовать условный оператор `if` без оператора `else`?**
Да, можно



- ◆ **Что произойдет, если мы попытаемся целое число разделить на 0?**
Увидим ошибку о делении на ноль - `ZeroDivisionError`
- ◆ **Какую ошибку увидим на экране, если попытаемся обратиться к элементу списка по индексу, которого не существует?**
Увидим ошибку `IndexError`
- ◆ **Функция `int()` может преобразовать любой объект в целое число?**
Нет, только число и строку, состоящую из цифр
- ◆ **Можно ли использовать условный оператор `if` без оператора `else`?**
Да, можно
- ◆ **Можно ли использовать условный оператор `elif` без оператора `if`?**



- ◆ **Что произойдет, если мы попытаемся целое число разделить на 0?**
Увидим ошибку о делении на ноль - `ZeroDivisionError`
- ◆ **Какую ошибку увидим на экране, если попытаемся обратиться к элементу списка по индексу, которого не существует?**
Увидим ошибку `IndexError`
- ◆ **Функция `int()` может преобразовать любой объект в целое число?**
Нет, только число и строку, состоящую из цифр
- ◆ **Можно ли использовать условный оператор `if` без оператора `else`?**
Да, можно
- ◆ **Можно ли использовать условный оператор `elif` без оператора `if`?**
Нет, нельзя





Синтаксическая ошибка – появляется, если написанный код не соответствует правилам языка (например, не стоит отступ там, где это нужно)

Исключение – появляется во время работы программы (например, обращение к элементу списка по несуществующему индексу)





» **SyntaxError** – синтаксическая ошибка в коде.

Например, `data = [1, 2, 3` – появится ошибка из-за незакрытой квадратной скобки

» **IndentationError** – ошибка, возникающая из-за проблем с отступами.

Например,

```
if 2 + 2 > 3:  
    print('Здесь была математика')
```

В таком коде появится ошибка из-за того, что код внутри тела оператора `if` должен выделяться отступом.





Задание № 1

«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».



Задание № 1

«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
num_1 = int(input())  
num_2 = int(input())
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
num_1 = int(input())  
num_2 = int(input())  
if num_2 == 0:
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
num_1 = int(input())
num_2 = int(input())
if num_2 == 0:
    print('Делить на ноль нельзя!')
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
num_1 = int(input())
num_2 = int(input())
if num_2 == 0:
    print('Делить на ноль нельзя!')
else:
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
num_1 = int(input())
num_2 = int(input())
if num_2 == 0:
    print('Делить на ноль нельзя!')
else:
    print(num_1 / num_2)
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
num_1 = int(input())
num_2 = int(input())
if num_2 == 0:
    print('Делить на ноль нельзя!')
else:
    print(num_1 / num_2)
```

Если бы **не было** проверки на равенство нулю



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
num_1 = int(input())
num_2 = int(input())
if num_2 == 0:
    print('Делить на ноль нельзя!')
else:
    print(num_1 / num_2)
```

```
>>> ZeroDivisionError
```

Если бы **не было** проверки на равенство нулю





«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
num_1 = int(input())  
num_2 = int(input())
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
num_1 = int(input())  
num_2 = int(input())  
try: # код, где потенциально может возникнуть ошибка
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
num_1 = int(input())
num_2 = int(input())
try: # код, где потенциально может возникнуть ошибка
    print(num_1 / num_2)
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
num_1 = int(input())
num_2 = int(input())
try: # код, где потенциально может возникнуть ошибка
    print(num_1 / num_2)
except: # обработка ошибки деления на ноль
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
num_1 = int(input())
num_2 = int(input())
try: # код, где потенциально может возникнуть ошибка
    print(num_1 / num_2)
except: # обработка ошибки деления на ноль
    print('Возникла ошибка')
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
try: # код, где потенциально может возникнуть ошибка
    num_1 = int(input())
    num_2 = int(input())
    print(num_1 / num_2)
except: # обработка ошибки деления на ноль
    print('Возникла ошибка')
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
try: # код, где потенциально может возникнуть ошибка
    num_1 = int(input())
    num_2 = int(input())
    print(num_1 / num_2)
# обработка ошибки деления на ноль
except ZeroDivisionError:
    print('На ноль делить нельзя!')
```




«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
try: # код, где потенциально может возникнуть ошибка
    num_1 = int(input())
    num_2 = int(input())
    print(num_1 / num_2)
# обработка ошибки деления на ноль
except ZeroDivisionError:
    print('На ноль делить нельзя!')
# обработка некорректного ввода
except ValueError:
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
try: # код, где потенциально может возникнуть ошибка
    num_1 = int(input())
    num_2 = int(input())
    print(num_1 / num_2)
# обработка ошибки деления на ноль
except ZeroDivisionError:
    print('На ноль делить нельзя!')
# обработка некорректного ввода
except ValueError:
    print('Введено некорректное значение!')
```



