

Теоретический материал к занятию Декораторы 4 занятие.

В Python исключения обрабатываются с помощью блоков try/except. Для этого операция, которая может вызвать исключение, помещается внутрь блока try. А код, который должен быть выполнен при возникновении ошибки, находится внутри except.

Исключения (exceptions) - ещё один тип данных в python. Исключения необходимы для того, чтобы сообщать программисту об ошибках.

Самый простейший пример исключения - деление на ноль, с которым мы встречались в калькуляторе

```
print(100 / 0)
```

Результат

```
Traceback (most recent call last):
  File "", line 1, in
    print(100 / 0)
ZeroDivisionError: division by zero
```

Разберём это сообщение подробнее: интерпретатор нам сообщает о том, что он поймал исключение и напечатал информацию (Traceback (most recent call last)).

Далее имя файла (File ""). Имя пустое, потому что мы находимся в интерактивном режиме, строка в файле (line 1);

Выражение, в котором произошла ошибка (100 / 0).

Название исключения (ZeroDivisionError) и краткое описание исключения (division by zero).

Разумеется, возможны и другие исключения:

```
print(2 + '1')
Traceback (most recent call last):
  File "", line 1, in
    print(2 + '1')
TypeError: unsupported operand type(s) for +: 'int' and 'str'

print(int('qwerty'))
Traceback (most recent call last):
  File "", line 1, in
    print(int('qwerty'))
ValueError: invalid literal for int() with base 10: 'qwerty'
```

В этих двух примерах генерируются исключения TypeError и ValueError соответственно.

Подсказки дают нам полную информацию о том, где порождено исключение, и с чем оно связано.

Рассмотрим иерархию встроенных в python исключений, хотя иногда вам могут встретиться и другие, так как программисты могут создавать собственные исключения.

- BaseException - базовое исключение, от которого берут начало все остальные.

- SystemExit - исключение, порождается функцией sys.exit при выходе из программы.
- KeyboardInterrupt - порождается при прерывании программы пользователем (обычно сочетанием клавиш Ctrl+C).
- GeneratorExit - порождается при вызове метода close объекта generator.
- Exception - а вот тут уже заканчиваются полностью системные исключения (которые лучше не трогать) и начинаются обыкновенные, с которыми можно работать.
 - StopIteration - порождается встроенной функцией next, если в итераторе больше нет элементов.
 - ArithmeticError - арифметическая ошибка.
 - FloatingPointError - порождается при неудачном выполнении операции с плавающей запятой. На практике встречается нечасто.
 - OverflowError - возникает, когда результат арифметической операции слишком велик для представления. Не появляется при обычной работе с целыми числами (так как python поддерживает длинные числа), но может возникать в некоторых других случаях.
 - ZeroDivisionError - деление на ноль.
 - AssertionError - выражение в функции assert ложно.
 - AttributeError - объект не имеет данного атрибута (значения или метода).
 - BufferError - операция, связанная с буфером, не может быть выполнена.
 - EOFError - функция наткнулась на конец файла и не смогла прочитать то, что хотела.
 - ImportError - не удалось импортирование модуля или его атрибута.
 - LookupError - некорректный индекс или ключ.
 - IndexError - индекс не входит в диапазон элементов.
 - KeyError - несуществующий ключ (в словаре, множестве или другом объекте).
 - MemoryError - недостаточно памяти.
 - NameError - не найдено переменной с таким именем.
 - UnboundLocalError - сделана ссылка на локальную переменную в функции, но переменная не определена ранее.
 - OSError - ошибка, связанная с системой.
 - BlockingIOError
 - ChildProcessError - неудача при операции с дочерним процессом.
 - ConnectionError - базовый класс для исключений, связанных с подключениями.

- BrokenPipeError
 - ConnectionAbortedError
 - ConnectionRefusedError
 - ConnectionResetError
- FileExistsError - попытка создания файла или директории, которая уже существует.
- FileNotFoundError - файл или директория не существует.
- InterruptedError - системный вызов прерван входящим сигналом.
- IsADirectoryError - ожидался файл, но это директория.
- NotADirectoryError - ожидалась директория, но это файл.
- PermissionError - не хватает прав доступа.
- ProcessLookupError - указанного процесса не существует.
- TimeoutError - закончилось время ожидания.
- ReferenceError - попытка доступа к атрибуту со слабой ссылкой.
- RuntimeError - возникает, когда исключение не попадает ни под одну из других категорий.
- NotImplementedError - возникает, когда абстрактные методы класса требуют переопределения в дочерних классах.
- SyntaxError - синтаксическая ошибка.
 - IndentationError - неправильные отступы.
 - TabError - смешивание в отступах табуляции и пробелов.
- SystemError - внутренняя ошибка.
- TypeError - операция применена к объекту несоответствующего типа.
- ValueError - функция получает аргумент правильного типа, но некорректного значения.
- UnicodeError - ошибка, связанная с кодированием / раскодированием unicode в строках.
 - UnicodeEncodeError - исключение, связанное с кодированием unicode.
 - UnicodeDecodeError - исключение, связанное с декодированием unicode.
 - UnicodeTranslateError - исключение, связанное с переводом unicode.
- Warning - предупреждение.