

ООП в Python

Полушин Владимир Владимирович

ВМК МГУ

- Инкапсуляция :(
 - Все данные открыты
 - *Исключение:* `--`
- Наследование :/
 - Множественное наследование
 - Duck typing
- Полиморфизм :)
 - Все методы виртуальные
 - Динамическая типизация
 - Duck typing

```
1 class Duck:
2     def __init__(self, weight,
3                 feathers_count=1000):
4         self.weight = weght
5         self.feathers_count = feathers_count
6
7 bald_fat_duck = Duck(100, 50)
8 healthy_duck = Duck(10, feathers_count=1500)
9
10 bald_fat_duck.weight # == 100
11 healthy_duck.feathers_count # == 1500
```

```
1 class Duck:
2     def __init__ ...
3
4     def quack(self):
5         print "Quack!"
6
7     def quack_twice(self):
8         self.quack()
9         self.quack()
10
11 bald_fat_duck.quack()
12 healthy_duck.quack_twice()
```

```
1     class RobotDuck(Duck):
2         def __init__ ...
3
4         def quack(self):
5             print "Quacking mechanism initialized."
6             print "Preparing for quacking..."
7             print "New quack: Quack!"
8             print "Quacking finished."
9
10    robot_duck = RobotDuck(1000)
11    robot_duck.quack_twice()
```

```
1 class QuackingMachine:
2     def __init__ ...
3
4     def quack_twice(self):
5         print "Qack Quack"
6
7     def do_quack(something_quacking):
8         something_quacking.quack_twice()
9
10    do_quack(QuackingMachine())
11    do_quack(robot_duck)
12    do_quack(bald_fat_duck)
```

- `@staticmethod` - статические методы
- `@classmethod` - методы класса
- `__call__` - функторы
- `__repr__` - преобразование в строку
- сеттеры, геттеры, перегрузка операторов, декораторы, метаклассы...