- **try** – блок кода, где могут возникнуть *ошибки*
- **except** – блок кода, который выполняется при *возникновении* ошибки. Без указания конкретной ошибки способен отлавливать все возникающие ошибки и исключения.







«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
try: # код, где потенциально может возникнуть ошибка
    num_1 = int(input())
    num_2 = int(input())
    print(num_1 / num_2)
# обработка ошибки деления на ноль
except ZeroDivisionError:
    print('На ноль делить нельзя!')
except ValueError: # обработка некорректного ввода
    print('Введено некорректное значение!')
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
try: # код, где потенциально может возникнуть ошибка
    num_1 = int(input())
    num_2 = int(input())
    result = num_1 / num_2
# обработка ошибки деления на ноль
except ZeroDivisionError:
    print('На ноль делить нельзя!')
except ValueError: # обработка некорректного ввода
    print('Введено некорректное значение!')
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
try: # код, где потенциально может возникнуть ошибка
    num_1 = int(input())
    num_2 = int(input())
    result = num_1 / num_2
# обработка ошибки деления на ноль
except ZeroDivisionError:
    print('На ноль делить нельзя!')
except ValueError: # обработка некорректного ввода
    print('Введено некорректное значение!')
else: # если ошибок нет - выводим частное от деления
    print(f'Частное от деления: {result}')
```



- **try** – блок кода, где могут возникнуть *ошибки*
- **except** – блок кода, который выполняется при *возникновении* ошибки. Без указания конкретной ошибки способен отлавливать все возникающие ошибки и исключения.
- **else** – блок кода, который выполняется при *отсутствии* ошибок





«Второй шанс»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран остаток от деления первого числа на второе. Если происходит деление на ноль – вывести на экран фразу «Делить на ноль нельзя, введите новое число:» и запросить новое число для ввода. Если пользователь вводит некорректные значения для чисел – вывести на экран фразу «Введены некорректные значения» и попросить пользователя указать оба числа заново.



«Второй шанс»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран остаток от деления первого числа на второе. Если происходит деление на ноль – вывести на экран фразу «Делить на ноль нельзя, введите новое число:» и запросить новое число для ввода. Если пользователь вводит некорректные значения для чисел – вывести на экран фразу «Введены некорректные значения» и попросить пользователя

```
try: # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
```



«Второй шанс»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран остаток от деления первого числа на второе. Если происходит деление на ноль – вывести на экран фразу «Делить на ноль нельзя, введите новое число:» и запросить новое число для ввода. Если пользователь вводит некорректные значения для чисел – вывести на экран фразу «Введены некорректные значения» и попросить пользователя

```
try: # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
```



«Второй шанс»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран остаток от деления первого числа на второе. Если происходит деление на ноль – вывести на экран фразу «Делить на ноль нельзя, введите новое число:» и запросить новое число для ввода. Если пользователь вводит некорректные значения для чисел – вывести на экран фразу «Введены некорректные значения» и попросить пользователя

```
try:    # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
    # повторно вводим второе число
    num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
```



«Второй шанс»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран остаток от деления первого числа на второе. Если происходит деление на ноль – вывести на экран фразу «Делить на ноль нельзя, введите новое число:» и запросить новое число для ввода. Если пользователь вводит некорректные значения для чисел – вывести на экран фразу «Введены некорректные значения» и попросить пользователя

