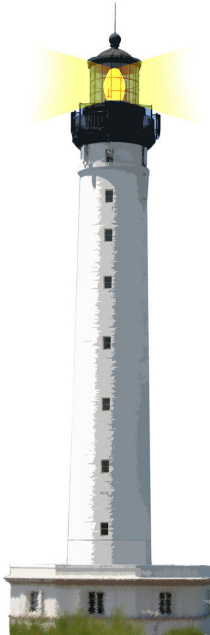
Learning Object-Oriented Programming and Design with TDD

First Look at Class and Method Definitions

Stéphane Ducasse
http://stephane.ducasse.free.fr

http://www.pharo.org

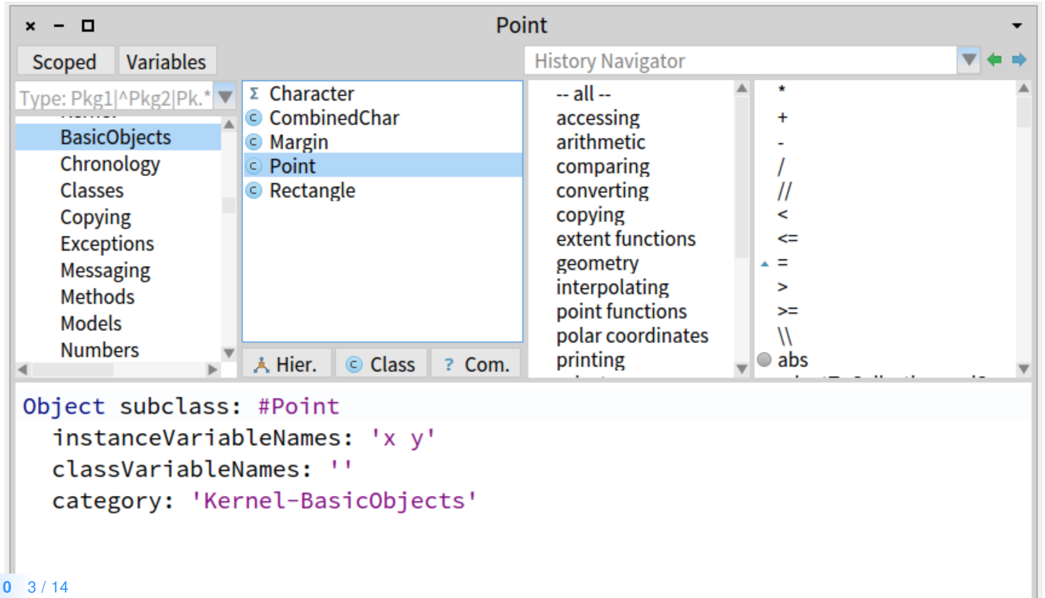
W1S10



Class and Method Definitions in Pharo

- We will have specific lectures on classes and methods.
- Now this is just to give you a first impression
- Most of the time you will define classes and methods using tools
- There is no dedicated syntax for class definition just a message

Class Definition in Pharo



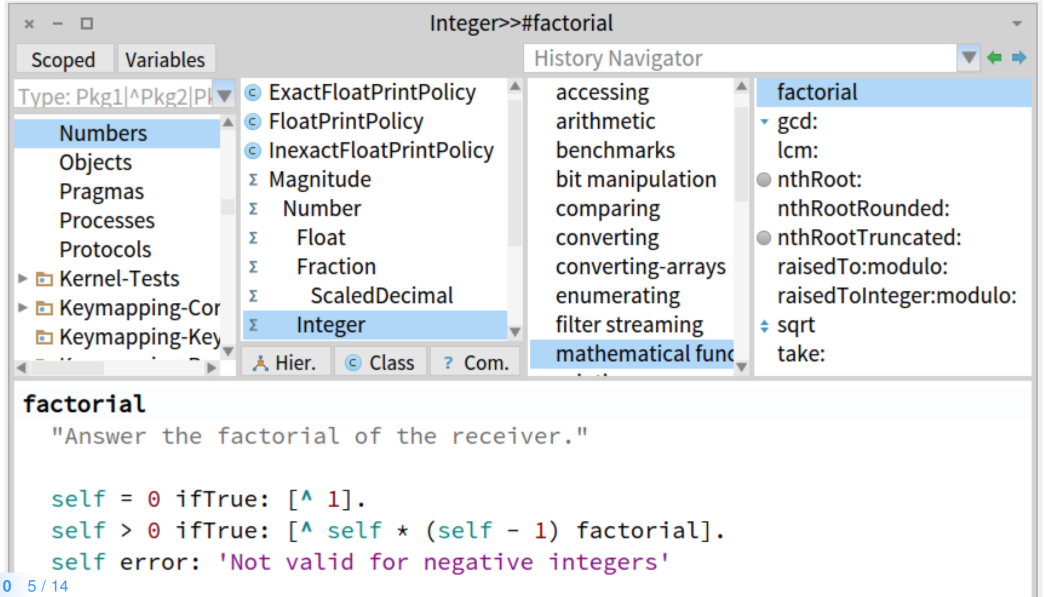
```
Object subclass: #Point
  instanceVariableNames: 'x y'
  classVariableNames: ''
  category: 'Kernel-BasicObjects'
```

Class Definition is a Message

```
Object subclass: #Point
  instanceVariableNames: 'x y'
  classVariableNames: ''
  package: 'Graphics'
```

We send the message subclass:inst.... to the superclass to create the class

Method Definition in Pharo



```
factorial
  "Answer the factorial of the receiver."

  self = 0 ifTrue: [^ 1].
  self > 0 ifTrue: [^ self * (self - 1) factorial].
  self error: 'Not valid for negative integers'
```

Method Definition in Pharo

```
factorial
  "Answer the factorial of the receiver."
  self = 0 ifTrue: [ ^ 1 ].
  self > 0 ifTrue: [ ^ self * (self – 1) factorial ].
  self error: 'Not valid for negative integers'
```

In which class is factorial defined?

