Лекция 9

Язык программирования Python.

Хайрулин Сергей Сергеевич

email: s.khairulin@g.nsu.ru, s.khayrulin@gmail.com

Ссылка на материалы

План

- Лекции/практические занятия
  - Тест
- Дифференцированный зачет в конце семестра
  - Защита задания

Литература

Начальный уровень

- Mark Pilgrim. Dive into Python - http://www.diveintopython.net/
- Марк Лутц. Изучаем Python, 4-е издание // Символ-Плюс 2011.
- ...

Стандарт/Документация

- PEP-8 - https://www.python.org/dev/peps/pep-0008/
- https://www.python.org/
- https://github.com/python/cpython

Экспертный уровень

- Лучано Рамальо: Python. К вершинам мастерства
- Mitchell L. Model. Bioinformatics Programming Using Python // O'Reilly 2010.

Версии Python

- Python 2 вышел 2010 году последняя версия 2.7.16 - исправлялись только баги(ошибки) с января 2020 года поддержка прекращена.
- Python 3 в появился в 2008, является актуальной версией языка. Текущая стабильная версия 3.9, в разработке 3.10
  - Python 3 не гарантирует совместимости кода с Python 2

Summary

- Абстрактные типы данных (АТД)
- ООП
  - Наследование
  - Полиморфизм
  - Инкапсуляция
- ООП в Python
  - Объект object
  - Ключевое слово class
  - метод __init__
  - ключевое слово self
  - свойства/атрибуты класса
  - методы/функции

Абстрактные типы данных (АТД)

АТД - математическая модель с совокупностью операторов, определенных в рамках этой модели (А. Ахо).

Пример списки, графы, множества….

Graph - insert vertex, delete vertex….

Абстрактные типы данных (АТД)

В python:

Базовые типы int, float, str,....,

Комплексные типы:

list: append, del ....

dict: insert, get, del ….

…..

ООП

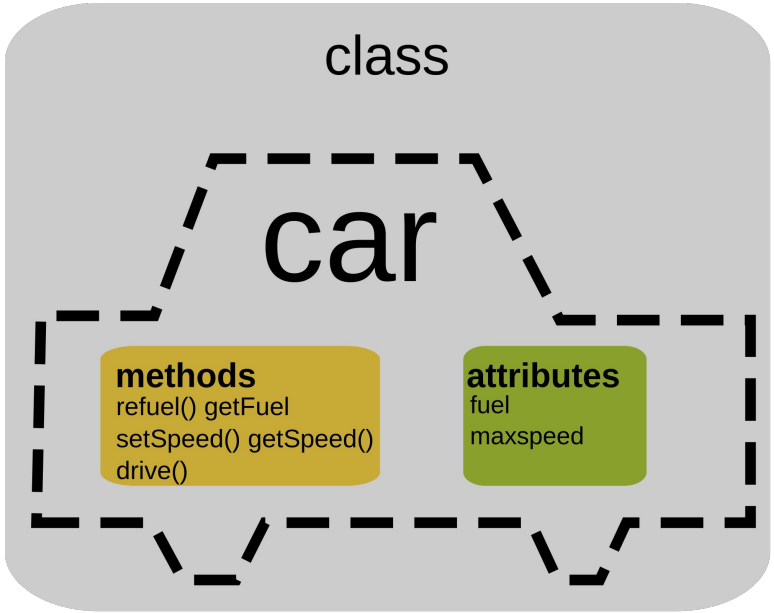
Объе́ктно-ориенти́рованное программи́рование (ООП) — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования(Г. Буч 1998)

Языки поддерживающие ООП: C++, Java, C#, Python, Perl, Go….