```
try: # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
    # повторно вводим второе число
    num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
except ValueError: # обработка некорректного ввода
```



«Второй шанс»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран остаток от деления первого числа на второе. Если происходит деление на ноль – вывести на экран фразу «Делить на ноль нельзя, введите новое число:» и запросить новое число для ввода. Если пользователь вводит некорректные значения для чисел – вывести на экран фразу «Введены некорректные значения» и попросить пользователя

```
try:    # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
    # повторно вводим второе число
    num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
except ValueError: # обработка некорректного ввода
    print('Введены некорректные значения!')
```



«Второй шанс»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран остаток от деления первого числа на второе. Если происходит деление на ноль – вывести на экран фразу «Делить на ноль нельзя, введите новое число:» и запросить новое число для ввода. Если пользователь вводит некорректные значения для чисел – вывести на экран фразу «Введены некорректные значения» и попросить пользователя

```
try:    # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
    # повторно вводим второе число
    num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
except ValueError: # обработка некорректного ввода
    print('Введены некорректные значения!')
    # повторно вводим оба числа
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
```



«Второй шанс»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран остаток от деления первого числа на второе. Если происходит деление на ноль – вывести на экран фразу «Делить на ноль нельзя, введите новое число:» и запросить новое число для ввода. Если пользователь вводит некорректные значения для чисел – вывести на экран фразу «Введены некорректные значения» и попросить пользователя

```
try: # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
    # повторно вводим второе число
    num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
except ValueError: # обработка некорректного ввода
    print('Введены некорректные значения!')
    # повторно вводим оба числа
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
else: # если ошибки отсутствуют - выводим остаток от деления
    print(f'Остаток от деления: {result}')
```




«Второй шанс»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран остаток от деления первого числа на второе. Если происходит деление на ноль – вывести на экран фразу «Делить на ноль нельзя, введите новое число:» и запросить новое число для ввода. Если пользователь вводит некорректные значения для чисел – вывести на экран фразу «Введены некорректные значения» и попросить пользователя

```
try: # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
    # повторно вводим второе число
    num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
    result = num_1 % num_2 # повторно вычисляем остаток от деления
except ValueError: # обработка некорректного ввода
    print('Введены некорректные значения!')
    # повторно вводим оба числа
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
else: # если ошибки отсутствуют - выводим остаток от деления
    print(f'Остаток от деления: {result}')
```



«Второй шанс»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран остаток от деления первого числа на второе. Если происходит деление на ноль – вывести на экран фразу «Делить на ноль нельзя, введите новое число:» и запросить новое число для ввода. Если пользователь вводит некорректные значения для чисел – вывести на экран фразу «Введены некорректные значения» и попросить пользователя

```
try:    # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
    # повторно вводим второе число
    num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
    result = num_1 % num_2 # повторно вычисляем остаток от деления
    print(f'Остаток от деления: {result}')
except ValueError: # обработка некорректного ввода
    print('Введены некорректные значения!')
    # повторно вводим оба числа
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
else: # если ошибки отсутствуют - выводим остаток от деления
    print(f'Остаток от деления: {result}')
```



«Второй шанс»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран остаток от деления первого числа на второе. Если происходит деление на ноль – вывести на экран фразу «Делить на ноль нельзя, введите новое число:» и запросить новое число для ввода. Если пользователь вводит некорректные значения для чисел – вывести на экран фразу «Введены некорректные значения» и попросить пользователя

```
try:    # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
    # повторно вводим второе число
    num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
    result = num_1 % num_2 # повторно вычисляем остаток от деления
    print(f'Остаток от деления: {result}')
except ValueError: # обработка некорректного ввода
    print('Введены некорректные значения!')
    # повторно вводим оба числа
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2 # повторно вычисляем остаток от деления
else: # если ошибки отсутствуют - выводим остаток от деления
    print(f'Остаток от деления: {result}')
```



«Второй шанс»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран остаток от деления первого числа на второе. Если происходит деление на ноль – вывести на экран фразу «Делить на ноль нельзя, введите новое число:» и запросить новое число для ввода. Если пользователь вводит некорректные значения для чисел – вывести на экран фразу «Введены некорректные значения» и попросить пользователя

