Python-разработчик

Исключения

skypro

Сегодня на уроке

План урока

1. Познакомимся с исключениями и узнаем, для чего используются блоки try/except.
2. Увидим, как распространяются исключения.
3. Напишем свои исключения.

Исключения. Блоки try/except

Синтаксические ошибки — появляются в результате нарушения синтаксиса языка при написании исходного кода.

определение

Ошибки выполнения (исключения) — появляются в процессе выполнения программы.

определение

Пример исключения

Файл был удален, пока программа работала.
Попытка чтения несуществующего файла приведет к возникновению исключения.

Блок try/except

```
try:
  # попытка выполнить код программы
except ОбрабатываемоеИсключение:
  # код, который будет выполняться в случае
  # возникновения ошибки в коде программы
```

Исключения. Блоки try/except

Исключение	Причина	Пример
RecursionError	Ошибка рекурсии	def recursion(): return recursion()
TypeError	Ошибка типа	2 + '2'
OverflowError	Ошибка переполнения	math.exp(1000)
AssertionError	Ошибка утверждения	a, b = 1, 'a'; assert a == b

Исключения. Блоки try/except

Исключение	Причина	Пример
AttributeError	Ошибка атрибута	Попытка сослаться на несуществующий атрибут
NameError	Ошибка имени	Попытка использовать несуществующее имя переменной
ZeroDivisioError	Ошибка деления на ноль	1 / 0
FileNotFoundError	Ошибка отсутствия файла	Попытка открыть несуществующий файл

Задача

Спровоцировать и обработать исключения разных типов:

