

УРОК 32. МАГИЧЕСКИЕ МЕТОДЫ КЛАССОВ

ЧТО ТАКОЕ МАГИЧЕСКИЕ МЕТОДЫ КЛАССОВ	2
ДЕКОРАТОР <code>@FUNCTIONAL.TOTAL_ORDERING</code>	3
ИСПОЛЬЗОВАНИЕ МЕТОДОВ	4
ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ	5
ПРАКТИЧЕСКАЯ РАБОТА	8
ПОЛЕЗНЫЕ МАТЕРИАЛЫ	9



ЧТО ТАКОЕ МАГИЧЕСКИЕ МЕТОДЫ КЛАССОВ

Магические методы в Python представляют специальные методы с особым именем, начинающимся и заканчивающимся на два подчеркивания (`__`) - например, `__init__`, `__add__`, `__eq__`, и т.д. Эти методы позволяют определить поведение объектов при выполнении определенных операций, таких как математические операции, сравнения, преобразования и другие.

Методы математических операций определяют поведение объекта при выполнении арифметических операций, таких как сложение (`__add__`), вычитание (`__sub__`), умножение (`__mul__`), деление (`__div__`) и другие. Префикс "r" (например, `__radd__`) указывает на правостороннюю операцию.



ДЕКОРАТОР @FUNCTIONAL.TOTAL_ORDERING

Декоратор `@functools.total_ordering` позволяет автоматически генерировать недостающие методы сравнения (`__lt__`, `__le__`, `__gt__`, `__ge__`) на основе метода `__eq__` и одного из методов `__lt__` или `__gt__`. Это упрощает реализацию полного набора операций сравнения для класса.



ИСПОЛЬЗОВАНИЕ МЕТОДОВ

Метод `__len__` определяет поведение объекта при вызове встроенной функции `len()` и возвращает размер или длину объекта.

Метод `__hash__` определяет хэш-значение объекта, которое используется, например, при добавлении объекта в множество или использовании в качестве ключа в словаре.

Метод `__bool__` определяет логическое значение объекта при вызове встроенной функции `bool()`.

Методы `__getitem__` и `__setitem__` определяют поведение объекта при доступе к элементам через индексацию, подобно словарю. `__getitem__` позволяет получить значение по ключу, а `__setitem__` - присвоить значение по ключу.

Метод `__delitem__` определяет поведение при удалении элемента по ключу.

Метод `__call__` позволяет вызывать экземпляр класса как функцию. Это полезно, когда класс представляет собой вызываемый объект и нужно определить логику вызова.

Классы с методом `__call__` могут быть вызваны с использованием круглых скобок, как функции:

Python

```
class CallableClass:
    def __call__(self, *args, **kwargs):
        # Логика вызова класса

obj = CallableClass()
obj() # Вызов класса как функции
```



ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Объясните, что происходит в этом коде:

Python

```
class Test:
    def __call__(self, message):
        print(message)
        return True
```

```
test = Test()
test("Hello World")
'Hello World'
True
```



Метод `__iter__` определяет, как объект должен быть перебираемым, и возвращает итератор. Метод `__next__` определяет логику получения следующего элемента из итератора.

```
Python
class MyIterator:
    def __iter__(self):
        return self

    def __next__(self):
        # Логика получения следующего элемента
        # Если достигнут конец итерации, вызывается исключение StopIteration
        raise StopIteration

obj = MyIterator()
for item in obj:
    # Логика перебора элементов
    pass
```

Метод `__dict__` возвращает словарь, содержащий атрибуты объекта.

```
Python
class MyClass:
    def __init__(self):
        self.x = 1
        self.y = 2

obj = MyClass()
print(obj.__dict__) # {'x': 1, 'y': 2}
```

Для добавления магических методов в класс нужно определить их внутри класса. Например:

```
Python
class MyClass:
    def __init__(self, x):
        self.x = x
```



```
def __str__(self):  
    return f"MyClass with x = {self.x}"  
  
obj = MyClass(10)  
print(obj) # MyClass with x = 10
```



ПРАКТИЧЕСКАЯ РАБОТА

Дополнить класс `Person` операцией сложения. При сложении двух объектов этого класса операция должна возвращать новый объект класса `Person`, у которого рост - четверть от среднего арифметического роста родителей, вес - двадцатая часть от среднего арифметического веса родителей, год рождения - текущий.

Дополнительно придумать и определить методы `_len_`, `_bool_`.



ПОЛЕЗНЫЕ МАТЕРИАЛЫ

1. [Магические методы в Python — подробный гайд](#)
2. [Руководство по магическим методам в Питоне](#)