

Class and Method Definition

Stéphane Ducasse and Damien Cassou

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

Class Definition

```
Object subclass: #NameOfSubclass  
  instanceVariableNames: ''  
  classVariableNames: ''  
  category: 'PackageName'
```

Example

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

Methods

- Public.
- Virtual (late bound): always dynamically looked up.
- By default return the receiver (like if we add a return self at the end of any method)
- This ease composition

```
initializePlayers
  super initializePlayers.
  self players
    at: 'tileAction'
    put: ( MITileAction director: self )
```

is equivalent to

```
initializePlayers
  super initializePlayers.
  self players
    at: 'tileAction'
    put: ( MITileAction director: self )
^ self
```

Methods

yourself

is equivalent to

yourself
^ self

Typographical convention

When you edit a method, you do not type the class name

```
crossProduct: aPoint
```

```
"Answer a number that is the cross product of the receiver and the  
argument, aPoint."
```

```
^ (x * aPoint y) - (y * aPoint x)
```

But when we show code you do not know in which class the method is defined.

```
Point >> crossProduct: aPoint
```

```
"Answer a number that is the cross product of the receiver and the  
argument, aPoint."
```

```
^ (x * aPoint y) - (y * aPoint x)
```

An Example of Method

```
Point >> dist: aPoint
```

```
"Answer the distance between aPoint and the receiver."
```

```
| dx dy |
```

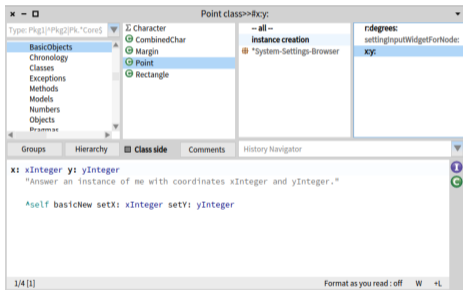
```
dx := aPoint x - x.
```

```
dy := aPoint y - y.
```

```
^ (dx * dx + (dy * dy)) sqrt
```

Class Method

To define a class method, press the class side button



In books and lecture notes 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

Summary

- Classes are defined inside packages.
- Methods are public and virtual.
- Class methods are just methods of the class.
- By default a method returns the receiver.