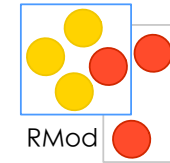


# A Little Journey in the Smalltalk Model

Stéphane Ducasse  
stephane.ducasse@inria.fr  
<http://stephane.ducasse.free.fr/>



# A pure and minimal object model

No constructors

No types declaration

No interfaces

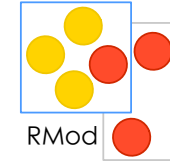
No packages/private/protected

No parametrized types

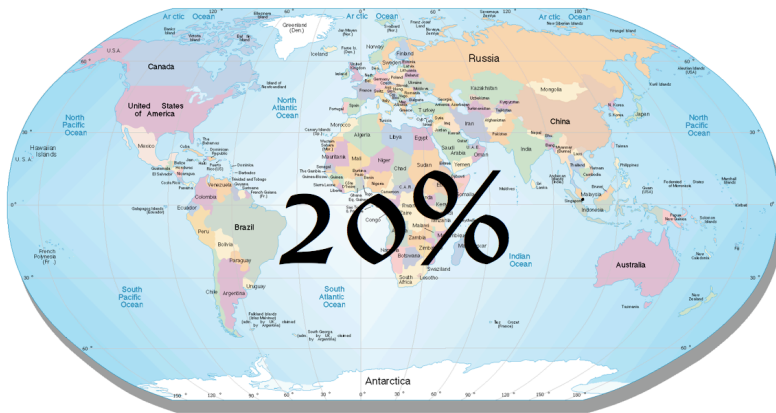
No boxing/unboxing

Still powerful





*We take it personally*



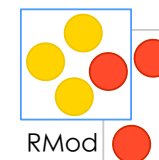
1600 simultaneous users

1.9 billions objects

10 000 updates/day

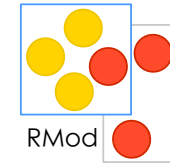


Smarter Choice

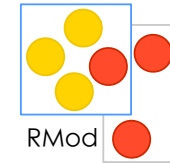


**AIRBUS**



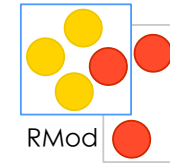


# Objects are instances of Classes



# Objects are instances of Classes

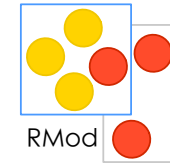
(10@200)



# Objects are instances of Classes

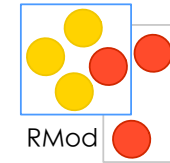
(10@200) class



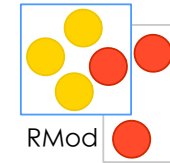


# Objects are instances of Classes

(10@200) class  
Point

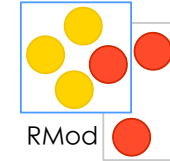


# Classes are objects too



# Classes are objects too

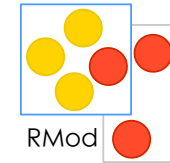
## Point selectors



# Classes are objects too

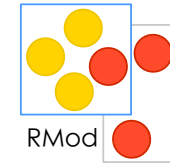
## Point selectors

```
> an IdentitySet(#eightNeighbors #+ #isZero #sortsBefore: #degrees #printOn: #sideOf: #fourNeighbors #hash
#roundUpTo: #min: #min:max: #max #adaptToCollection:andSend: #quadrantOf: #crossProduct: #=
#nearestPointOnLineFrom:to: #bitShiftPoint: #* #guarded #insideTriangle:with:with: #grid: #truncateTo: #y #setR:degrees:
#normal #directionToLineFrom:to: #truncated #nearestPointAlongLineFrom:to: #theta #scaleTo: #encodePostscriptOn:
#> #asPoint #extent: #r #roundTo: #max: #interpolateTo:at: #triangleArea:with: #angleWith: #dotProduct:
#isSelfEvaluating #'<=' #to:intersects:to: #'/' #isInsideCircle:with:with: #< #scaleFrom:to: #corner: #to:sideOf: #x #'>='
#roundDownTo: #onLineFrom:to:within: #transposed #ceiling #angle #basicType #translateBy: #asFloatPoint #'\'
#adaptToNumber:andSend: #abs #negated #octantOf: #asIntegerPoint #flipBy:centerAt: #scaleBy: #floor #onLineFrom:to:
#isPoint #reflectedAbout: #/ #dist: #asNonFractionalPoint #bearingToPoint: #reciprocal #rotateBy:centerAt:
#rotateBy:about: #rounded #setX:setY: #squaredDistanceTo: #normalized #veryDeepCopyWith: #- #storeOn: #rect:
#deepCopy #isIntegerPoint #min #adhereTo: #adaptToString:andSend:)
```



