LIRMM

HMIN306
Evolution et restructuration

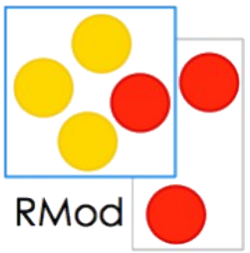
A Short introduction

Nicolas Anquetil

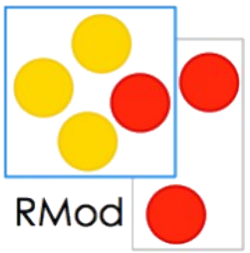
nicolas.anquetil@inria.fr

RMod

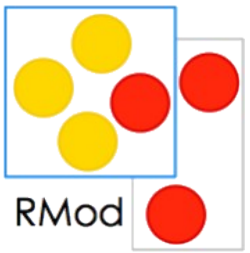
- A programming language
  - Pure Object-Oriented (Reflective)
  - Dynamically typed
  - Open and flexible
- An IDE
  - Programming
  - Debugging
  - Live inspection of objects
- An execution environment

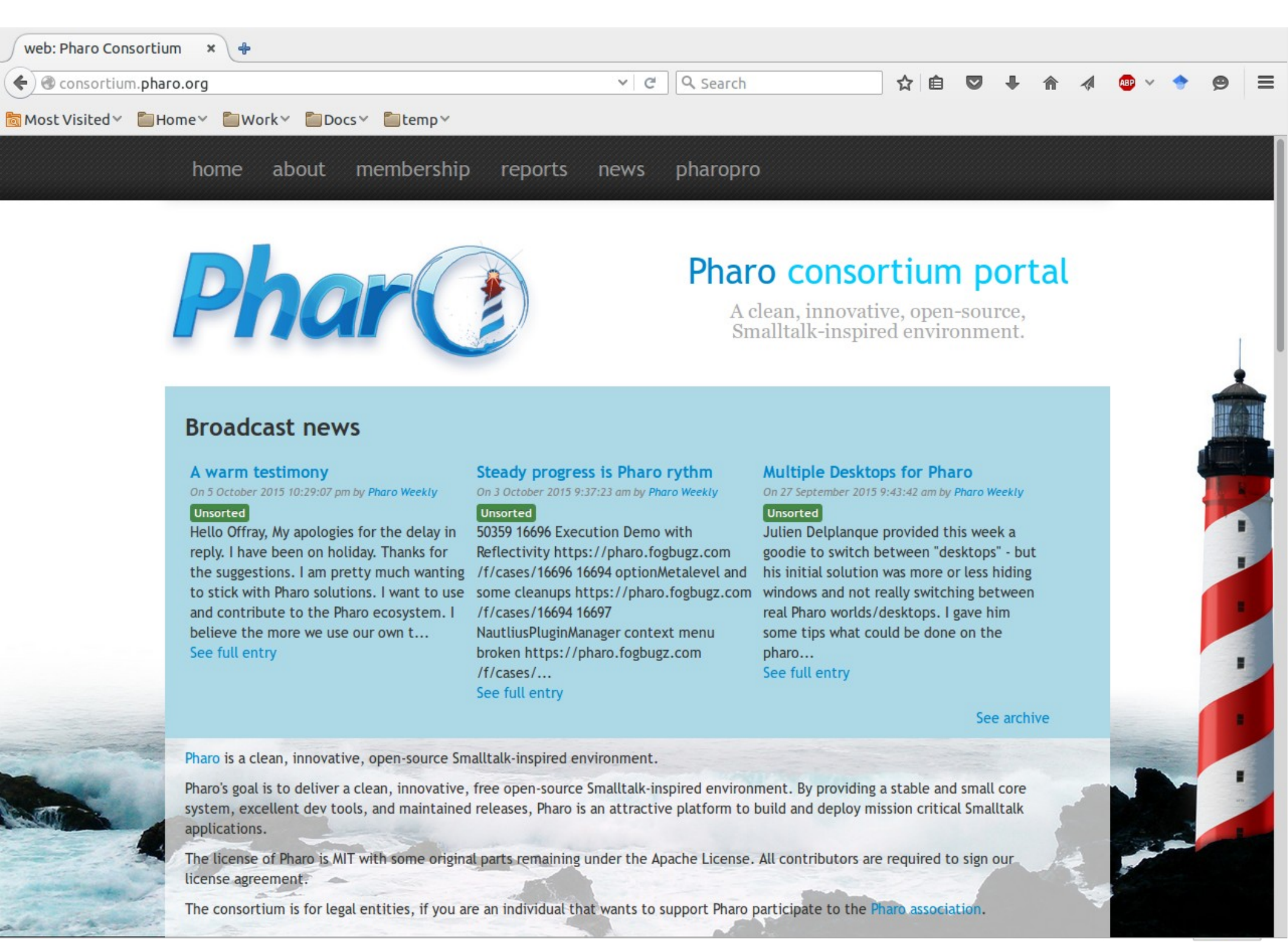


- MIT license
- Powerful, elegant
- Multiplatform: Mac, Linux, Windows, Android

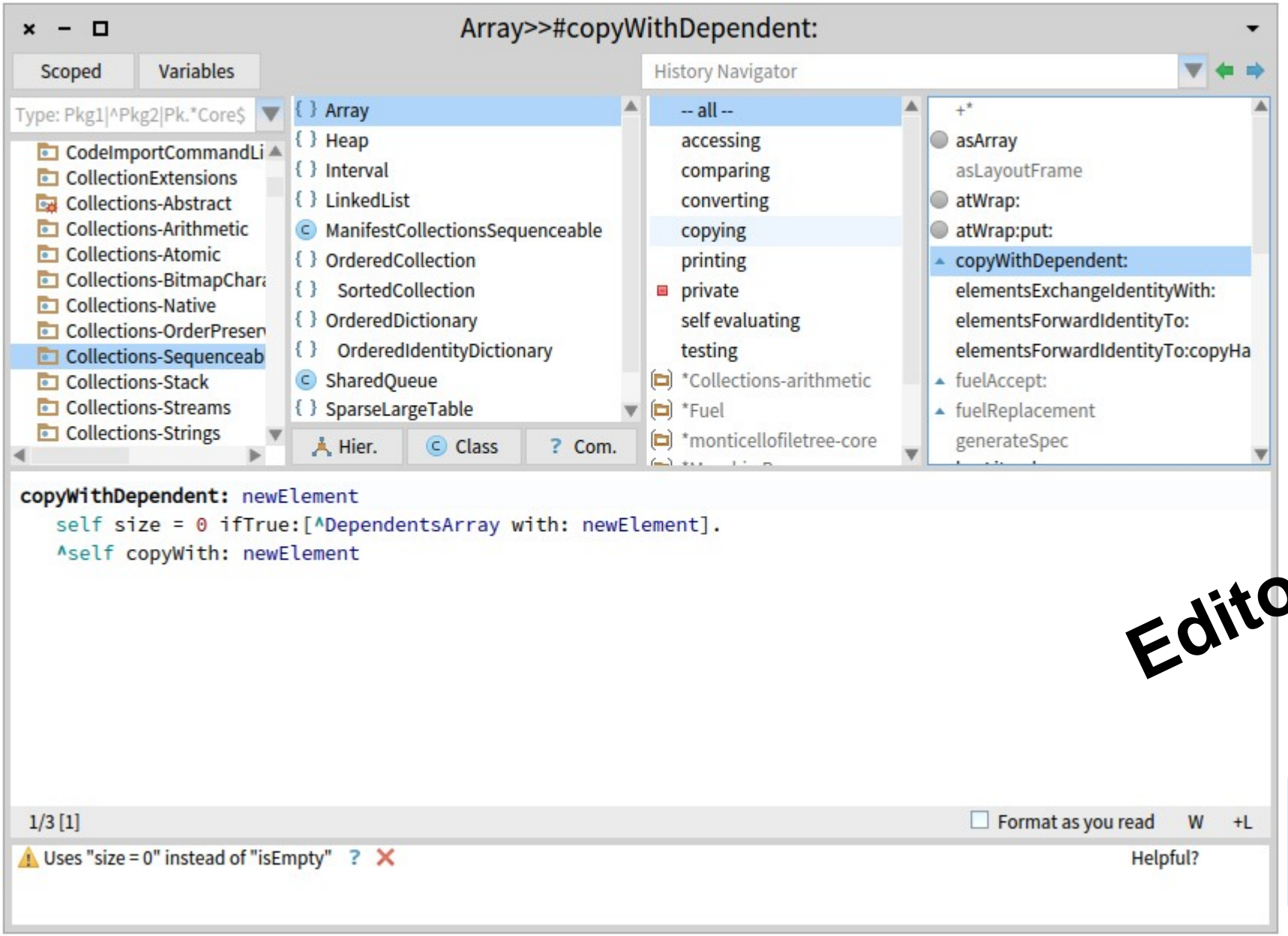


- Ultimate goal:
  - Create an ecosystem where business and innovation can bloom
- Pharo consortium
  - `http://consortium.pharo.org/`