```
try: # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
    # повторно вводим второе число
    num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
    result = num_1 % num_2 # повторно вычисляем остаток от деления
    print(f'Остаток от деления: {result}')
except ValueError: # обработка некорректного ввода
    print('Введены некорректные значения!')
    # повторно вводим оба числа
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2 # повторно вычисляем остаток от деления
    print(f'Остаток от деления: {result}')
else: # если ошибки отсутствуют - выводим остаток от деления
    print(f'Остаток от деления: {result}')
```





«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
try:
    num_1 = int(input())
    num_2 = int(input())
    result = num_1 / num_2
except ZeroDivisionError:
    print('На ноль делить нельзя!')
except ValueError:
    print('Введено некорректное значение!')
else:
    print(f'Частное от деления: {result}')
```



«Деление на ноль»

На вход программе с клавиатуры подаются 2 натуральных числа. Программа должна вывести на экран частное от деления первого числа на второе. Если второе число равно нулю – вывести на экран фразу «Делить на ноль нельзя!».

```
try:
    num_1 = int(input())
    num_2 = int(input())
    result = num_1 / num_2
except ZeroDivisionError:
    print('На ноль делить нельзя!')
except ValueError:
    print('Введено некорректное значение!')
else:
    print(f'Частное от деления: {result}')
finally:
    print('Работа программы закончена!')
```



- **try** – блок кода, где могут возникнуть *ошибки*
- **except** – блок кода, который выполняется при *возникновении* ошибки. Без указания конкретной ошибки способен отлавливать все возникающие ошибки и исключения.
- **else** – блок кода, который выполняется при *отсутствии* ошибок
- **finally** – блок кода, который выполняется *в любом случае*

Возможные конфигурации блоков:

- try-except
- try-except-else
- try-except-else-finally
- try-except-finally
- try-finally





«Второй шанс 2.0»

Доработайте программу с нахождением остатка от деления, используя блок `finally` для устранения дублирования в коде.

```
try:  # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
    # повторно вводим второе число
    num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
    result = num_1 % num_2 # повторно вычисляем остаток от деления
    print(f'Остаток от деления: {result}')
except ValueError: # обработка некорректного ввода
    print('Введены некорректные значения!')
    # повторно вводим оба числа
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2 # повторно вычисляем остаток от деления
    print(f'Остаток от деления: {result}')
else: # если ошибки отсутствуют - выводим остаток от деления
    print(f'Остаток от деления: {result}')
```



«Второй шанс 2.0»

Доработайте программу с нахождением остатка от деления, используя блок `finally` для устранения дублирования в коде.

```
try: # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
    num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
except ValueError: # обработка некорректного ввода
    print('Введено некорректны значения!')
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
finally:
```



«Второй шанс 2.0»

Доработайте программу с нахождением остатка от деления, используя блок `finally` для устранения дублирования в коде.

```
try: # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
    num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
except ValueError: # обработка некорректного ввода
    print('Введено некорректны значения!')
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
finally:
    result = num_1 % num_2
```



«Второй шанс 2.0»

