Python

базовый тренинг

Рузин Алексей
ruzin@me.com

Классы

Определение

- class MyClass(object):
    instances = 0
    def __init__(self):
        self.field = 0
        MyClass.instances += 1
    def __del__(self):
        MyClass.instances -= 1
    def get_field_value(self)
        return self.field

- instance = MyClass()

MRO
method resolution order

- MyClass.__mro__

- class X(object): pass
  class Y(X): pass
  class Z(X,Y): pass # TypeError.

Статические методы

- @staticmethod

- class Date(object):
    @staticmethod
    def now():
        t = time.localtime()
        return Date(t.tm_year, t.tm_month, t.tm_day)

a = Date(2000, 1, 1)
b = Date.now()
c = Date.tomorrow()

Методы класса

- @classmethod
  def method(cls, param):
      cls.class_attribute = param

- class MoscowDate(Date):
      …
  md = MoscowDate.now() # Date()

- class Date(object):
      @classmethod
      def now(cls):
          t = time.localtime()
          return cls(t.tm_year, t.tm_month, t.tm_day)

Свойства

- @property

- class Person(object):
    def __init__(self, name):
        self.__name = name
    @property
    def name(self):
        return self.__name

p = Person(‘Natalie’)
print(p.name)

Private

- class MyClass(object):
  def __private(self):
  pass
  # будет переименован в _MyClass__private

- class A(object):
  def __spam(self):
  print('A.spam')
  def spam(self):
  self.__spam()

class B(A):
  def __spam(self):
  print('B.spam')

b = B()
b.spam()

Встроенные аттрибуты

- a = MyClass()
  dir(a) или dir(MyClass)

- __dict__

- __class__

- __slots__ = (‘field1’, ‘field2’)

Перегрузка операторов

- __str__(self)
- __add__(self, other) # и другие, типа __sub__
- __getattr__(self, name)
- __setattr__(self, name, value)

Проверка типов

- isinstance(var, MyClass)

- issubclass(A, B)

Абстрактные классы

- from abc import ABCMeta, abstractmethod

```
class A(metaclass=ABCMeta):
    @abstractmethod
    def spam(self):
        pass

a = A()
```

Задача 1

- Сделать производный класс от **dict**, изменив поведение метода **update** так, чтобы он возвращал ссылку на экземпляр измененного словаря.

d = ExtendedDict(key1=1, key2=‘value2’)
d.update({‘key3’: 3}).update({‘key4’: ‘value4’})

Задача 2

- Спроектировать систему классов, описывающих работу сотрудников в компании с несколькими отделами. Сделать методы для отдела:
  - список всех сотрудников
  - суммарная зарплата всех сотрудников отдела.
  У каждого отдела должен быть ровно один менеджер. Менеджер является сотрудником, но ему должен быть доступен список его подчиненных. Зарплата для обычных сотрудников это константа. Зарплата для Менеджера это константа + надбавка помноженная на количество подчиненных.