web: Pharo Consortium
consortium.pharo.org
Search
Most Visited
Home
Work
Docs
temp
home
about
membership
reports
news
pharopro
Pharo consortium portal
A clean, innovative, open-source,
Smalltalk-inspired environment.
Broadcast news
A warm testimony
On 5 October 2015 10:29:07 pm by Pharo Weekly
Unsorted
Hello Offray, My apologies for the delay in reply. I have been on holiday. Thanks for the suggestions. I am pretty much wanting to stick with Pharo solutions. I want to use and contribute to the Pharo ecosystem. I believe the more we use our own t...
See full entry
Steady progress is Pharo rythm
On 3 October 2015 9:37:23 am by Pharo Weekly
Unsorted
50359 16696 Execution Demo with Reflectivity https://pharo.fogbugz.com /f/cases/16696 16694 optionMetalevel and some cleanups https://pharo.fogbugz.com /f/cases/16694 16697 NautliusPluginManager context menu broken https://pharo.fogbugz.com /f/cases/...
See full entry
Multiple Desktops for Pharo
On 27 September 2015 9:43:42 am by Pharo Weekly
Unsorted
Julien Delplanque provided this week a goodie to switch between "desktops" - but his initial solution was more or less hiding windows and not really switching between real Pharo worlds/desktops. I gave him some tips what could be done on the pharo...
See full entry
See archive
Pharo is a clean, innovative, open-source Smalltalk-inspired environment.
Pharo's goal is to deliver a clean, innovative, free open-source Smalltalk-inspired environment. By providing a stable and small core system, excellent dev tools, and maintained releases, Pharo is an attractive platform to build and deploy mission critical Smalltalk applications.
The license of Pharo is MIT with some original parts remaining under the Apache License. All contributors are required to sign our license agreement.
The consortium is for legal entities, if you are an individual that wants to support Pharo participate to the Pharo association.