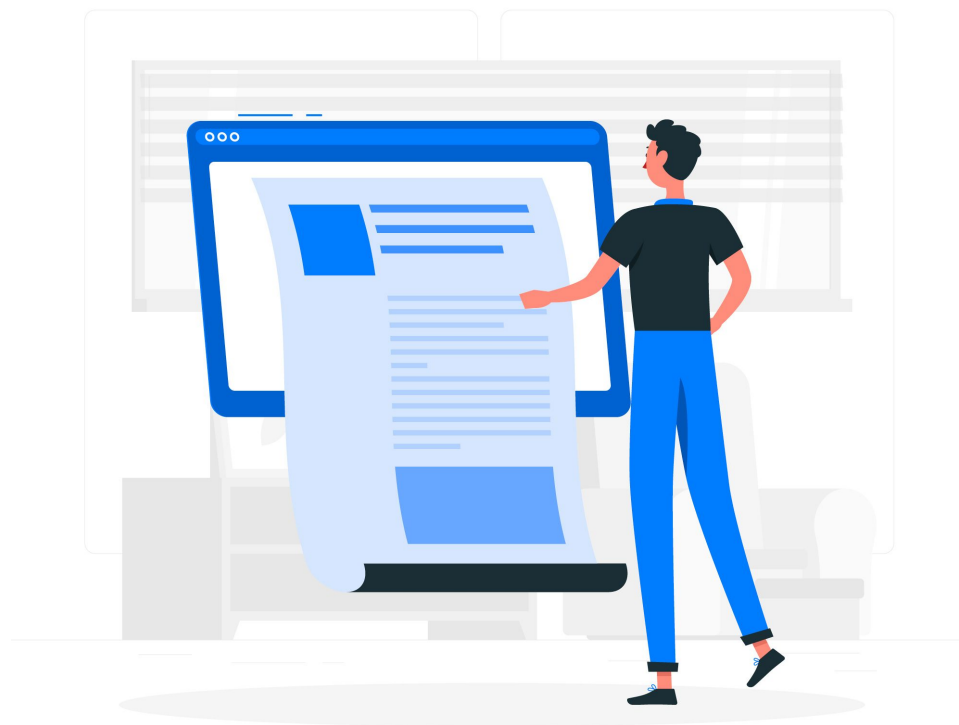
Доработайте программу с нахождением остатка от деления, используя блок `finally` для устранения дублирования в коде.

```
try: # код с вычислением остатка от деления
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
    result = num_1 % num_2
except ZeroDivisionError: # обработка ошибки деления на ноль
    num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
except ValueError: # обработка некорректного ввода
    print('Введено некорректны значения!')
    num_1 = int(input('Введите первое число:'))
    num_2 = int(input('Введите второе число:'))
finally:
    result = num_1 % num_2
    print(f'Остаток от деления: {result}')
```





Лог – специальный текстовый файл, в котором хранится информация о действиях пользователя или программы.





«Лог калькулятора»

На вход программе с клавиатуры вводится имя пользователя, после чего пользователь вводит одно число, затем операцию, после вводит второе число. Программа работает с указанными числами и операцией. Если возникают исключения – программа выводит информацию об этом на экран и просит пользователя указать новые данные для работы. После ввода корректных данных программа выводит результат вычислений на экран и спрашивает пользователя, хочет ли он продолжать работать с калькулятором. В случае положительного ответа, пользователь продолжает работать с калькулятором в прежней логике.

Программа также должна сохранять в текстовый файл `log_calc.txt` следующую информацию:

- имя текущего пользователя
- введенные пользователем данные
- результаты вычислений
- информацию об ошибках
- новые данные от пользователя, если возникла ошибка

С каждым новым сеансом программа должна добавлять новую информацию в конец текстового файла.



Задание № 4

```
name = input('Укажите свое имя:') # ввод имени пользователя
```




Задание № 4

```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
```



Задание № 4

```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name} \n') # запись в лог-файл введенного имени пользователя
```



Задание № 4

```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name} \n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')
```




```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')

-----

print(eval('2 + 2'))
```



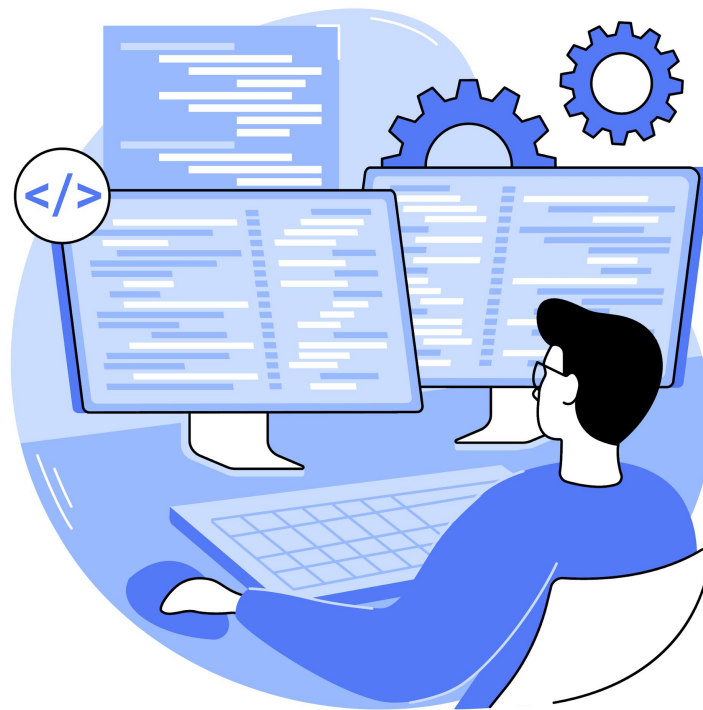
```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')
```

```
print(eval('2 + 2'))
```

```
>>> 4
```



