python - OOP

Christian Kniep

Internation Center of Applied Technologies Bandung

24. August 2010

Table of content

Good

- inheritance

```
>>> class myInt(int):
...     def __add__(self,x):
...         return self - x
>>> myInt(1)+1
0
```

Good

- inheritance

```
>>> class myInt(int):
...     def __add__(self,x):
...         return self - x
>>> myInt(1)+1
0
```

- polymorphism

```
>>> myInt(1)+2
-1
>>> 1+2
2
>>> '1'+'2'
'12'
```

Good

- inheritance

```
>>> class myInt(int):
...     def __add__(self,x):
...         return self - x
>>> myInt(1)+1
0
```

- polymorphism

```
>>> myInt(1)+2
-1
>>> 1+2
2
>>> '1'+'2'
'12'
```

- Data abstraction

Bad

- its not all rainbows and lipsticks!
- The real world is not all black and white (neat classes)

Bad

- its not all rainbows and lipsticks!
- The real world is not all black and white (neat classes)
- it could become complex to handle al the diverent objects and outcomes

OOPython

- create normal object

```
class myC(object):
    def __init__(self,a):
        self.val = a
    def __del__(self):
        del self.val
    def __str__(self):
        return "My Value is: '%s'" % self.val
    def func(self,b):
        self.val *= b
```

OOPython inheritated

- create inheritated object

```
class childC(myC):
    def func(self,b):
        self.val += b
```

OOPython inheritated

- create inheritated object

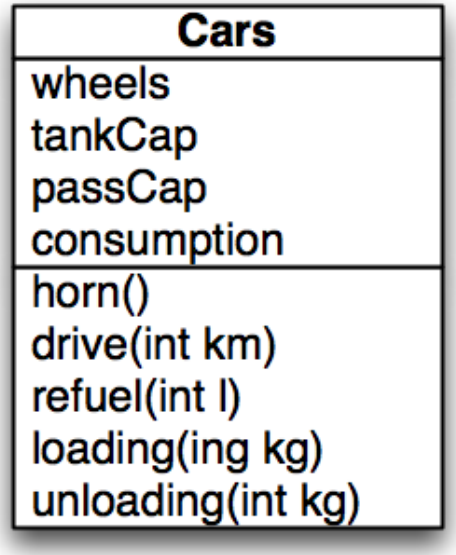
```
class childC(myC):
    def func(self,b):
        self.val += b
```

- exec

```
>>> mC = myC(2)
>>> mC.func(3)
>>> cC = childC(2)
>>> cC.func(3)
>>> print mC
My Value is: '6'
>>> print cC
My Value is: '5'
```

Simple OOP

- Example with cars



Simple OOP

- Example with cars