

Ruby元编程之对象模型

技术管理部 张哲

Ruby is simple in appearance, but is very complex inside, just like our human body.

为什么说 Ruby 是纯面向对象语言？

```
5.times { puts "how world!" }
```

puts 是什么?

- 方法
- 关键字
- 对象
- ?

```
5555.send(:puts, "I am 5555")  
"ss".send(:puts, "I am ss")
```

puts 在哪里定义?

Ruby对象模型


```
class Animal  
end
```

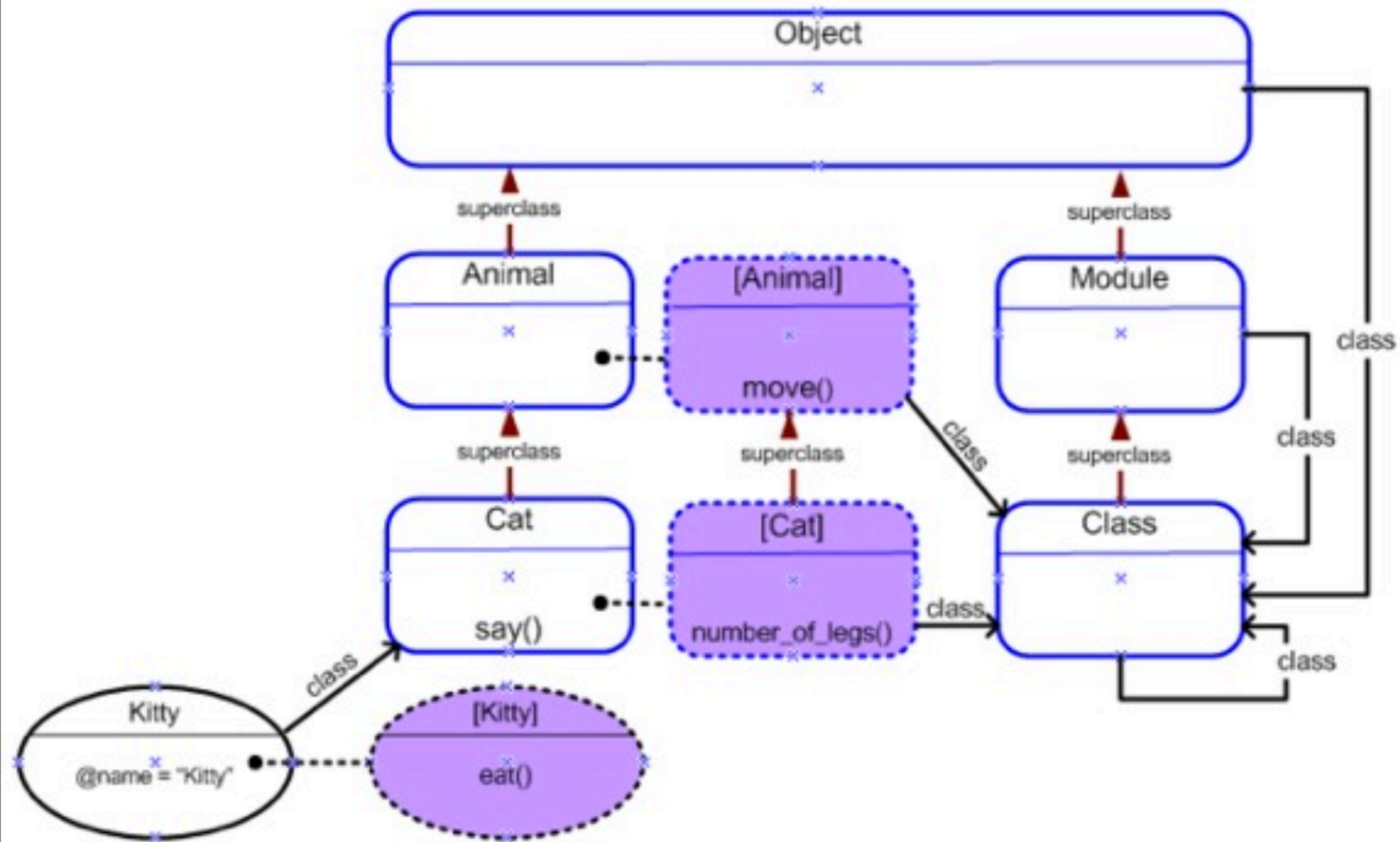
```
class Cat < Animal  
end
```

```
kitty = Cat.new  
p kitty
```

class
superclass
ancestors

Module is a collection of
methods

Class is a module with three
additional methods - new,
allocate, superclass



Classes are objects

Object is a class

干货

- 方法保存在哪里：类、还是对象？
- 定义方法时注意，不要重复已有方法
- 方法出自哪里
- 什么时候使用 `module`
- 不要做 `monkeypatching`

Rails Best Practice

- <http://www.slideshare.net/ihower/rails-best-practices>
- <http://rails-bestpractices.com>
- https://github.com/flyerhzm/rails_best_practices

谢谢

演示代码

```
class Animal
end
class Cat < Animal
end

kitty = Cat.new
p kitty

p kitty.class
p Cat.superclass
p Cat.superclass.superclass.superclass
p nil.class.methods
p nil.object_id

p 1.id
p 1.id
p "1".id
p "1".id
p Cat.ancestors

p "ss".class
p "ss".class.class
p "ss".class.class.class

p Class.superclass

p String.methods - Kernel.methods
s = "ss"
s.instance_eval do
  def test11
    puts "in test11"
  end
end

p s.method(:test11)
```