`eval()` – функция, принимающая выражение в виде строки и выполняющая его.





```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')
        result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')
        result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
    except ZeroDivisionError: # обработка ошибки деления на ноль
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')
        result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
    except ZeroDivisionError: # обработка ошибки деления на ноль
        num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')
        result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
    except ZeroDivisionError: # обработка ошибки деления на ноль
        num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
        file.write(f'Ошибка деления на ноль\n') # запись информации в лог-файл
```




```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')
        result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
    except ZeroDivisionError: # обработка ошибки деления на ноль
        num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
        file.write(f'Ошибка деления на ноль\n') # запись информации в лог-файл
        file.write(f'Введено новое число: {num_2}\n')
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')
        result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
    except ZeroDivisionError: # обработка ошибки деления на ноль
        num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
        file.write(f'Ошибка деления на ноль\n') # запись информации в лог-файл
        file.write(f'Введено новое число: {num_2}\n')
    except ValueError: # обработка некорректного ввода чисел
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')
        result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
    except ZeroDivisionError: # обработка ошибки деления на ноль
        num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
        file.write(f'Ошибка деления на ноль\n') # запись информации в лог-файл
        file.write(f'Введено новое число: {num_2}\n')
    except ValueError: # обработка некорректного ввода чисел
        print('Введены некорректные значения!')
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')
        result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
    except ZeroDivisionError: # обработка ошибки деления на ноль
        num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
        file.write(f'Ошибка деления на ноль\n') # запись информации в лог-файл
        file.write(f'Введено новое число: {num_2}\n')
    except ValueError: # обработка некорректного ввода чисел
        print('Введены некорректные значения!')
        num_1 = int(input('Введите первое число:')) # повторный ввод первого числа
        num_2 = int(input('Введите второе число:')) # повторный ввод второго числа
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')
        result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
    except ZeroDivisionError: # обработка ошибки деления на ноль
        num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
        file.write(f'Ошибка деления на ноль\n') # запись информации в лог-файл
        file.write(f'Введено новое число: {num_2}\n')
    except ValueError: # обработка некорректного ввода чисел
        print('Введены некорректные значения!')
        num_1 = int(input('Введите первое число:')) # повторный ввод первого числа
        num_2 = int(input('Введите второе число:')) # повторный ввод второго числа
        file.write(f'Ввод некорректных значений\n')
```



```
name = input('Укажите свое имя:') # ввод имени пользователя
# открываем файл в режиме добавления
file = open('log_calc.txt', 'a', encoding='utf-8')
file.write(f'Пользователь: {name}\n') # запись в лог-файл введенного имени пользователя
while True: # цикл работы с калькулятором
    try: # ввод данных пользователем и вычисление результата
        num_1 = int(input('Введите первое число:')) # ввод чисел и знака операции
        oper = input('Введите операцию (+ - * / ** // %):')
        num_2 = int(input('Введите второе число:'))
        # запись в лог-файл о числах и операции
        file.write(f'Совершается операция: {num_1}{oper}{num_2}\n')
        result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
    except ZeroDivisionError: # обработка ошибки деления на ноль
        num_2 = int(input('Делить на ноль нельзя, введите новое число:'))
        file.write(f'Ошибка деления на ноль\n') # запись информации в лог-файл
        file.write(f'Введено новое число: {num_2}\n')
    except ValueError: # обработка некорректного ввода чисел
        print('Введены некорректные значения!')
        num_1 = int(input('Введите первое число:')) # повторный ввод первого числа
        num_2 = int(input('Введите второе число:')) # повторный ввод второго числа
        file.write(f'Ввод некорректных значений\n')
        file.write(f'Введены новые числа: {num_1} и {num_2}\n')
```