- NameError
- ZeroDivisionError
- FileNotFoundError

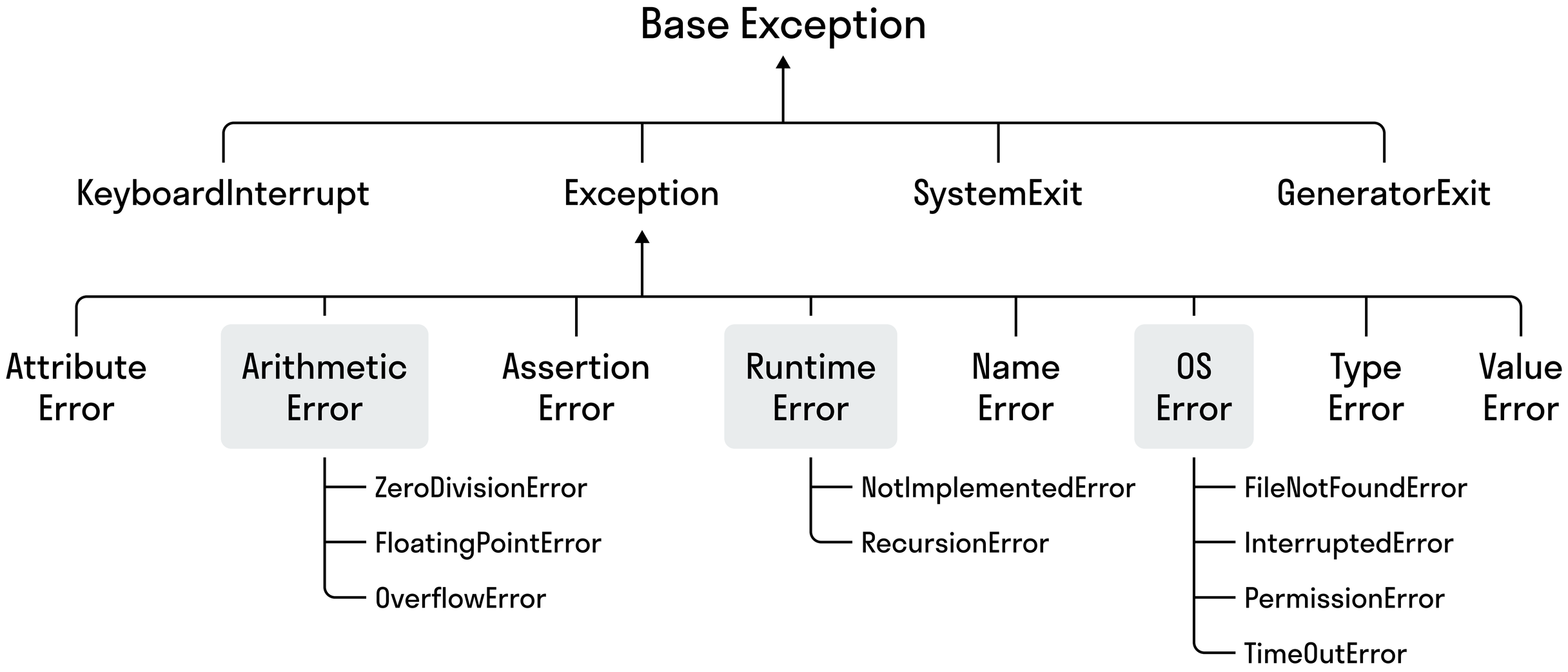
Задача

Типы исключений.

Флоу решения:

1. `print(a)`
2. `1/0`
3. `open('unknown_file')`

Иерархия исключений



Полная форма try/except

```
try:
    print('Основной код.')
except:
    print('Код, если возникло исключение.')
else:
    print('Код, если не возникло исключений.')
finally:
    print('Код, который выполняется всегда.')
```

Задача

Принять на вход от пользователя два числа a и b.
Разделить a на b. Убедиться, что пользователь ввел числа
и эти числа целые. Число b должно быть отличным от нуля.

Задача

Флоу решения:

1. Получить данные от пользователя
2. Обработать исключение ошибки ввода
3. Обработать исключение деления на ноль
4. Добавить блоки else и finally
для вывода сообщений

