

ООП: объекты и классы | Я.Шпора

Основные понятия

Классы — «чертежи», шаблоны для создания объектов.

Экземпляры класса — конкретные объекты, у которых есть все методы и атрибуты класса.

Атрибуты класса — переменные, которые объявлены внутри класса и относятся ко всем экземплярам этого класса.

Атрибуты объекта — переменные, связанные с конкретным экземпляром класса.

Методы — части класса, функции, которые отвечают за поведение объектов или самого класса.

Специальные, или магические, атрибуты и методы — «настройки по умолчанию»; эти методы и атрибуты, как правило, сообщают Python, как обрабатывать объекты и классы в определённых ситуациях

Классы и объекты в Python

Почти всё в Python — классы или объекты какого-то класса. Все типы данных, например `str` и `int` — это классы, а конкретные строки, числа, списки, словари — объекты этих классов.

Шаблон объявления собственного класса

```
class ИмяКласса:
    # Атрибуты класса (переменные, относящиеся к классу).
    атрибут1 = значение1
    атрибут2 = значение2

    # Методы объектов.
    def __init__(self, параметр1, параметр2):
        # Инициализатор класса, вызывается при создании экземпляра.
        self.параметр1 = параметр1
        self.параметр2 = параметр2
```

```
def метод1(self):
    # Код метода 1.
    pass

def метод2(self, аргумент):
    # Код метода 2.
    pass

# Создание экземпляра класса.
экземпляр = ИмяКласса(значение_аргумента1, значение_аргумента2)

# Использование атрибутов и методов экземпляра.
print(экземпляр.атрибут1)
print(экземпляр.метод1())
```

Магический метод `__init__` инициализирует уже созданный экземпляр класса, задавая начальные значения его атрибутов.

`self` — первый аргумент при определении любого метода. Этот аргумент предоставляет интерпретатору Python информацию о конкретном экземпляре. При вызове метода экземпляра `self` указывать не нужно.

Доступ к методам, атрибутам класса или объекта осуществляется через точечную нотацию.

Правила именования классов

- Пишите имя класса на английском и с большой буквы.
- Если имя класса состоит из нескольких слов, используйте *CamelCase*.
- Выбирайте для имён классов существительные.
- Выбирайте понятные и информативные имена, которые отражают назначение класса.
- Избегайте аббревиатур или неочевидных сокращений, если они не общеприняты.

- Не используйте имена, которые уже заняты встроенными типами или классами в Python.

Атрибуты класса vs атрибуты объекта

Атрибуты класса объявляются как обычные переменные, перед методами; атрибуты объекта объявляются в инициализаторе класса или в методах через `self` — указатель на текущий объект.

```
class Phone:

    # Атрибут класса.
    line_type = 'проводной'

    def __init__(self, dial_type_value):
        # Атрибут объекта.
        self.dial_type = dial_type_value
```

Если значение атрибута класса попытаться поменять через объект, то новое значение получит только этот объект.

Значения **атрибутов объекта** хранятся в самом объекте, а значения **атрибутов класса** объект получает из класса.

Магический метод `__str__`

По умолчанию есть у всех объектов — встроенных и пользовательских. Позволяет объектам «рассказывать» о себе в виде строк.

```
class Phone:
    ...
    def __str__(self):
        return f'Это {self.line_type} телефон. Набор - {self.dial_type}.'

rotary_phone = Phone(dial_type_value='дисковый')

print(rotary_phone)
```

```
# Вывод в терминал:
```

```
# Это проводной телефон. Набор - дисковый.
```

Полезные ресурсы

[Документация Python о методах и атрибутах всех встроенных типов данных](#)

Мудрец знает: шпаргалка хранит, а
человек - действует.

Аль' Горитм, персидский просветитель