ООП принципы

- Наследование
  - Возможность создание новых типов данных базирующихся на других, ранее определенных
- Полиморфизм
  - Возможность переопределения поведения базовых свойств объекта (свойств унаследованных от объектов предков)
- Инкапсуляция
  - Возможность скрывать реализацию тех или иных свойств объекта от конечного пользователя

Объект



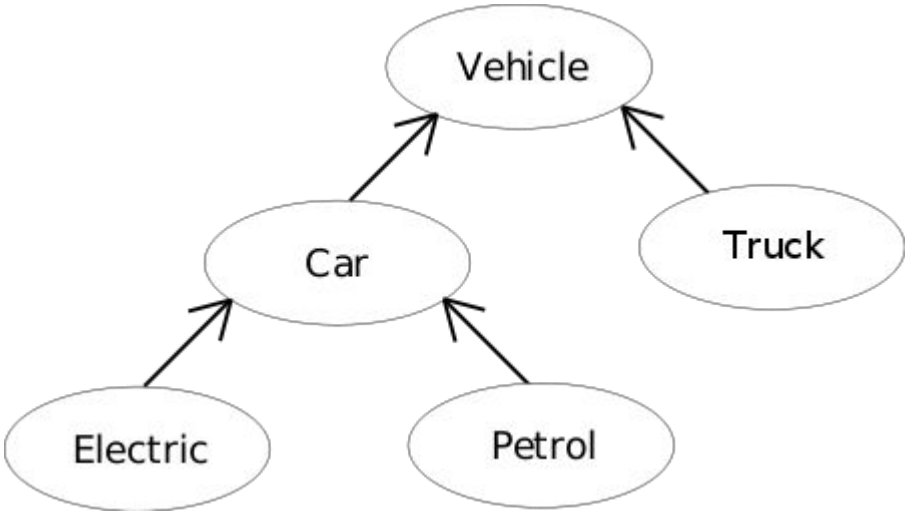
Classname
(Identifier)
Data Member
(Static attributes)
Member Functions
(Dynamic Operations)

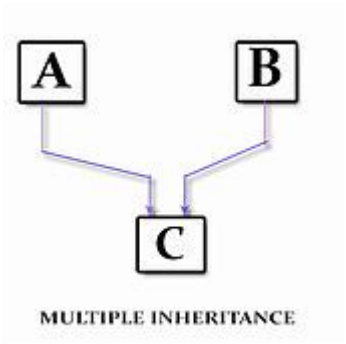
Student	Circle
name grade	radius color
getName() printGrade()	getRadius() getArea()

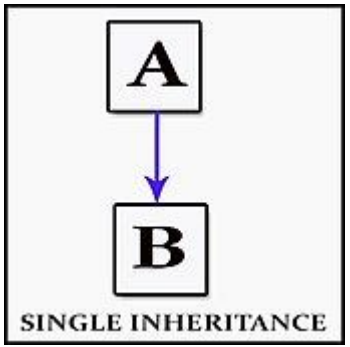
SoccerPlayer	Car
name number xLocation yLocation	plateNumber xLocation yLocation speed
run() jump() kickBall()	move() park() accelerate()

