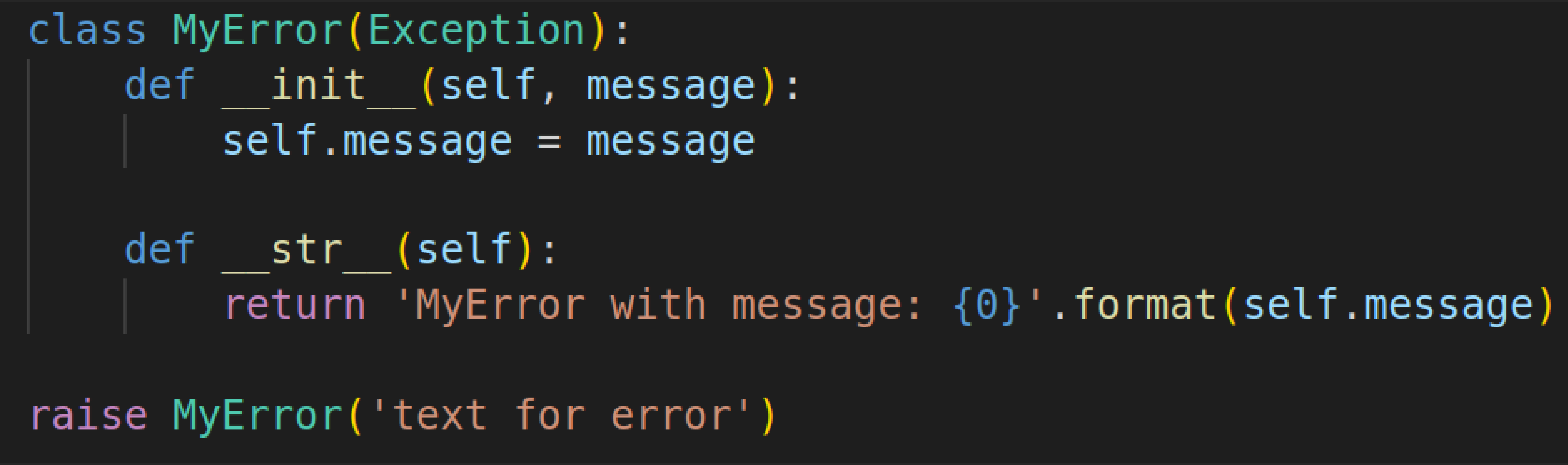
ООП Лекция 4.

