

Исключения в Python

Для вопросов по курсу:

natalya.razmochaeva@moevm.info

Префикс в теме письма [CS_03XX]

*Берленко Татьяна Андреевна
Шевская Наталья Владимировна
СПбГЭТУ “ЛЭТИ”, ФКТИ, МОЭВМ*

Ошибки и исключения

- Ошибки и исключения -- это особый класс объектов в языке Python.
- **Синтаксические** ошибки возникают в процессе компиляции программы на языке Python.
- Остальные ошибки (или исключения) возникают во время исполнения программы.
- Ошибки могут возникать в случае некорректного программного кода, а также генерироваться программистом.

Ошибки и исключения. Пример IndentationError

- Пример синтаксической ошибки:

```
data = '1'
```

```
    print(data)
```

```
# print(data)
```

```
# ^
```

```
# IndentationError: unexpected indent
```

Ошибки и исключения. Пример ValueError

➤ Пример исключения **ValueError**:

```
data = '1q'  
print(int(data))
```

Мы поймаем исключение ValueError:

ValueError: invalid literal for int() with base 10: '1q'

Ошибки и исключения -- внутреннее устройство

У любой ошибки есть:

➤ Тип (класс), например: `TypeError`, `ValueError`, ...

➤ Сообщение, например:

“invalid literal for int() with base 10: '1q'”

➤ Состояние стека вызовов на момент ошибки.

Стек вызовов

- Хранит список функций, которые вызывались.
- Функция должна быть вызвана -- интерпретатор кладется на стек.
- Функция отработала -- интерпретатор снимается со стека.
- В самом низу стека всегда находится функция `<module>`

Трассировочная информация

- это объект, который представляет стек вызовов в точке, где возникло исключение
- можно посмотреть с использованием встроенной функции `print_exc()` из стандартного модуля **traceback**

Конструкция try-except

- Конструкция try-except нужна для того, чтобы перехватить и обработать исключительные ситуации.
- Синтаксис:

try:

<Инструкция>

except <Тип_Исключения>:

<Обработка_Исключения>

Конструкция try-except. Пример

Пример:

```
data = input()
```

```
try:
```

```
    data = int(data)
```

```
except ValueError:
```

```
    print('Это не число!')
```

Конструкция try-except as

- В конструкции try-except мы также можем поймать объект ошибки, для этого используется следующий синтаксис:

```
try:  
    <Инструкция>  
except <Тип_Исключения> as <Имя_Объекта_Исключения>:  
    <Обработка_Объекта_Исключения>
```

Конструкция try-except as

```
def func(arr):  
    try:  
        print(min(arr))  
    except TypeError as e:  
        print(e)
```

```
func([1, 2, 3, 'f'])
```

Иерархия исключений

- Все исключения в языке Python участвуют в иерархии наследования.
- Базовый класс для всех исключений - **BaseException**, но программистам рекомендуется использовать **Exception** для создания собственных исключений.
- Поскольку при обнаружении исключительной ситуации в *try-except* используется функция **isinstance()**, мы можем отлавливать как объект указанного исключения, так и объект классов-наследников этого исключения.

Иерархия исключений. Пример

Что будет выведено на экран?:

```
try:
```

```
    c = 5 / 0
```

```
except Exception:
```

```
    print('Exception occurred')
```

```
except ArithmeticError:
```

```
    print('ArithmeticError occurred')
```

```
except ZeroDivisionError:
```

```
    print('ZeroDivisionError occurred')
```

Полиморфизм в исключениях. Пример

```
L_base = BaseException.__subclasses__() # получили всех наследников класса
```

```
BaseException
```

```
print(L_base)
```

```
print(len(L_base))
```

```
L_obj = object.__subclasses__() # получили всех наследников класса object
```

```
print(L_obj)
```

```
print(len(L_obj))
```

```
for e in L_base:
```

```
    print(e) # вне зависимости от особенностей класса -- полиморфизм -- выводим на экран
```

```
    print(e.__subclasses__())
```

Конструкция try-except-else-finally

try:

 <Инструкция>

except <Тип_Исключения>:

 <Обработка_Исключения>

else:

 # код для обработки случая,

 # когда в try-блоке не было поймано исключение <Тип_Исключения>

finally:

 # код, который нужно выполнить при любом исходе

Инструкция raise

- Мы можем сгенерировать (или *бросить*) исключение с помощью инструкции **raise**.
- Синтаксис:

raise <Создание объекта исключения>

- Пример:

raise TypeError('Тип переменной указан неверно!')

Повторение из ООП

- Представьте, что вы пишете свой собственный класс для определенной задачи (не важно, какой именно).
- Что надо сделать, чтобы объекты этого класса `obj` можно было бы передавать в функцию `print(obj)`?
- Что надо сделать, чтобы можно было использовать функцию `len(obj)` с объектами вашего класса?
- Что надо сделать, чтобы объекты вашего класса можно было сравнивать друг с другом и выполнять математические операции?

Вопросы по курсу можно задавать:

Шевская Наталья Владимировна
natalya.razmochaeva@moevm.info,

Берленко Татьяна Андреевна
tatyana.berlenko@moevm.info