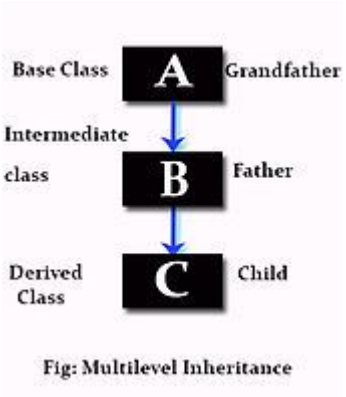
Examples of classes

Наследование

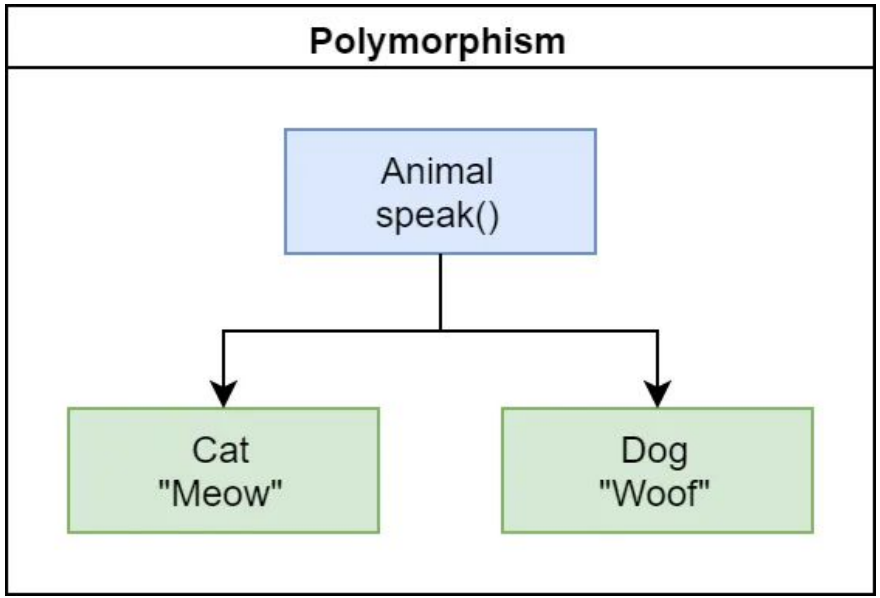


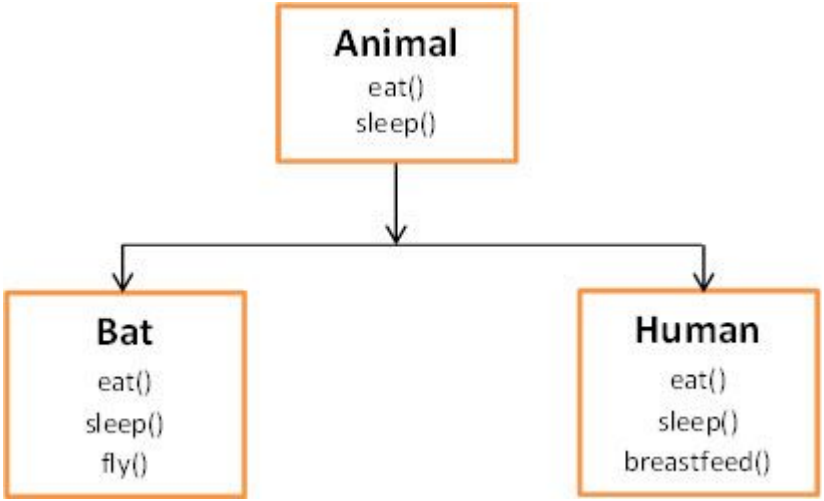




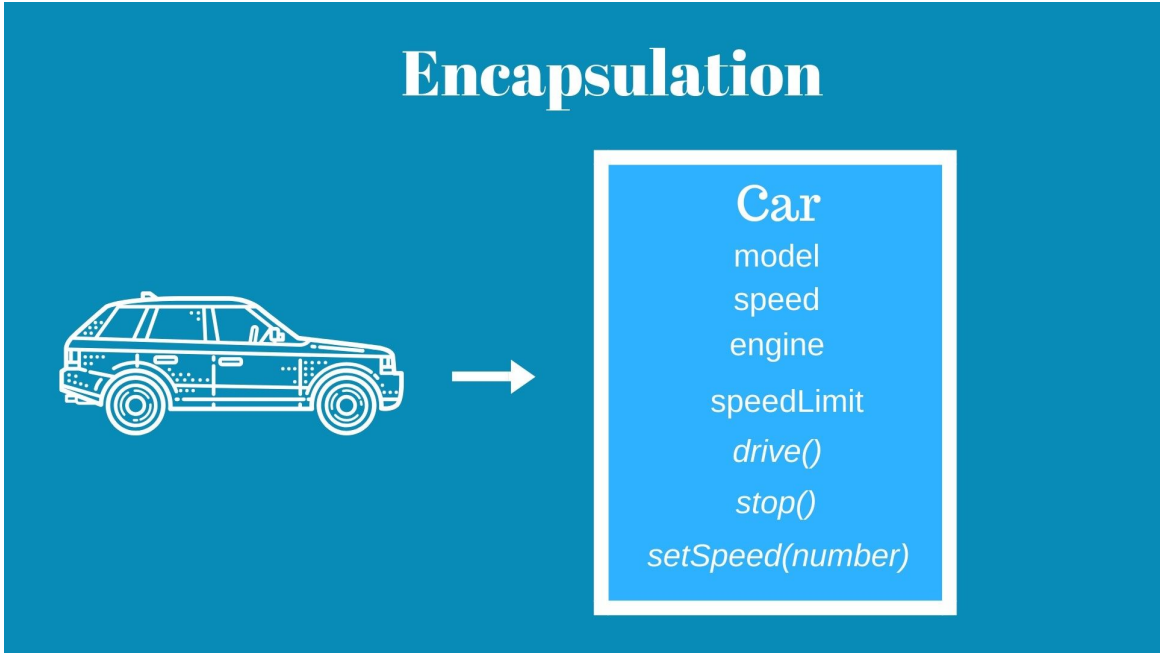


Полиморфизм





Инкапсуляция



ООП в Python

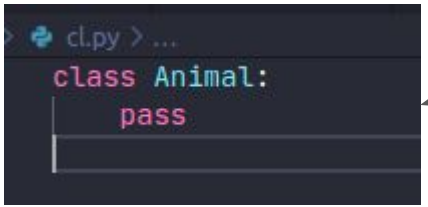
Python - поддерживает принципы ООП

Объект object

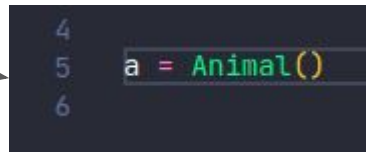