Преподаватель: Тенигин Альберт Андреевич

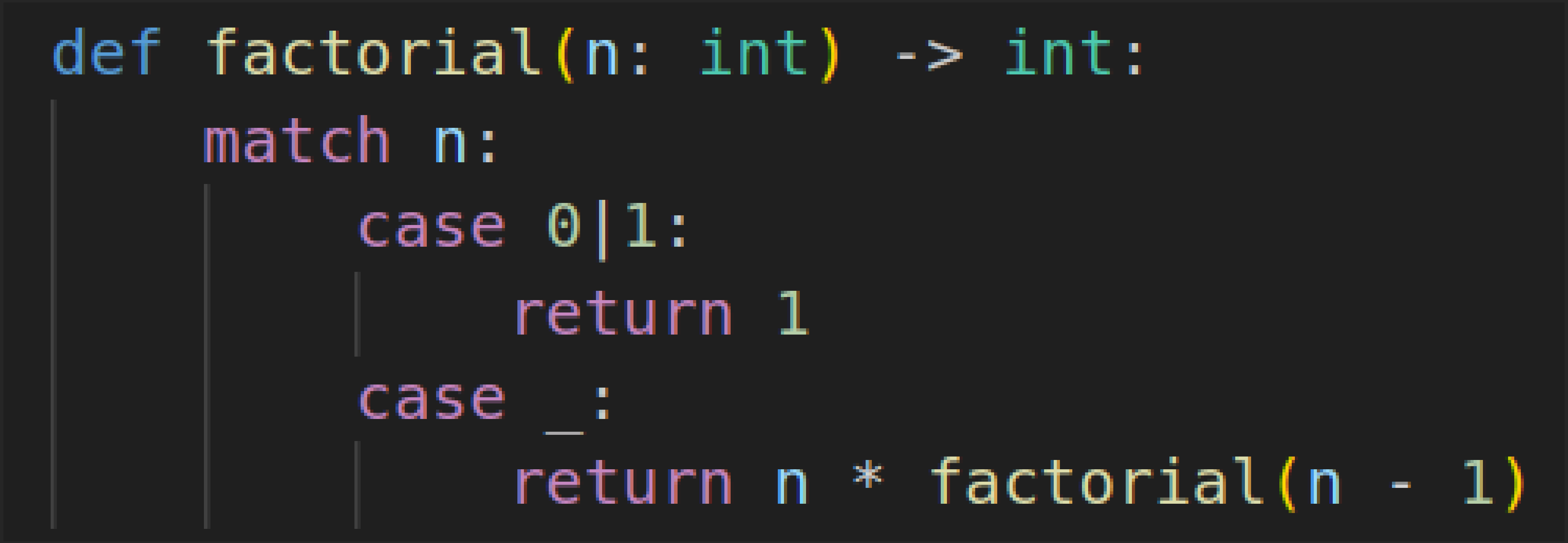
Сириус
IT-Колледж

Создание своих классов ошибок *(наследуемся от класса Exception!)*

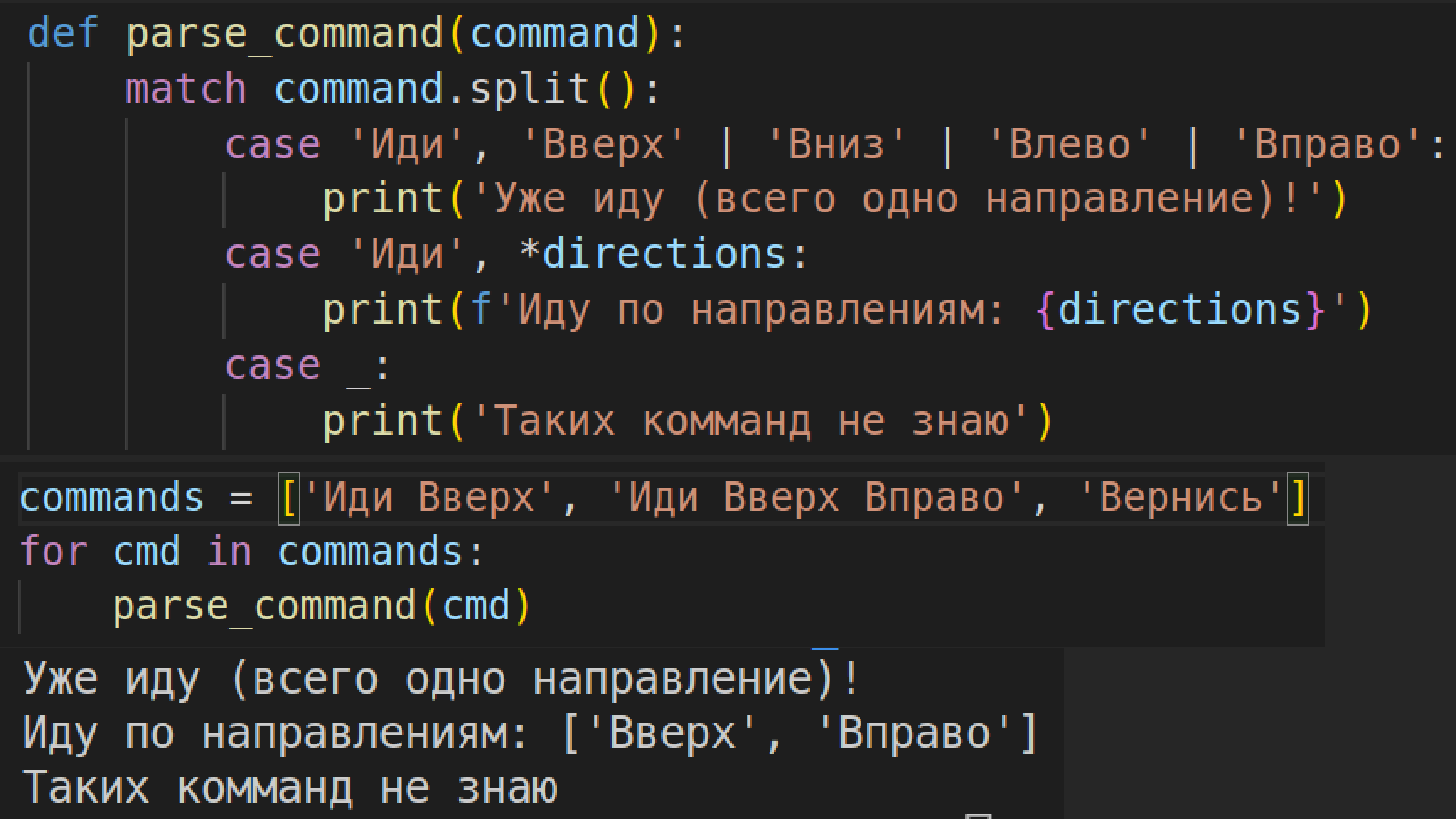


```python
class MyError(Exception):
    def __init__(self, message):
        self.message = message

    def __str__(self):
        return 'MyError with message: {0}'.format(self.message)

raise MyError('text for error')
```

```
BaseException
 ├── BaseExceptionGroup
 ├── GeneratorExit
 ├── KeyboardInterrupt
 ├── SystemExit
 └── Exception
      ├── ArithmeticError
      │    ├── FloatingPointError
      │    ├── OverflowError
      │    └── ZeroDivisionError
      ├── AssertionError
      ├── AttributeError
      ├── BufferError
      ├── EOFError
      ├── ExceptionGroup [BaseExceptionGroup]
      ├── ImportError
      │    └── ModuleNotFoundError
      ├── LookupError
      │    ├── IndexError
      │    └── KeyError
      ├── MemoryError
      ├── NameError
      │    └── UnboundLocalError
      ├── OSError
      │    ├── BlockingIOError
      │    ├── ChildProcessError
      │    ├── ConnectionError
      │    │    ├── BrokenPipeError
      │    │    ├── ConnectionAbortedError
```



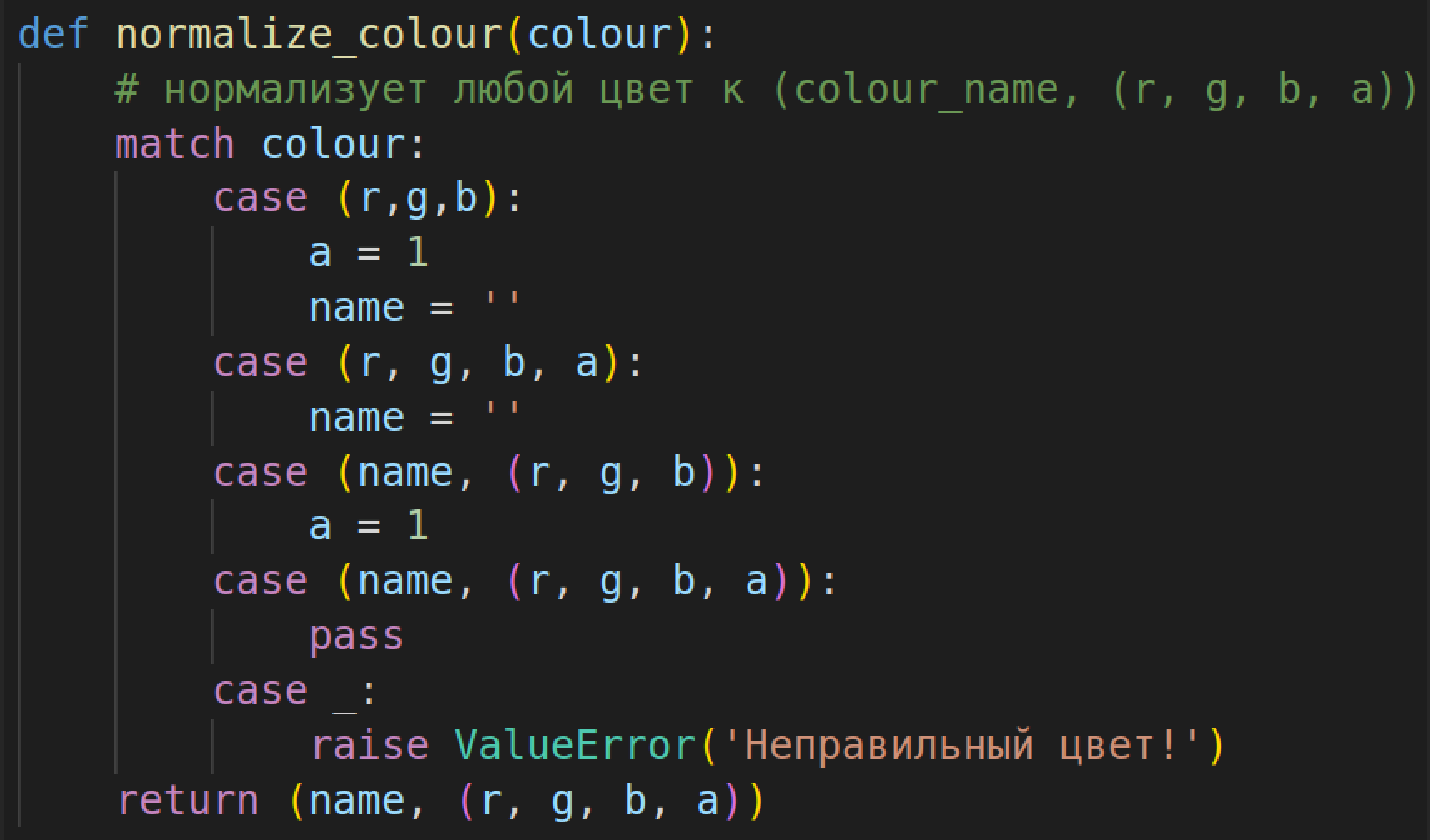
```python
def factorial(n: int) -> int:
    match n:
        case 0|1:
            return 1
        case _:
            return n * factorial(n - 1)
```