# Classes are objects too

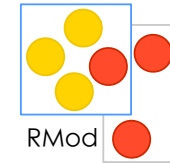
## Point instVarNames



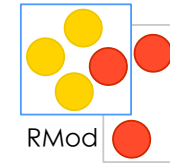
# Classes are objects too

Point instVarNames

```
>#('x' 'y')
```

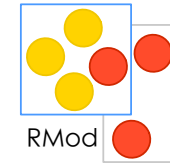


# Methods are public

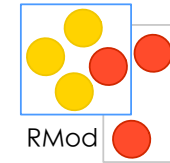


# Instance variables are protected



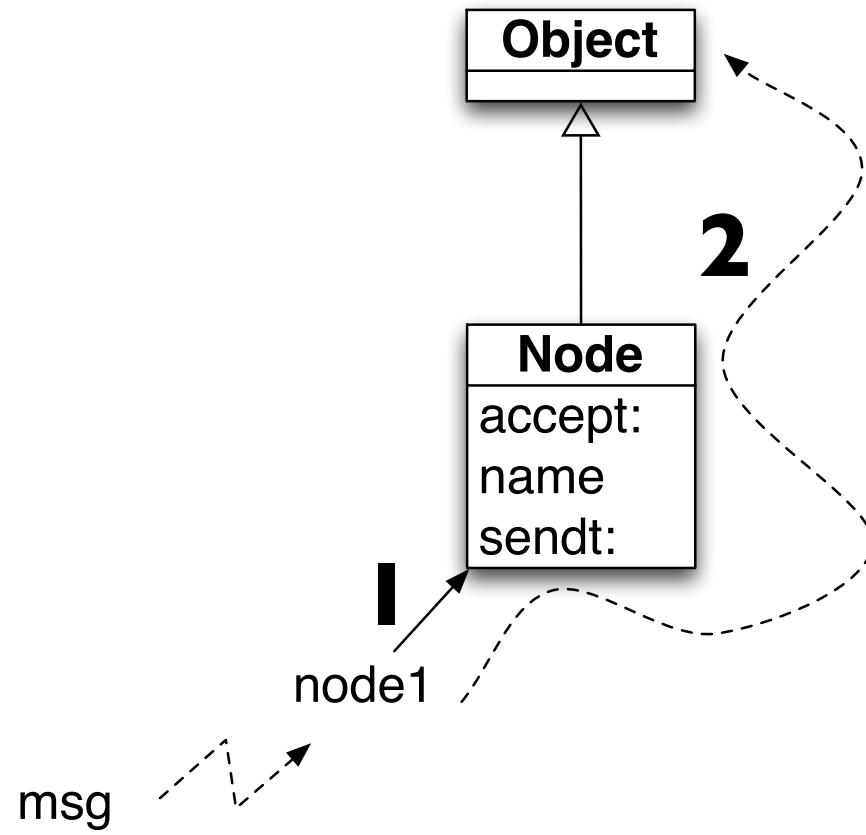


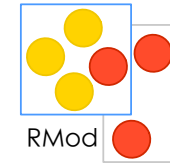
# Single Inheritance



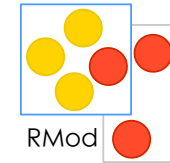
# Single Inheritance

```
Object subclass: #Point  
instanceVariableNames: 'x y'  
classVariableNames: ''  
poolDictionaries: ''  
category: 'Graphics-Primitives'
```