Return a new featureless object. `object` is a base for all classes. It has the methods that are common to all instances of Python classes. This function does not accept any arguments (https://docs.python.org/3/library/functions.html?highlight=object#object)

Ключевое слово class

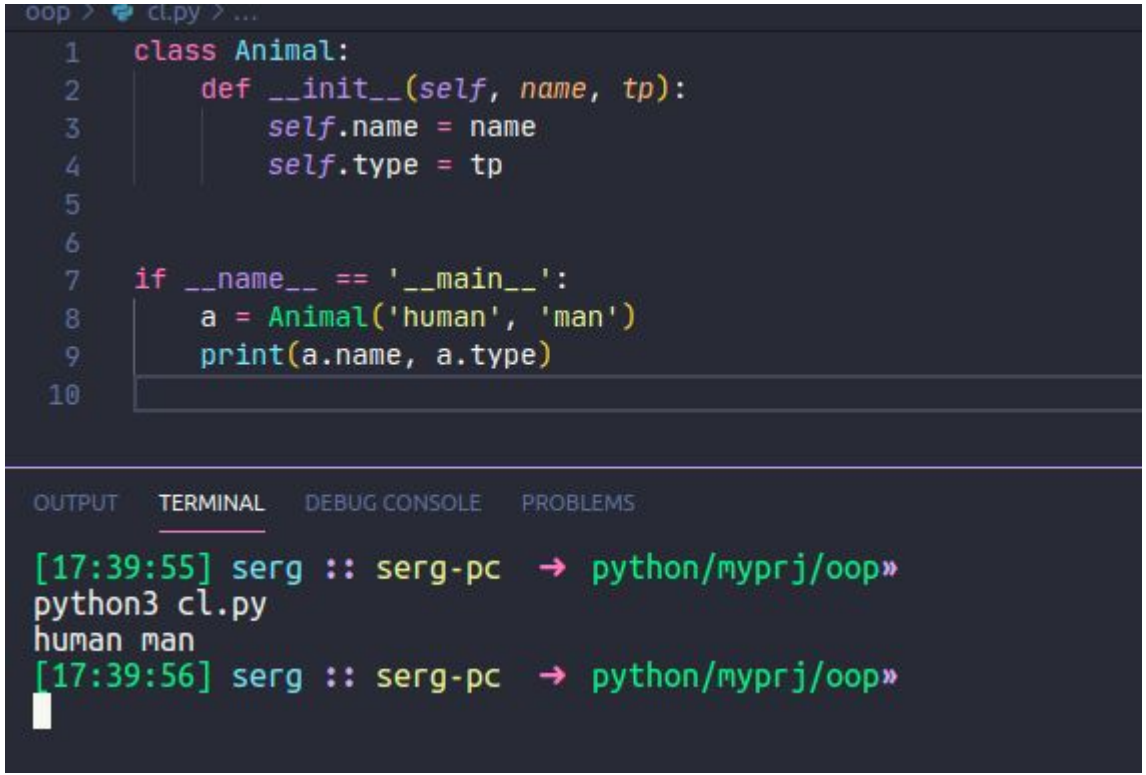
Определение класса



Создание экземпляра класса



Метод конструктора __init__



Ключевое слово self

oop > cl.py > ...