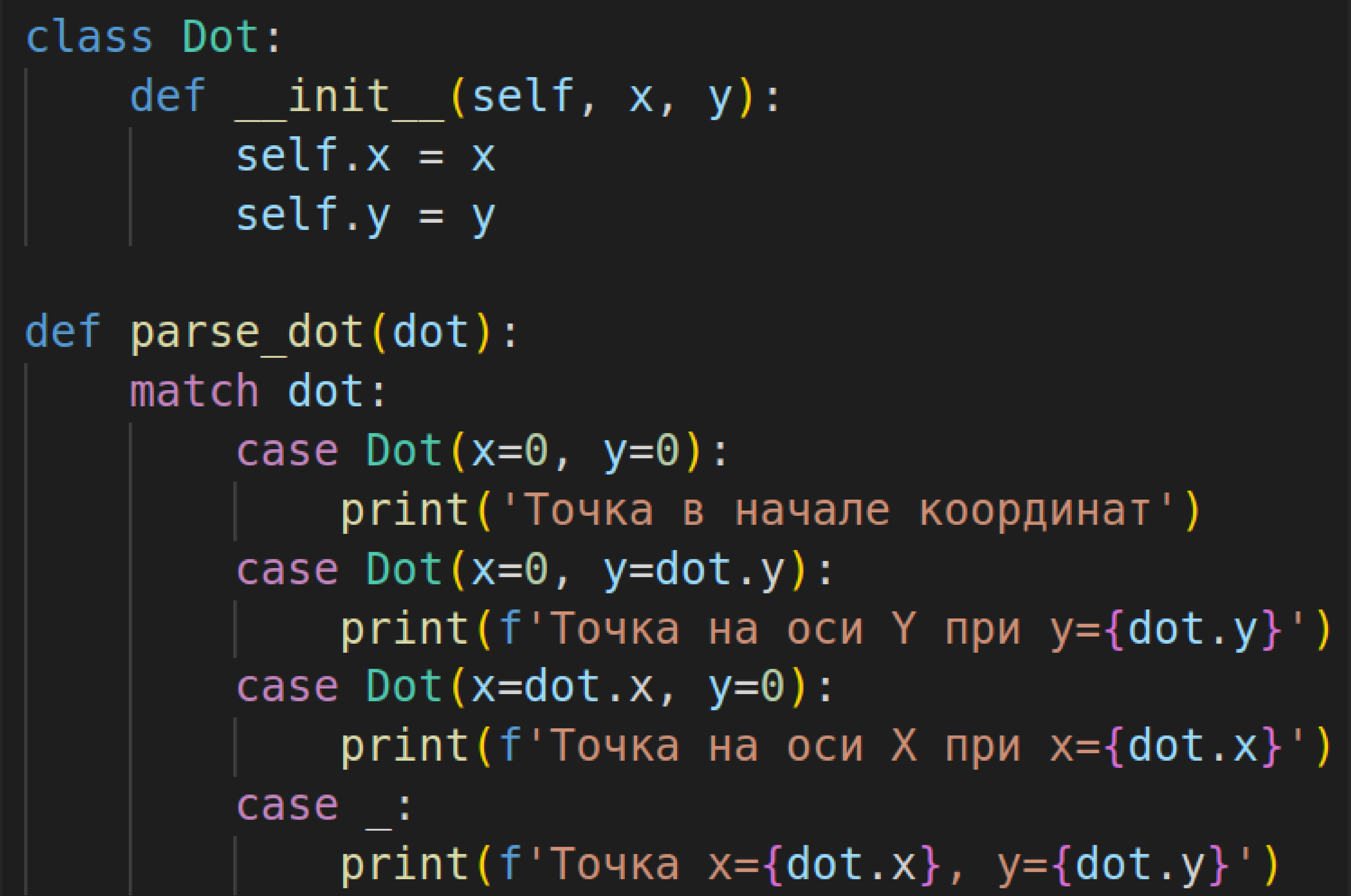
```python
def parse_command(command):
    match command.split():
        case 'Иди', 'Вверх' | 'Вниз' | 'Влево' | 'Вправо':
            print('Уже иду (всего одно направление)!')
        case 'Иди', *directions:
            print(f'Иду по направлениям: {directions}')
        case _:
            print('Таких комманд не знаю')
```

```python
commands = ['Иди Вверх', 'Иди Вверх Вправо', 'Вернись']
for cmd in commands:
    parse_command(cmd)
```

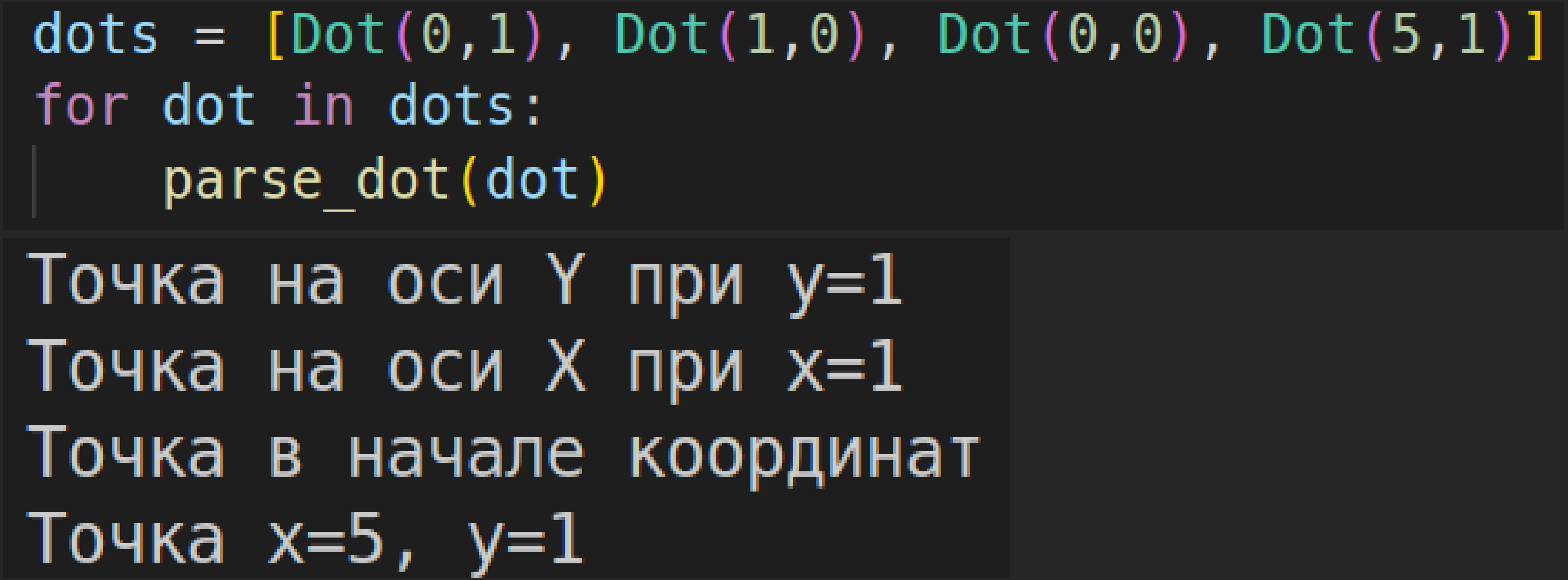
```
Уже иду (всего одно направление)!
Иду по направлениям: ['Вверх', 'Вправо']
Таких комманд не знаю
```



```python
def normalize_colour(colour):
    # нормализует любой цвет к (colour_name, (r, g, b, a))
    match colour:
        case (r,g,b):
            a = 1
            name = ''
        case (r, g, b, a):
            name = ''
        case (name, (r, g, b)):
            a = 1
        case (name, (r, g, b, a)):
            pass
        case _:
            raise ValueError('Неправильный цвет!')
    return (name, (r, g, b, a))
```



```python
class Dot:
    def __init__(self, x, y):
        self.x = x
        self.y = y

def parse_dot(dot):
    match dot:
        case Dot(x=0, y=0):
            print('Точка в начале координат')
        case Dot(x=0, y=dot.y):
            print(f'Точка на оси Y при y={dot.y}')
        case Dot(x=dot.x, y=0):
            print(f'Точка на оси X при x={dot.x}')
        case _:
            print(f'Точка x={dot.x}, y={dot.y}')
```