-
-
-

```
except SyntaxError: # обработка некорректного ввода операции
```



• • •

```
except SyntaxError: # обработка некорректного ввода операции  
    # повторный ввод операции
```

```
oper = input('Введите корректную операцию (+ - * / ** // %):' )
```




• • •

```
except SyntaxError: # обработка некорректного ввода операции
    # повторный ввод операции
    oper = input('Введите корректную операцию (+ - * / ** // %):' )
    file.write(f'Ввод некорректной операций \n') # запись информации в лог-файл
    file.write(f'Введена новая операция: {oper} \n')
```



```
• • •  
except SyntaxError: # обработка некорректного ввода операции  
    # повторный ввод операции  
    oper = input('Введите корректную операцию (+ - * / ** // %):' )  
    file.write(f'Ввод некорректной операций \n') # запись информации в лог-файл  
    file.write(f'Введена новая операция: {oper} \n')  
finally: # финальное вычисление результата и запрос новых данных от пользователя
```



• • •

```
except SyntaxError: # обработка некорректного ввода операции
    # повторный ввод операции
    oper = input('Введите корректную операцию (+ - * / ** // %):' )
    file.write(f'Ввод некорректной операций \n') # запись информации в лог-файл
    file.write(f'Введена новая операция: {oper} \n')
finally: # финальное вычисление результата и запрос новых данных от пользователя
    result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
```



```
• • •
except SyntaxError: # обработка некорректного ввода операции
    # повторный ввод операции
    oper = input('Введите корректную операцию (+ - * / ** // %):' )
    file.write(f'Ввод некорректной операций \n') # запись информации в лог-файл
    file.write(f'Введена новая операция: {oper} \n')
finally: # финальное вычисление результата и запрос новых данных от пользователя
    result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
    print(f'Результат операции: {num_1}{oper}{num_2} = {result}')
```



```
• • •  
except SyntaxError: # обработка некорректного ввода операции  
    # повторный ввод операции  
    oper = input('Введите корректную операцию (+ - * / ** // %):' )  
    file.write(f'Ввод некорректной операций \n') # запись информации в лог-файл  
    file.write(f'Введена новая операция: {oper} \n')  
finally: # финальное вычисление результата и запрос новых данных от пользователя  
    result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата  
    print(f'Результат операции: {num_1}{oper}{num_2} = {result} ')  
    # запись информации в лог-файл  
    file.write(f'Результат: {num_1}{oper}{num_2} = {result} \n')
```



```
• • •
except SyntaxError: # обработка некорректного ввода операции
    # повторный ввод операции
    oper = input('Введите корректную операцию (+ - * / ** // %):' )
    file.write(f'Ввод некорректной операций \n') # запись информации в лог-файл
    file.write(f'Введена новая операция: {oper} \n')
finally: # финальное вычисление результата и запрос новых данных от пользователя
    result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
    print(f'Результат операции: {num_1}{oper}{num_2} = {result} ')
    # запись информации в лог-файл
    file.write(f'Результат: {num_1}{oper}{num_2} = {result} \n')
    # запрос пользователю на продолжение работы
    is_continue = input('Продолжить работу с калькулятором? (Y/N):' )
```



```
• • •  
except SyntaxError: # обработка некорректного ввода операции  
    # повторный ввод операции  
    oper = input('Введите корректную операцию (+ - * / ** // %):' )  
    file.write(f'Ввод некорректной операций \n') # запись информации в лог-файл  
    file.write(f'Введена новая операция: {oper} \n')  
finally: # финальное вычисление результата и запрос новых данных от пользователя  
    result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата  
    print(f'Результат операции: {num_1}{oper}{num_2} = {result} ')  
    # запись информации в лог-файл  
    file.write(f'Результат: {num_1}{oper}{num_2} = {result} \n')  
    # запрос пользователю на продолжение работы  
    is_continue = input('Продолжить работу с калькулятором? (Y/N):' )  
    # если пользователь не хочет работать с калькулятором  
    if is_continue == 'N':
```



```
• • •