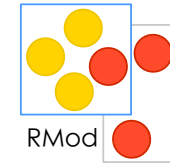


# Classes are objects too



# Classes are objects too

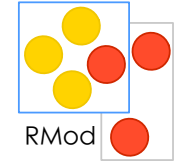
## Point class



# Classes are objects too

Point class

>Point class



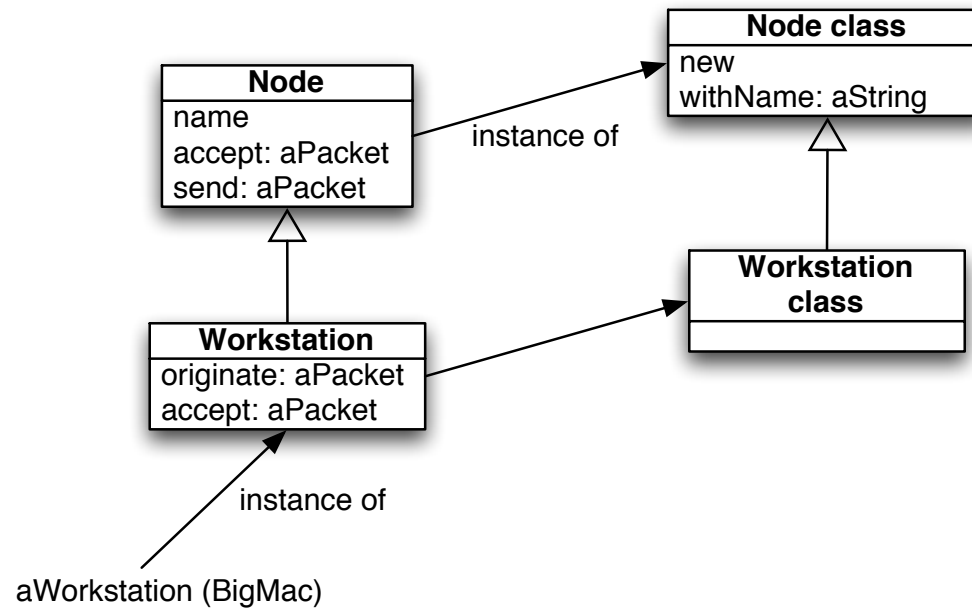
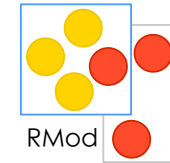
# Classes are objects too