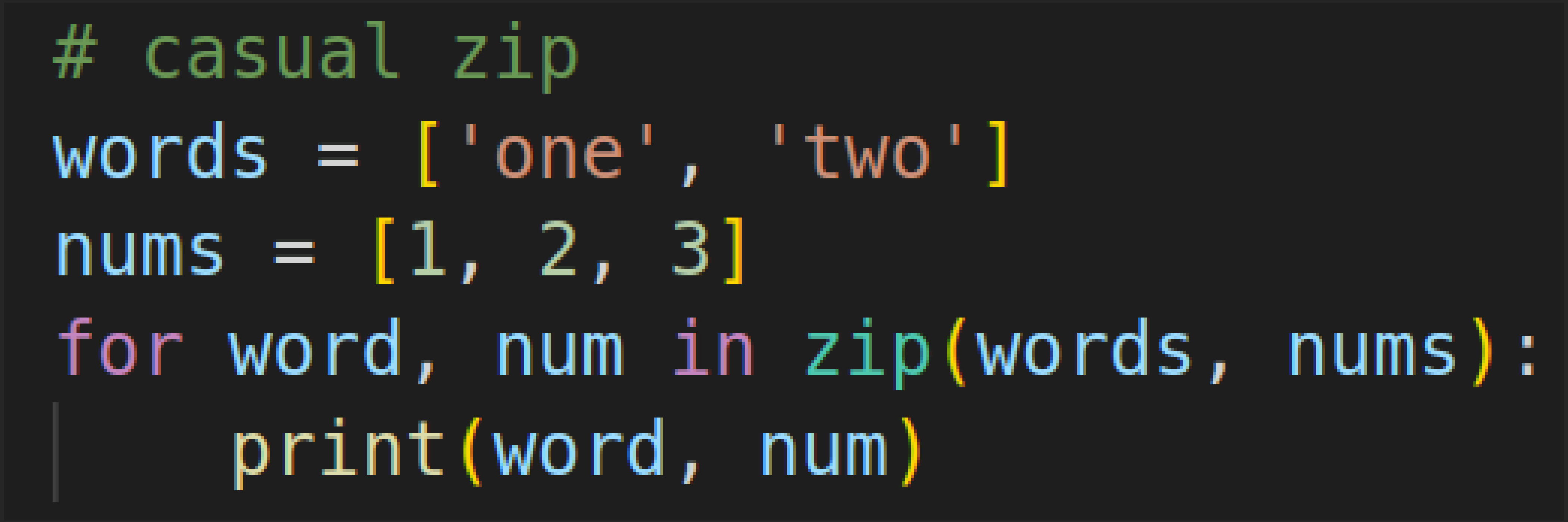


```python
dots = [Dot(0,1), Dot(1,0), Dot(0,0), Dot(5,1)]
for dot in dots:
    parse_dot(dot)
```

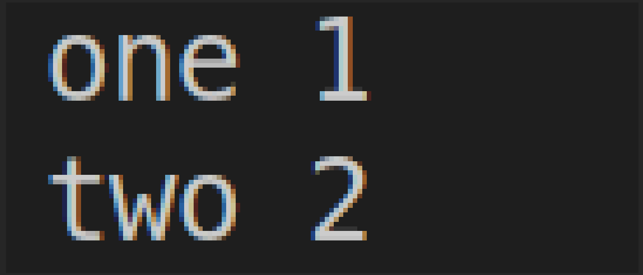
```
Точка на оси Y при y=1
Точка на оси X при x=1
Точка в начале координат
Точка x=5, y=1
```

Pattern matching: https://peps.python.org/pep-0636/

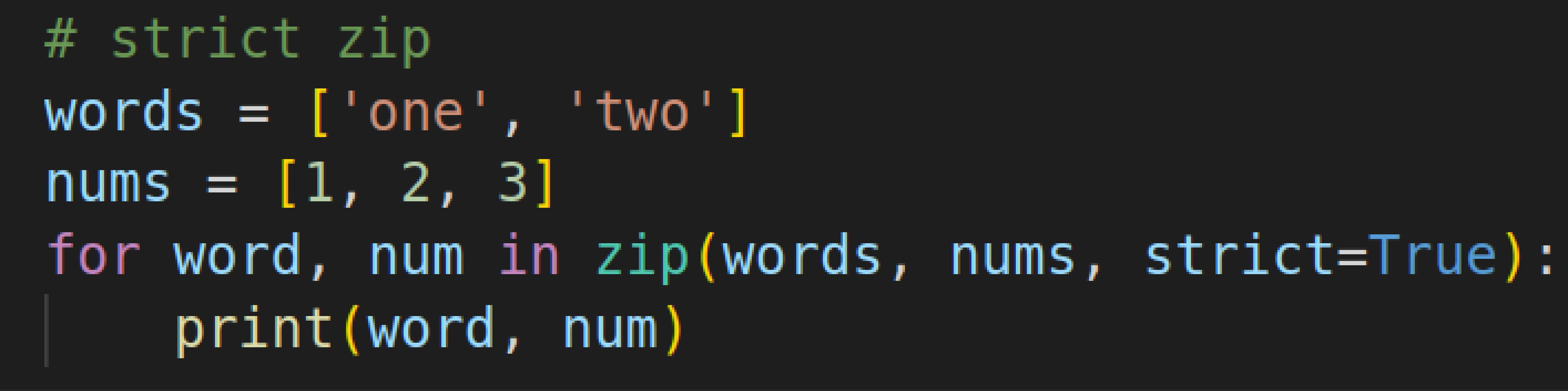
Строгий zip



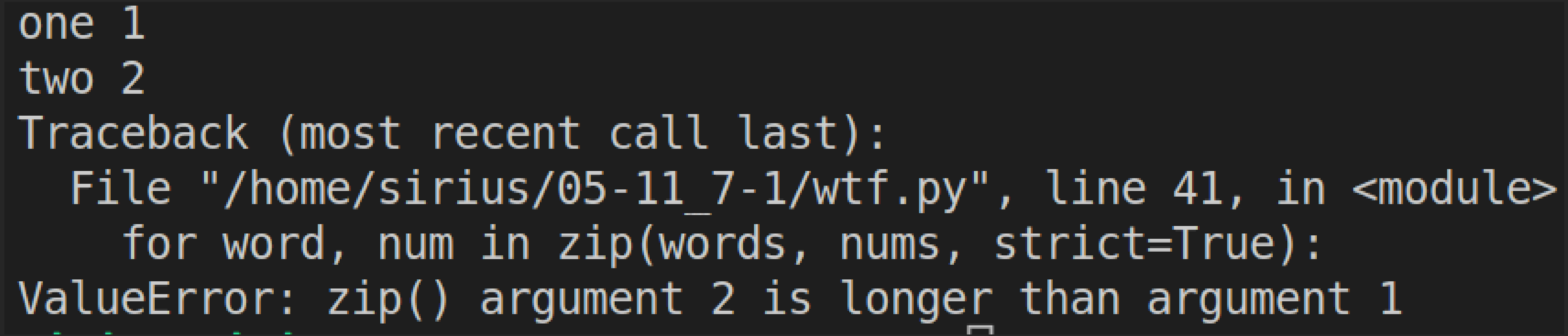
```python
# casual zip
words = ['one', 'two']
nums = [1, 2, 3]
for word, num in zip(words, nums):
    print(word, num)
```



```
one 1
two 2
```