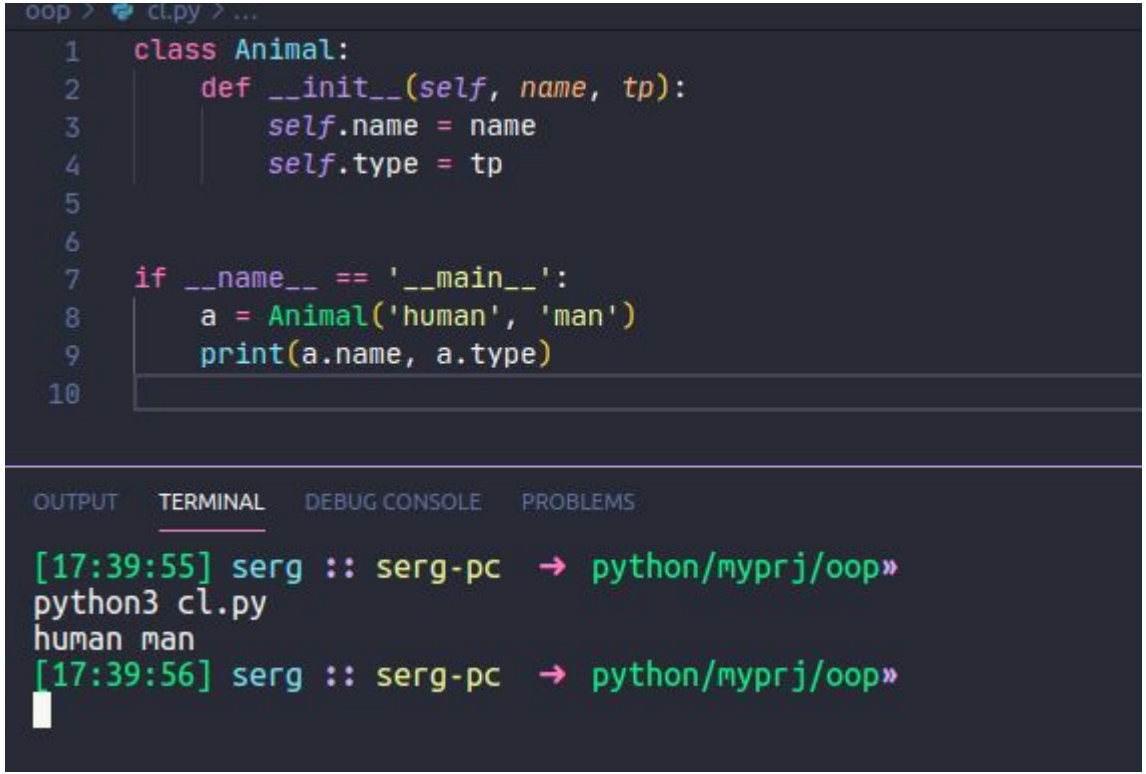
```python
class Animal:
    def __init__(self, name, tp):
        self.name = name
        self.type = tp


if __name__ == '__main__':
    a = Animal('human', 'man')
    print(a.name, a.type)

```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
[17:39:55] serg :: serg-pc  → python/myprj/oop»
python3 cl.py
human man
[17:39:56] serg :: serg-pc  → python/myprj/oop»
```

a = A()
self == a
a.someMeth(arg1) <==>
A.someMeth(a, arg1)

Свойства/атрибуты класса



Методы/функции

oop > cl.py > Animal > say

```python
class Animal:
    def __init__(self, name, tp):
        self.name = name
        self.type = tp

    def run(self):
        print("Animal is running")

    def say(self):
        print("Animal is saying somethig")


if __name__ == '__main__':
    a = Animal('human', 'man')
    print(a.name, a.type)
    a.run()
    a.say()
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
[17:46:49] serg :: serg-pc  → python/myprj/oop»
python3 cl.py
human man
Animal is running
Animal is saying somethig
```

Задачи

1. ООП