Point class

>Point class

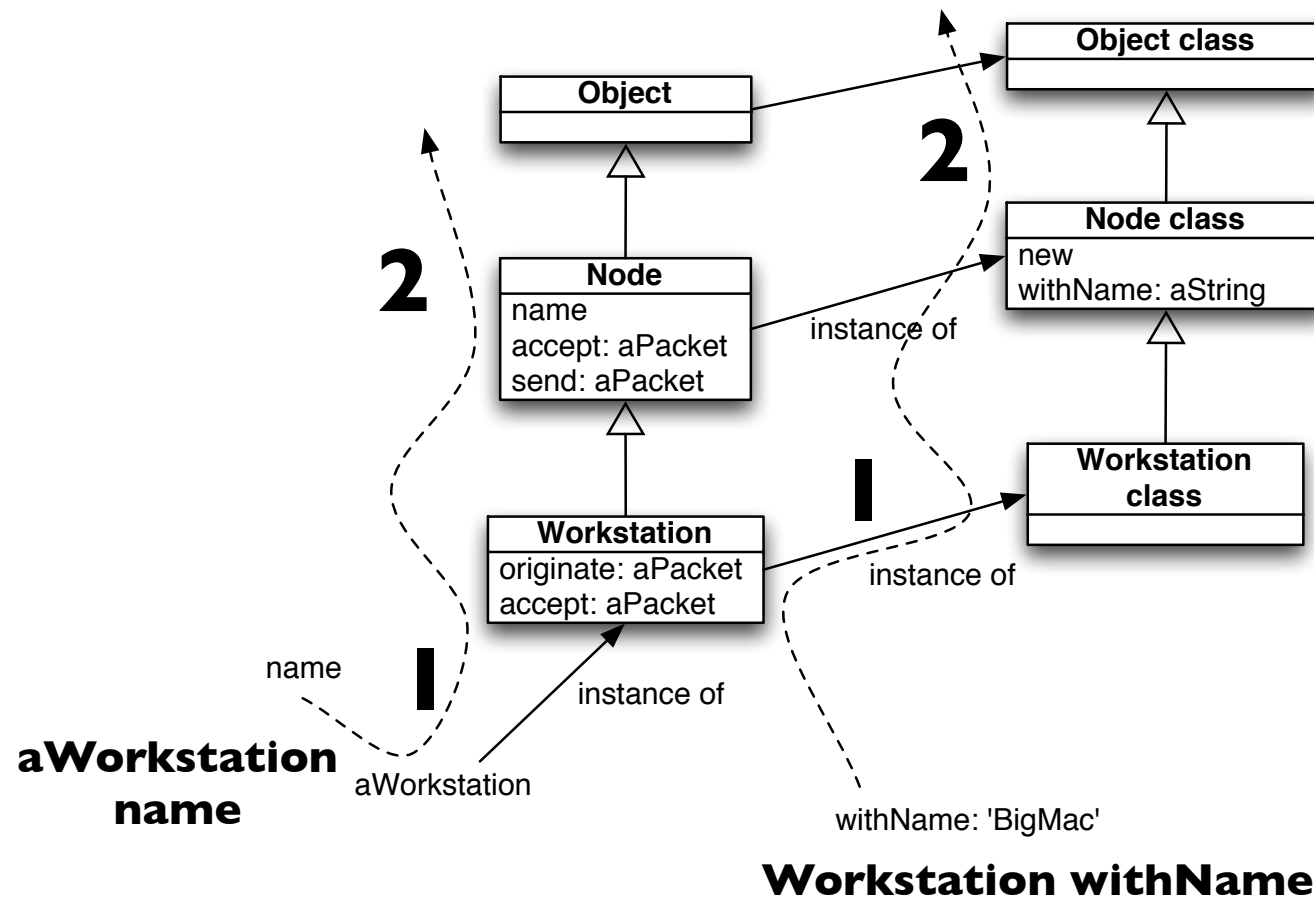
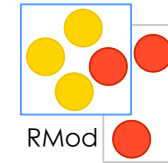
“Point class” is an anonymous class with only one instance: Point

# Class Parallel Inheritance



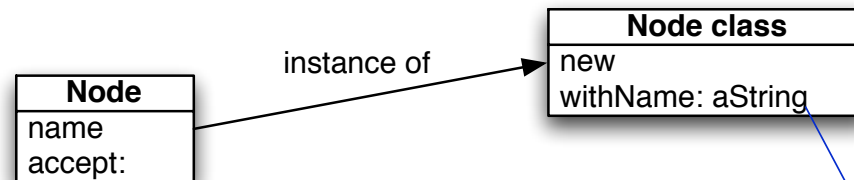
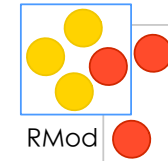


# Lookup and Class Methods



**Workstation withName: 'BigMac'**

# About the Buttons



System Browser: Node

Refactory-Supp Node -- all -- name

Refactory-Parse accessing

Refactory-Parse

Refactory-Squeez

Refactory-Scann instance ? class

Refactory-Refac

browse senders implementors versions inheritance hierarchy inst v

**name**

↑ name

System Browser: Node

Refactory-Supp Node -- all -- withName:

Refactory-Parse instance creati

Refactory-Parse

Refactory-Squeez

Refactory-Scann instance ? class

Refactory-Refac

browse senders implementors versions inheritance hierarchy inst v

**withName: aSymbol**

↑ self new name: aSymbol

# Summary

- Everything is an object
- One single model
- Single inheritance
- Public methods
- Protected attributes
- Classes are simply objects too
- Class is instance of another class
- One unique method lookup  
look in the class of the receiver