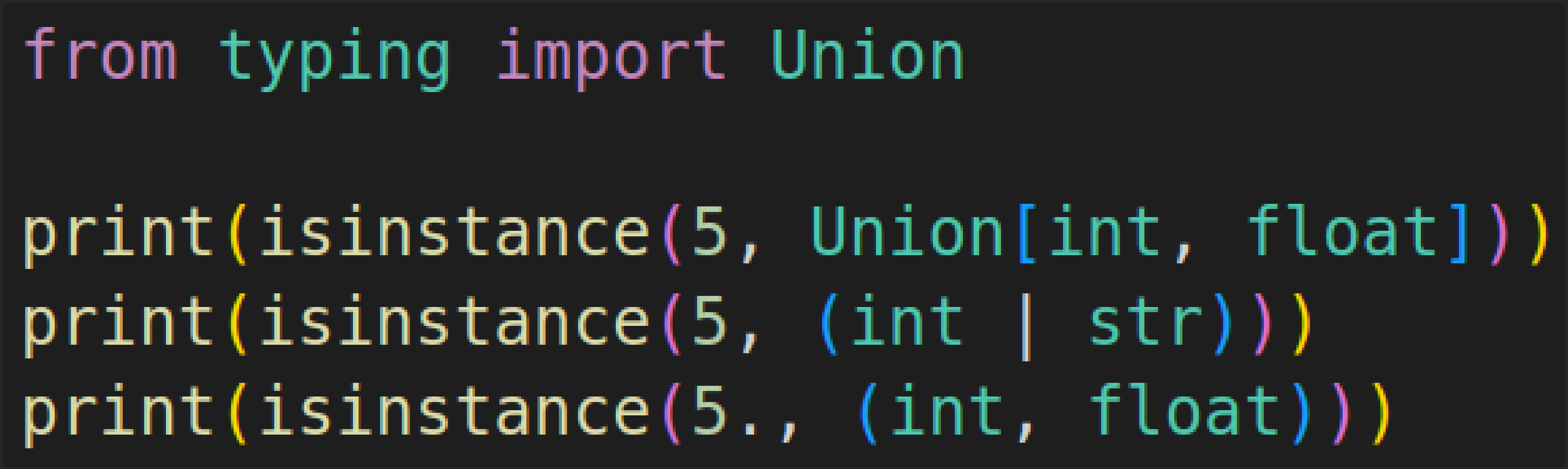


```python
# strict zip
words = ['one', 'two']
nums = [1, 2, 3]
for word, num in zip(words, nums, strict=True):
    print(word, num)
```



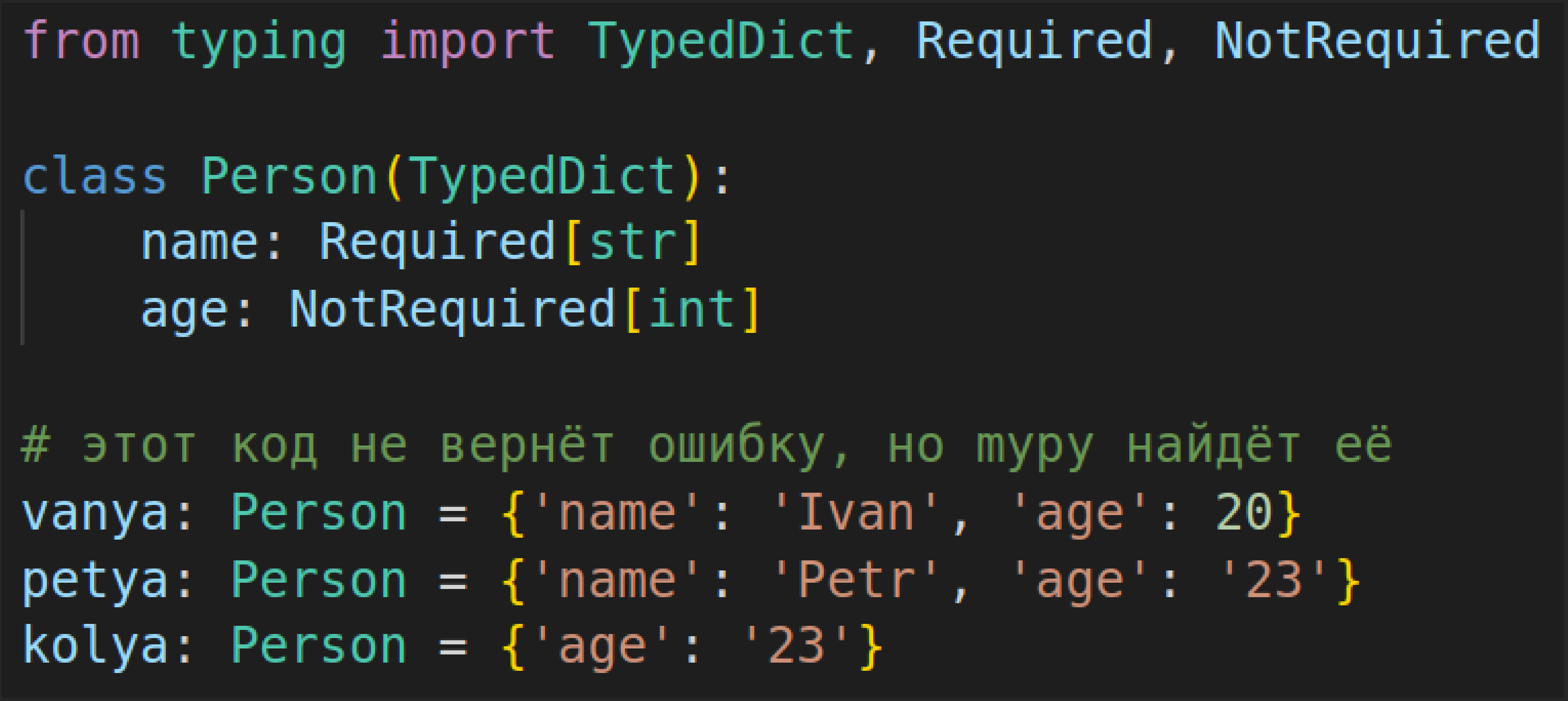
```
one 1
two 2
Traceback (most recent call last):
  File "/home/sirius/05-11_7-1/wtf.py", line 41, in <module>
    for word, num in zip(words, nums, strict=True):
ValueError: zip() argument 2 is longer than argument 1
```