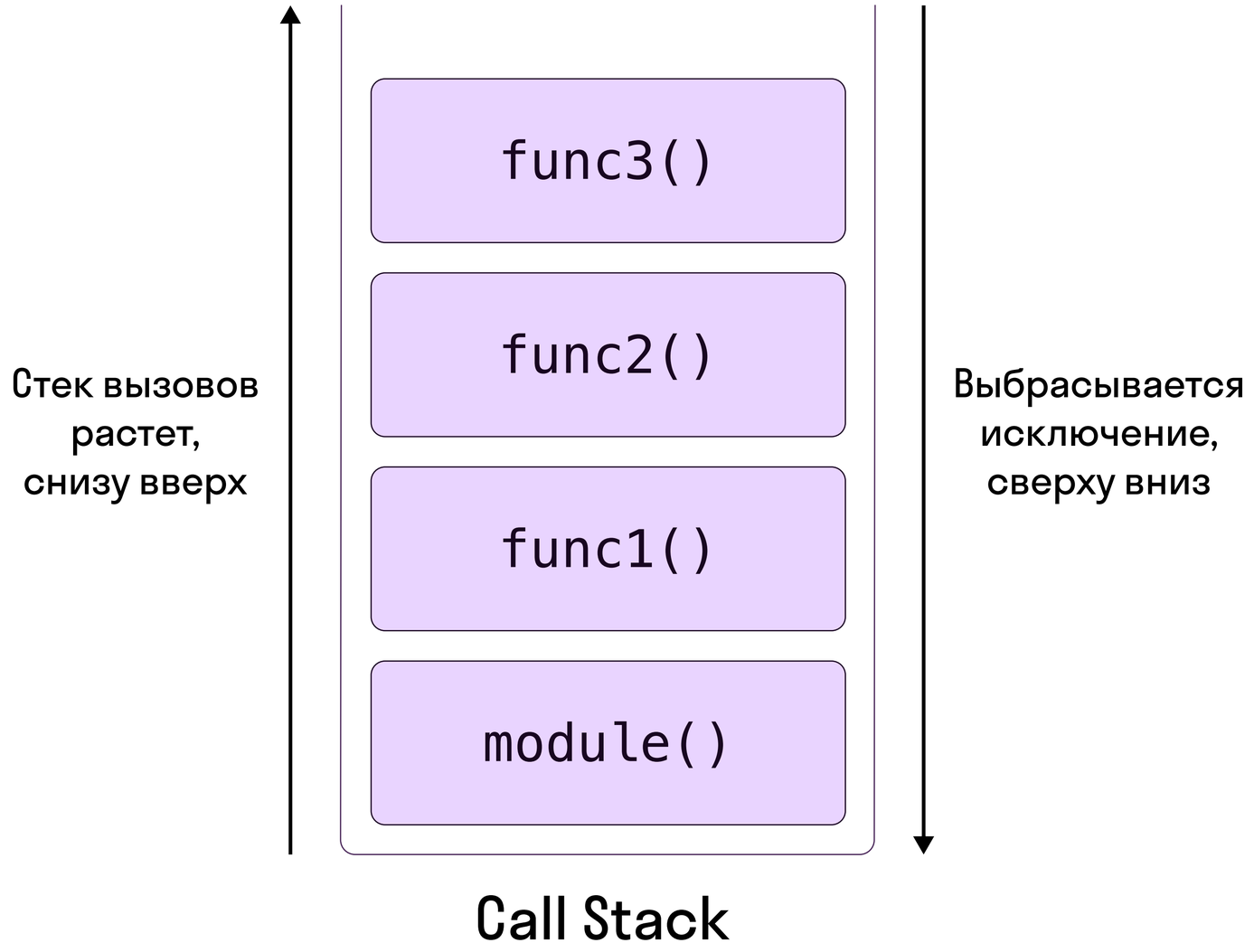
Отслеживание ошибок

Traceback (трассировка, трэйс) —
отчет, который показывает стек вызовов, где появилась ошибка. Каждое исключение содержит краткую информацию и полный путь до места ошибки.

определение

Распространение исключений

При возникновении ошибки появляется сообщение и полный стек вызовов до точного места возникновения исключения.



Задача

Написать класс с тремя методами, в одном из которых возникает исключение. Сделать обработку исключения на разных уровнях.

Задача

Перехват исключения.

Флоу решения:

1. Реализовать класс
2. Прописать методы
3. Прописать блоки try/except

Пользовательские исключения

С помощью инструкции raise можно вызвать:

- встроенные исключения,
- самостоятельно реализованные классы исключений.

Необходимо наследоваться от класса Exception.

важно запомнить

Пример

```
class MyException(Exception):
    """Пользовательский класс исключения"""

    def __init__(self, *args, **kwargs):
        self.message = args[0] if args else 'Неизвестная ошибка.'

    def __str__(self):
        return self.message
```

Задача

Реализовать класс для работы со скриптами.
Реализовать свои классы исключений для обработки исключений, связанных с шелл-скриптами.

Shebang & Shell Script

Шебанг (shebang) — последовательность символов #! , которая указывает операционной системе, какую программу использовать для анализа остальной части файла.

определение

Шелл-скрипт (Shell Script) — сценарий командной строки, или командной оболочки, — программа, выполняемая командной оболочкой операционной системы.

определение

Пример шебанга: `#!/bin/bash`

Задача

Пользовательские исключения.

Флоу решения:

1. Создать класс исключения
2. Наследоваться от Exception
3. Прописать инициализатор
для задания сообщения по умолчанию

Задача

Для магического метода `__add__` класса Employee добавить обработку исключений при передаче в метод значений, которые должны быть объектами этого же класса или числом.
Написать тесты на обработку исключений.

Задача

Флоу решения:

1. Восстановить класс Employee
2. Написать проверку значений и возбуждать исключение
3. Написать тесты на обработку возникающих исключений

Подведем итоги

Итоги урока

1. Исключения возникают при ошибках в процессе выполнения программ.

Итоги урока

1. Исключения возникают при ошибках в процессе выполнения программ.
2. Есть различные типы исключений, у них есть своя иерархия.

Итоги урока

1. Исключения возникают при ошибках в процессе выполнения программ.
2. Есть различные типы исключений, у них есть своя иерархия.
3. Исключения можно обрабатывать в блоке try/except и вызывать с помощью инструкции raise.

Итоги урока

1. Исключения возникают при ошибках в процессе выполнения программ.
2. Есть различные типы исключений, у них есть своя иерархия.
3. Исключения можно обрабатывать в блоке try/except и вызывать с помощью инструкции raise.
4. Можно создавать собственные классы исключения, унаследовавшись от существующих.

Спасибо!

skypro