except SyntaxError: # обработка некорректного ввода операции
    # повторный ввод операции
    oper = input('Введите корректную операцию (+ - * / ** // %):' )
    file.write(f'Ввод некорректной операций \n') # запись информации в лог-файл
    file.write(f'Введена новая операция: {oper} \n')
finally: # финальное вычисление результата и запрос новых данных от пользователя
    result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
    print(f'Результат операции: {num_1}{oper}{num_2} = {result} ')
    # запись информации в лог-файл
    file.write(f'Результат: {num_1}{oper}{num_2} = {result} \n')
    # запрос пользователю на продолжение работы
    is_continue = input('Продолжить работу с калькулятором? (Y/N):' )
    # если пользователь не хочет работать с калькулятором
    if is_continue == 'N':
        print('До свидания!')
```




```
• • •  
except SyntaxError: # обработка некорректного ввода операции  
    # повторный ввод операции  
    oper = input('Введите корректную операцию (+ - * / ** // %):' )  
    file.write(f'Ввод некорректной операций \n') # запись информации в лог-файл  
    file.write(f'Введена новая операция: {oper} \n')  
finally: # финальное вычисление результата и запрос новых данных от пользователя  
    result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата  
    print(f'Результат операции: {num_1}{oper}{num_2} = {result} ')  
    # запись информации в лог-файл  
    file.write(f'Результат: {num_1}{oper}{num_2} = {result} \n')  
    # запрос пользователю на продолжение работы  
    is_continue = input('Продолжить работу с калькулятором? (Y/N):' )  
    # если пользователь не хочет работать с калькулятором  
    if is_continue == 'N':  
        print('До свидания!')  
    # закрываем файл и выходим из цикла  
    file.close()
```



```
• • •  
except SyntaxError: # обработка некорректного ввода операции  
    # повторный ввод операции  
    oper = input('Введите корректную операцию (+ - * / ** // %):' )  
    file.write(f'Ввод некорректной операций \n') # запись информации в лог-файл  
    file.write(f'Введена новая операция: {oper} \n')  
finally: # финальное вычисление результата и запрос новых данных от пользователя  
    result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата  
    print(f'Результат операции: {num_1}{oper}{num_2} = {result} ')  
    # запись информации в лог-файл  
    file.write(f'Результат: {num_1}{oper}{num_2} = {result} \n')  
    # запрос пользователю на продолжение работы  
    is_continue = input('Продолжить работу с калькулятором? (Y/N):' )  
    # если пользователь не хочет работать с калькулятором  
    if is_continue == 'N':  
        print('До свидания!')  
    # закрываем файл и выходим из цикла  
    file.close()  
    break
```



```
• • •
except SyntaxError: # обработка некорректного ввода операции
    # повторный ввод операции
    oper = input('Введите корректную операцию (+ - * / ** // %):' )
    file.write(f'Ввод некорректной операций \n') # запись информации в лог-файл
    file.write(f'Введена новая операция: {oper} \n')
finally: # финальное вычисление результата и запрос новых данных от пользователя
    result = eval(f'{num_1}{oper}{num_2} ') # вычисление результата
    print(f'Результат операции: {num_1}{oper}{num_2} = {result} ')
    # запись информации в лог-файл
    file.write(f'Результат: {num_1}{oper}{num_2} = {result} \n')
    # запрос пользователю на продолжение работы
    is_continue = input('Продолжить работу с калькулятором? (Y/N):' )
    # если пользователь не хочет работать с калькулятором
    if is_continue == 'N':
        print('До свидания!')
    # закрываем файл и выходим из цикла
    file.write('----- \n')
    file.close()
    break
print('-----')
```



- **try** – блок кода, где могут возникнуть *ошибки*
- **except** – блок кода, который выполняется при *возникновении* ошибки. Без указания конкретной ошибки способен отлавливать все возникающие ошибки и исключения.
- **else** – блок кода, который выполняется при *отсутствии* ошибок
- **finally** – блок кода, который выполняется *в любом случае*

Возможные конфигурации блоков:

- try-except
- try-except-else
- try-except-else-finally
- try-except-finally
- try-finally





DS
Программирование
Python

Спасибо за внимание!