Объединения типов

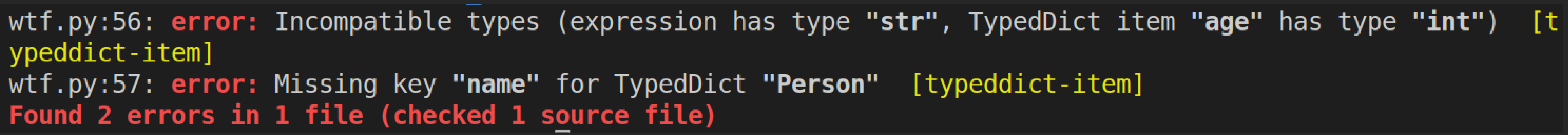


```python
from typing import Union

print(isinstance(5, Union[int, float]))
print(isinstance(5, (int | str)))
print(isinstance(5., (int, float)))
```

3.11: TypedDict

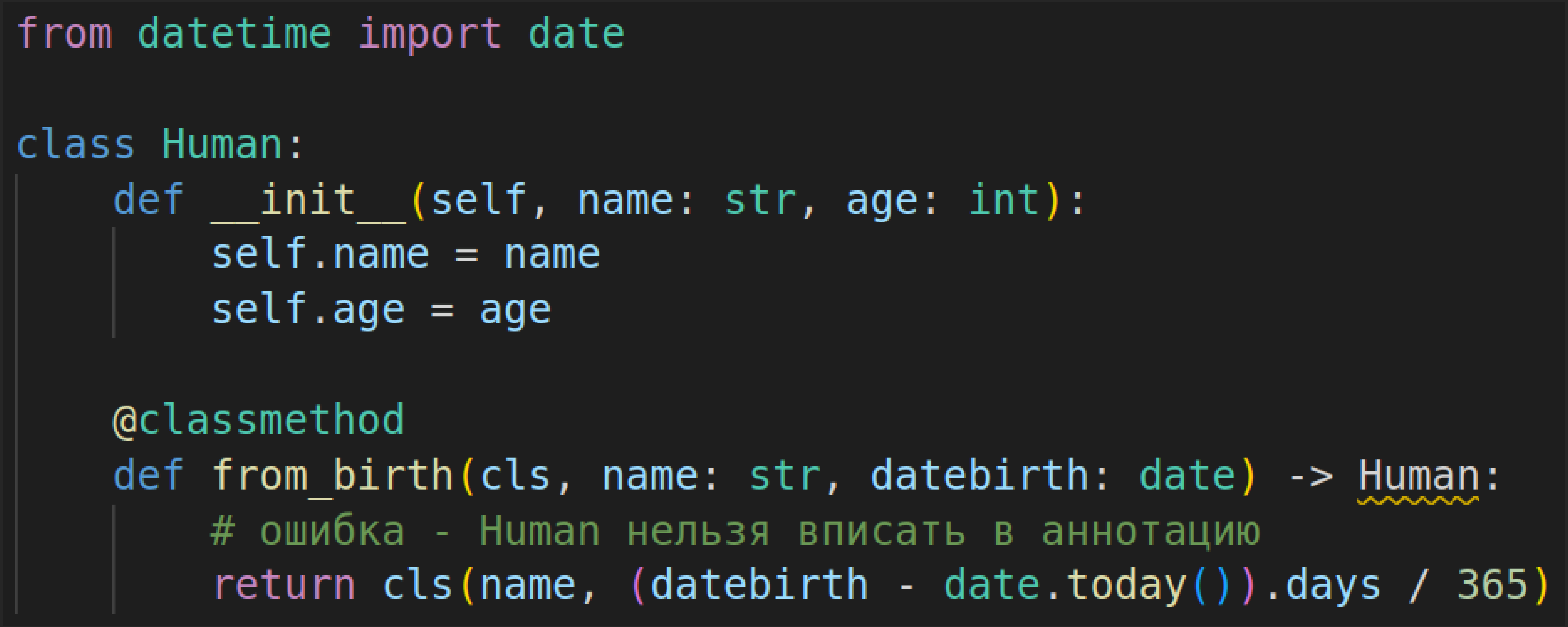


```python
from typing import TypedDict, Required, NotRequired

class Person(TypedDict):
    name: Required[str]
    age: NotRequired[int]

# этот код не вернёт ошибку, но mypy найдёт её
vanya: Person = {'name': 'Ivan', 'age': 20}
petya: Person = {'name': 'Petr', 'age': '23'}
kolya: Person = {'age': '23'}
```

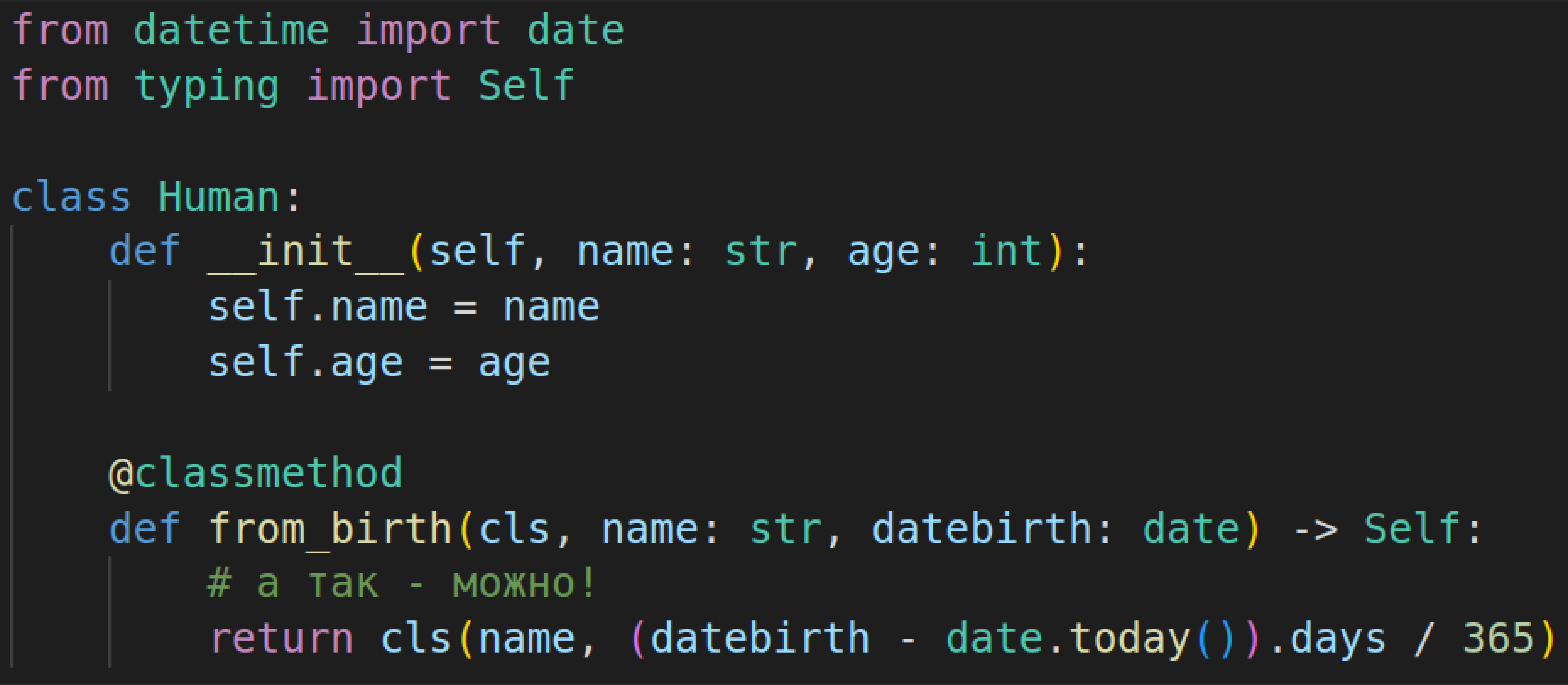


```
wtf.py:56: error: Incompatible types (expression has type "str", TypedDict item "age" has type "int")  [t
ypeddict-item]
wtf.py:57: error: Missing key "name" for TypedDict "Person"  [typeddict-item]
Found 2 errors in 1 file (checked 1 source file)
```