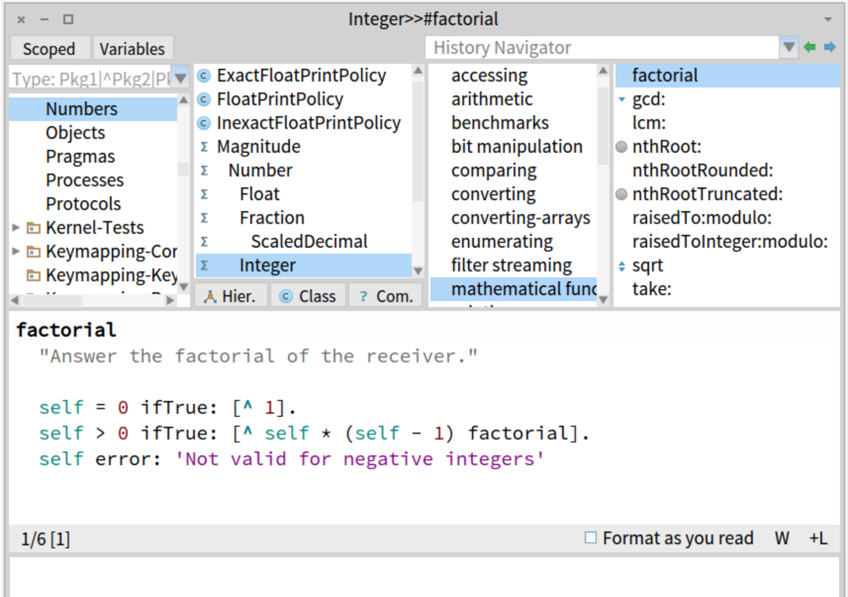
Presentation Convention

In this lecture, a method will be displayed as

```
Integer >> factorial
   "Answer the factorial of the receiver."
   self = 0 ifTrue: [ ^ 1 ].
   self > 0 ifTrue: [ ^ self * (self – 1) factorial ].
   self error: 'Not valid for negative integers'
```

- **Integer >>** is not part of the syntax
  - it tells you the method's class

Presentation Convention



```
factorial
  "Answer the factorial of the receiver."

  self = 0 ifTrue: [^ 1].
  self > 0 ifTrue: [^ self * (self - 1) factorial].
  self error: 'Not valid for negative integers'
```

Remember Messages

```
Integer >> factorial
  "Answer the factorial of the receiver."

  self = 0 ifTrue: [ ^ 1 ].
  self > 0 ifTrue: [ ^ self * (self – 1) factorial ].
  self error: 'Not valid for negative integers'
```

- factorial is the method name
- =, >, * and - are binary messages
- factorial is an unary message
- ifTrue: and error: are keyword messages
- the caret ^ is for returning a value

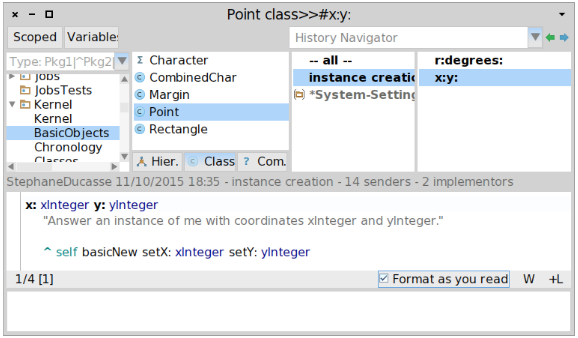
A Method Returns self by Default

```
Game >> initializePlayers
   self players
     at: 'tileAction'
     put: ( MITileAction director: self )
```

is equivalent to

```
Game >> initializePlayers
   self players
     at: 'tileAction'
     put: ( MITileAction director: self ).
   ^ self      "<—— optional"
```

Class Methods



StephaneDucasse 11/10/2015 18:35 - instance creation - 14 senders - 2 implementors

```
x: xInteger y: yInteger
    "Answer an instance of me with coordinates xInteger and yInteger."

    ^ self basicNew setX: xInteger setY: yInteger
```

- press the button class to define a class method
- in lectures, we add class

```
Point class >> x: xInteger y: yInteger
   "Answer an instance of me with coordinates xInteger and yInteger."

   ^ self basicNew setX: xInteger setY: yInteger
```

What You Should Know

- A class is defined by sending a message to its superclass
- Classes are defined inside packages
- Methods are public
- By default a method returns the receiver, self
- Class methods are just methods of the class side

Resources

- Pharo Mooc - W1S06 Videos
- Pharo by Example `http://books.pharo.org`

A course by Stéphane Ducasse
http://stephane.ducasse.free.fr

Reusing some parts of the Pharo Mooc by

Damien Cassou, Stéphane Ducasse, Luc Fabresse
http://mooc.pharo.org