Self

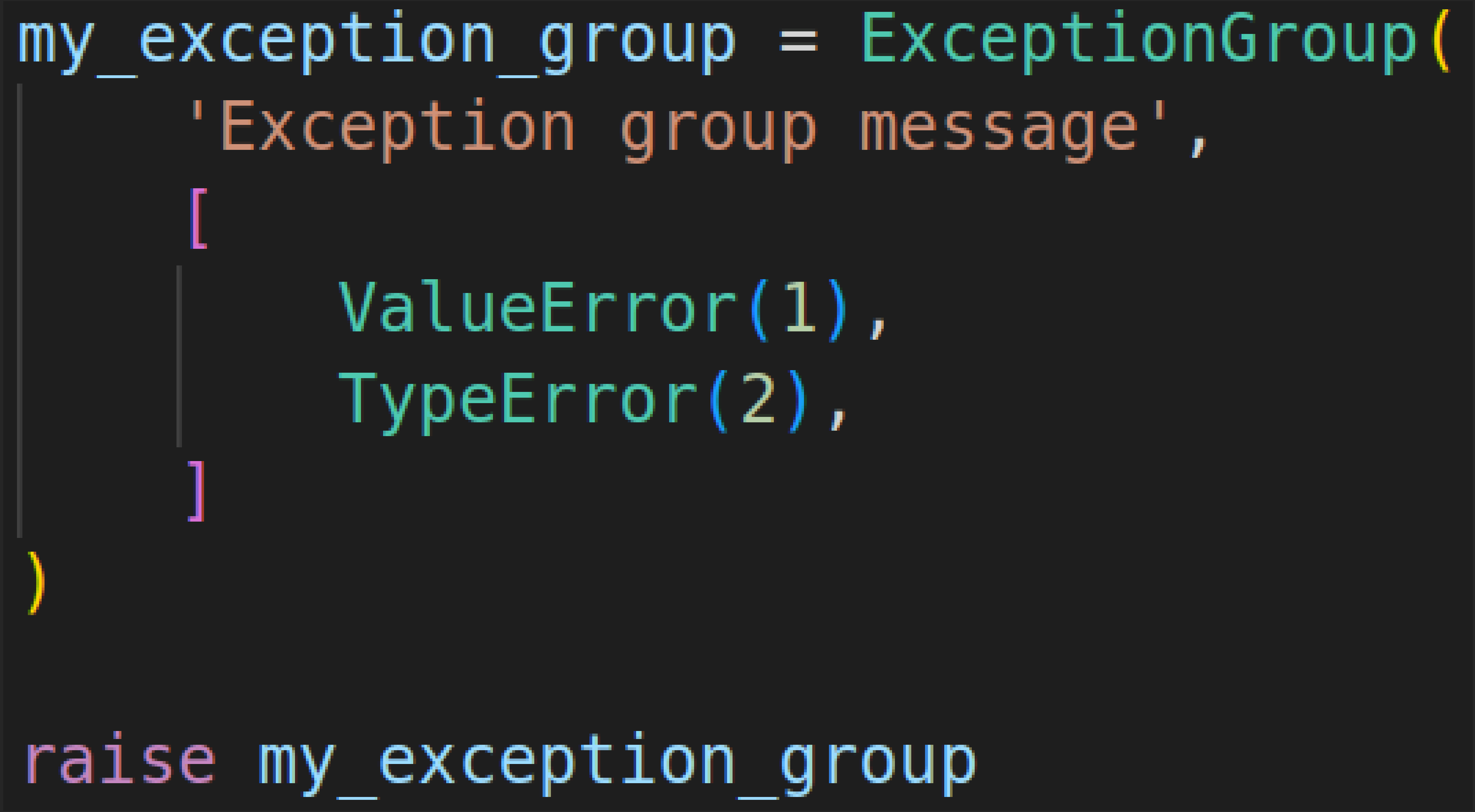


```python
from datetime import date

class Human:
    def __init__(self, name: str, age: int):
        self.name = name
        self.age = age

    @classmethod
    def from_birth(cls, name: str, datebirth: date) -> Human:
        # ошибка - Human нельзя вписать в аннотацию
        return cls(name, (datebirth - date.today()).days / 365)
```



```python
from datetime import date
from typing import Self

class Human:
    def __init__(self, name: str, age: int):
        self.name = name
        self.age = age

    @classmethod
    def from_birth(cls, name: str, datebirth: date) -> Self:
        # а так - можно!
        return cls(name, (datebirth - date.today()).days / 365)
```

Python 3.11: Exception groups (группы исключений)

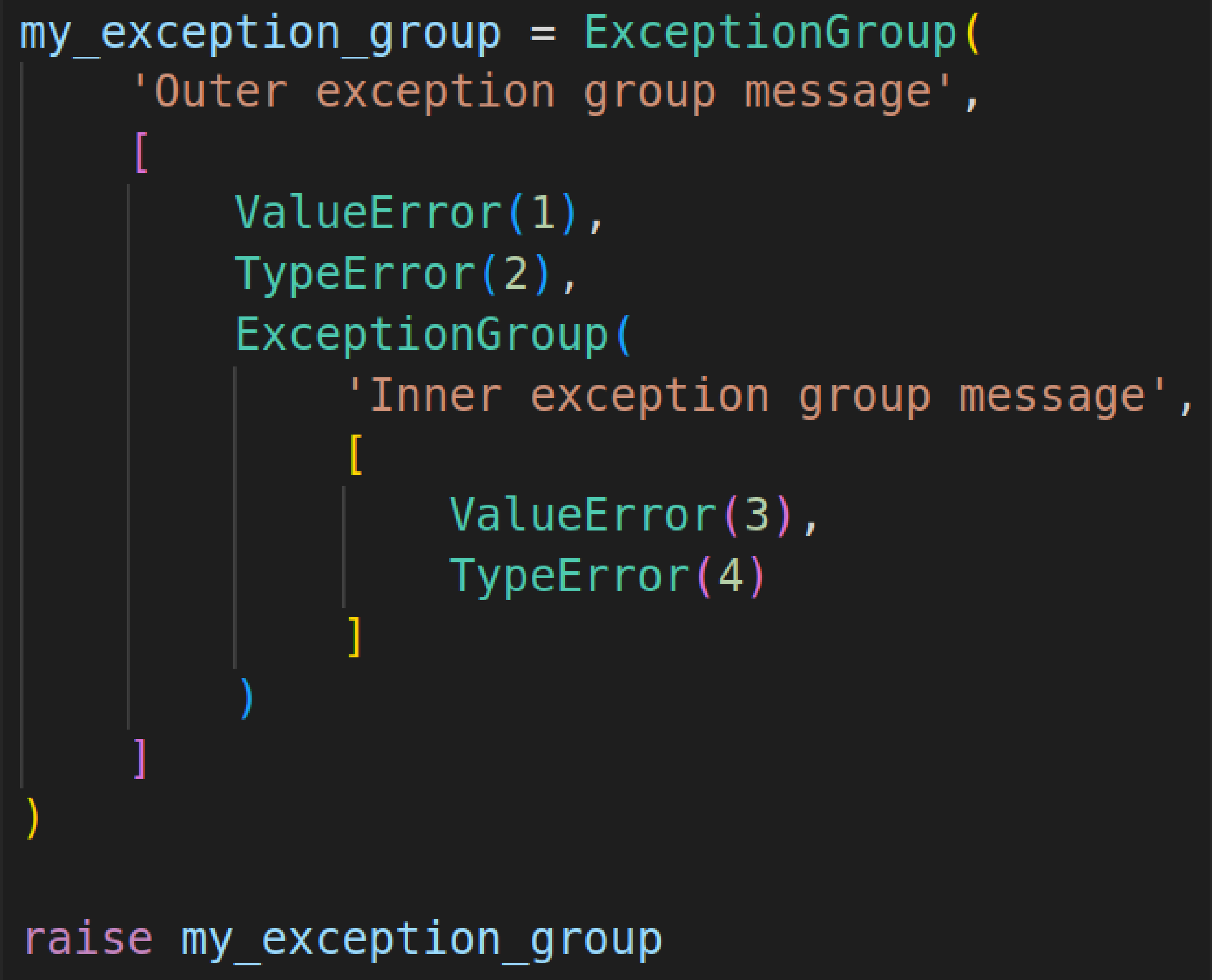


```python
my_exception_group = ExceptionGroup(
        'Exception group message',
        [
            ValueError(1),
            TypeError(2),
        ]
)

raise my_exception_group
```

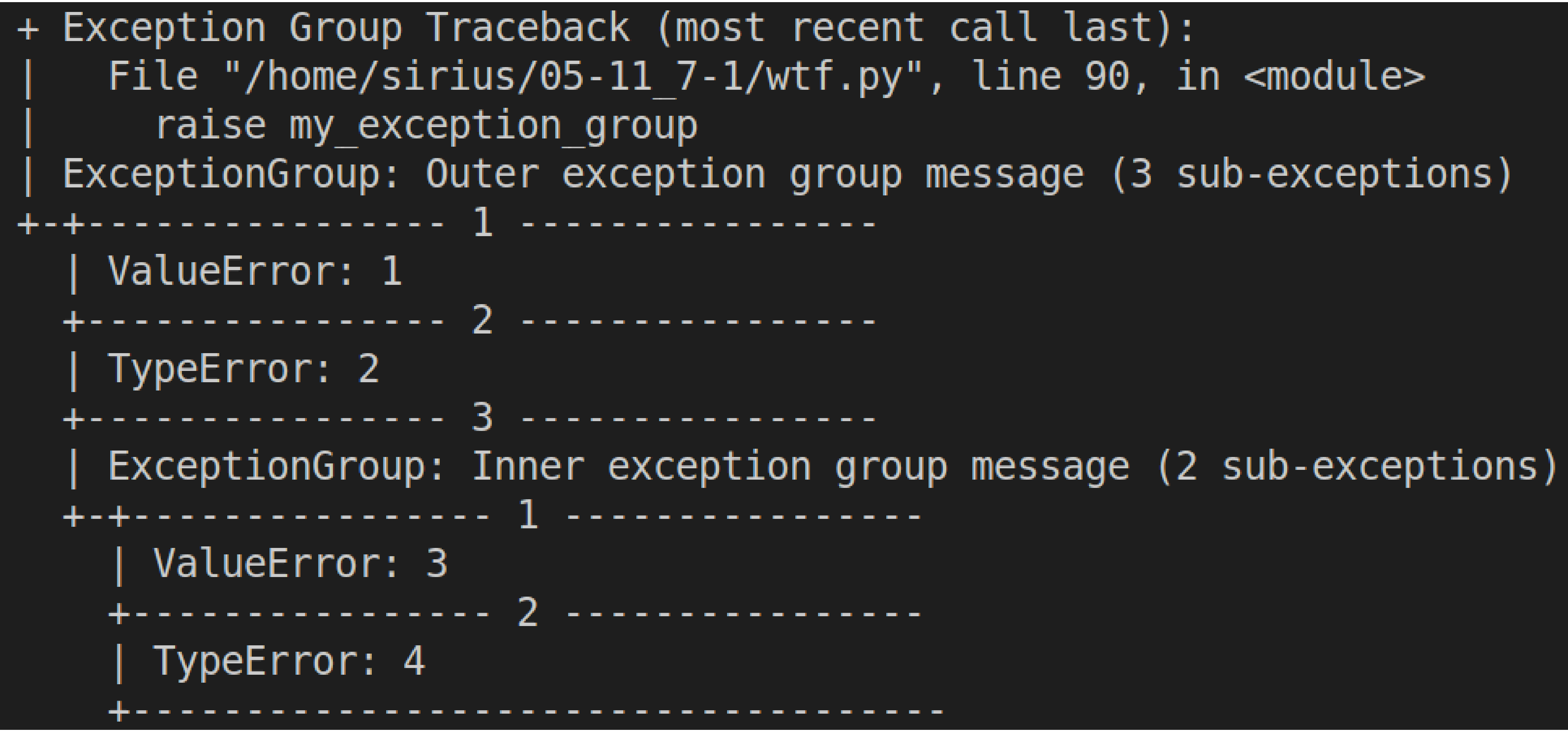
```
+ Exception Group Traceback (most recent call last):
|   File "/home/sirius/05-11_7-1/wtf.py", line 83, in <module>
|     raise my_exception_group
| ExceptionGroup: Exception group message (2 sub-exceptions)
+-+---------------- 1 ----------------
  | ValueError: 1
  +---------------- 2 ----------------
  | TypeError: 2
  +------------------------------------
```

```python
raise ExceptionGroup('Exception group message', [TypeError(1), ValueError(2)])
```

```
+ Exception Group Traceback (most recent call last):
|   File "/home/sirius/05-11_7-1/wtf.py", line 89, in <module>
|     raise ExceptionGroup('Exception group message', [TypeError(1), ValueError(2)])
| ExceptionGroup: Exception group message (2 sub-exceptions)
+-+---------------- 1 ----------------
  | TypeError: 1
  +---------------- 2 ----------------
  | ValueError: 2
  +------------------------------------
```



```python
my_exception_group = ExceptionGroup(
    'Outer exception group message',
    [
        ValueError(1),
        TypeError(2),
        ExceptionGroup(
            'Inner exception group message',
            [
                ValueError(3),
                TypeError(4)
            ]
        )
    ]
)

raise my_exception_group
```



```
+ Exception Group Traceback (most recent call last):
|   File "/home/sirius/05-11_7-1/wtf.py", line 90, in <module>
|     raise my_exception_group
| ExceptionGroup: Outer exception group message (3 sub-exceptions)
+-+---------------- 1 ----------------
  | ValueError: 1
  +---------------- 2 ----------------
  | TypeError: 2
  +---------------- 3 ----------------
  | ExceptionGroup: Inner exception group message (2 sub-exceptions)
  +-+---------------- 1 ----------------
    | ValueError: 3
    +---------------- 2 ----------------
    | TypeError: 4
    +------------------------------------
```