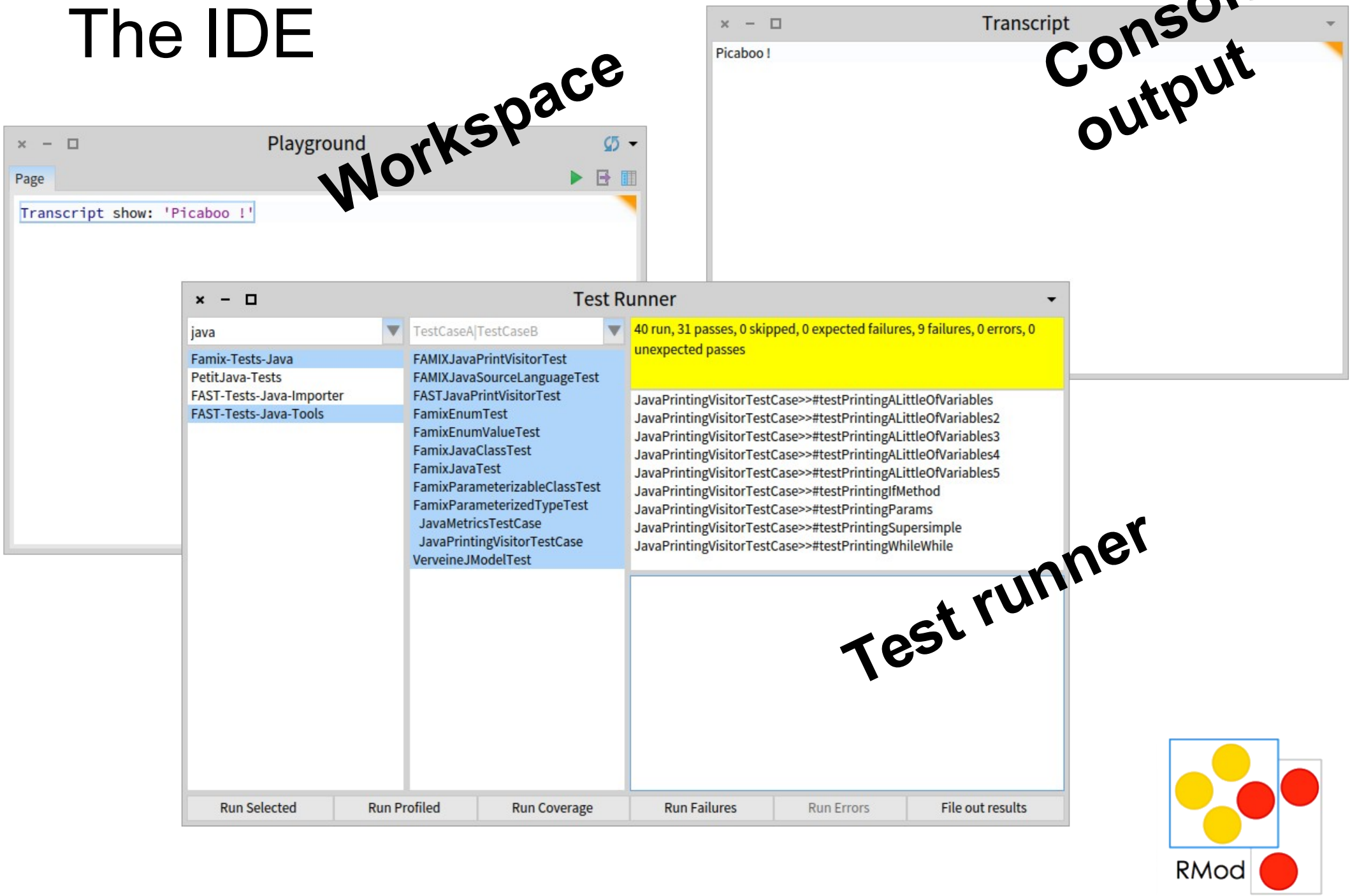
The IDE



Editor

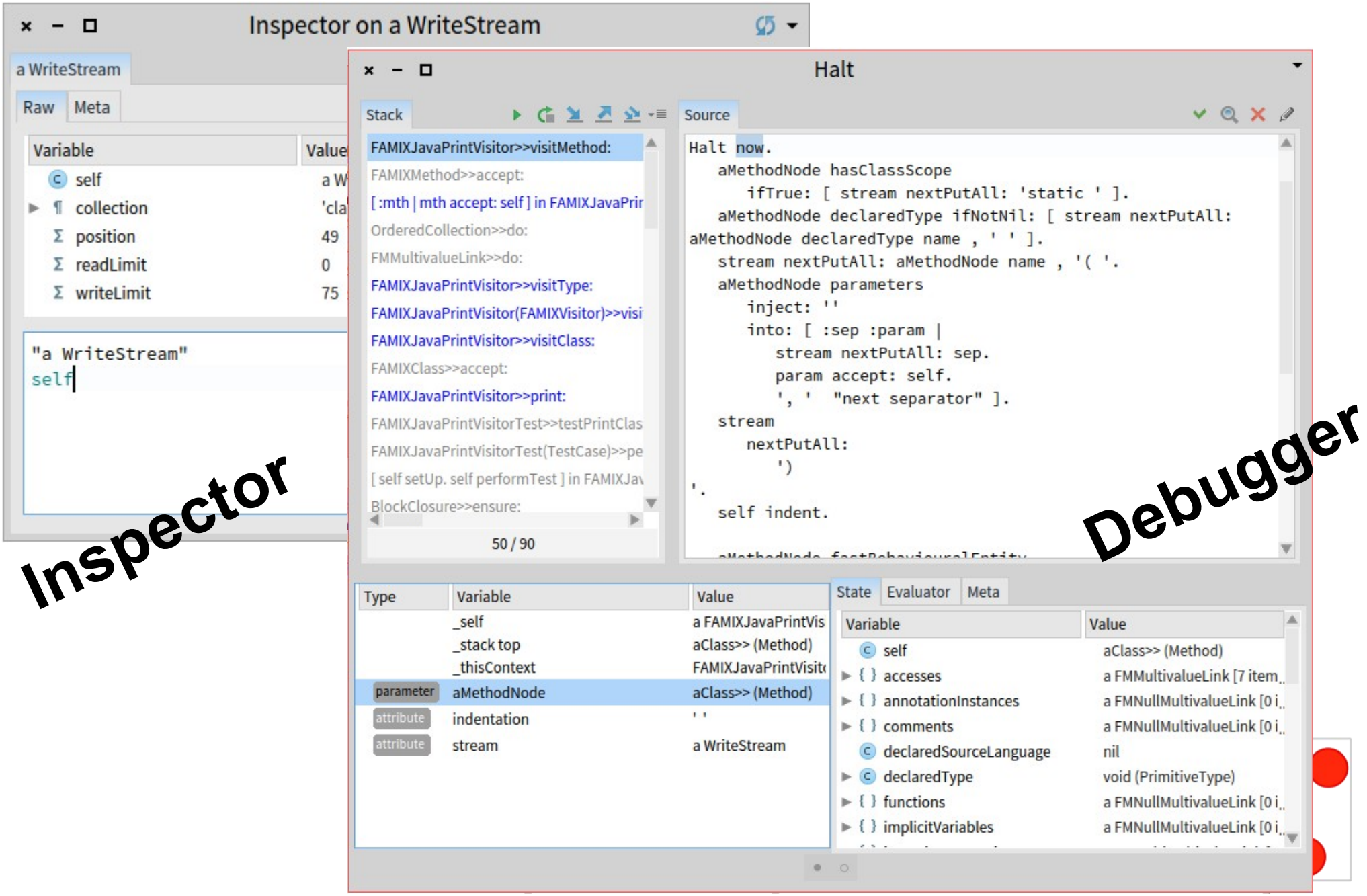
RMod

The IDE

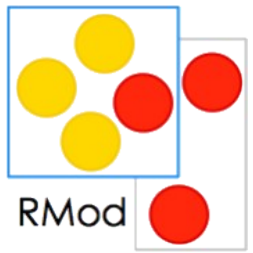

Workspace
Console output
Transcript
Picaboo !
Playground
Page
Transcript show: 'Picaboo !'
Test Runner
java
TestCaseA|TestCaseB
40 run, 31 passes, 0 skipped, 0 expected failures, 9 failures, 0 errors, 0 unexpected passes
Famix-Tests-Java
PetitJava-Tests
FAST-Tests-Java-Importer
FAST-Tests-Java-Tools
FAMIXJavaPrintVisitorTest
FAMIXJavaSourceLanguageTest
FASTJavaPrintVisitorTest
FamixEnumTest
FamixEnumValueTest
FamixJavaClassTest
FamixJavaTest
FamixParameterizableClassTest
FamixParameterizedTypeTest
JavaMetricsTestCase
JavaPrintingVisitorTestCase
VerveineJModelTest
JavaPrintingVisitorTestCase>>#testPrintingALittleOfVariables
JavaPrintingVisitorTestCase>>#testPrintingALittleOfVariables2
JavaPrintingVisitorTestCase>>#testPrintingALittleOfVariables3
JavaPrintingVisitorTestCase>>#testPrintingALittleOfVariables4
JavaPrintingVisitorTestCase>>#testPrintingALittleOfVariables5
JavaPrintingVisitorTestCase>>#testPrintingIfMethod
JavaPrintingVisitorTestCase>>#testPrintingParams
JavaPrintingVisitorTestCase>>#testPrintingSupersimple
JavaPrintingVisitorTestCase>>#testPrintingWhileWhile
Test runner
Run Selected
Run Profiled
Run Coverage
Run Failures
Run Errors
File out results

RMod

The IDE


Inspector on a WriteStream
a WriteStream
Raw
Meta
Variable
Value
self
collection
position
49
readLimit
0
writeLimit
75
"a WriteStream"
self
Inspector
Halt
Stack
FAMIXJavaPrintVisitor>>visitMethod:
FAMIXMethod>>accept:
[:mth | mth accept: self] in FAMIXJavaPrir
OrderedCollection>>do:
FMMultivalueLink>>do:
FAMIXJavaPrintVisitor>>visitType:
FAMIXJavaPrintVisitor(FAMIXVisitor)>>visi
FAMIXJavaPrintVisitor>>visitClass:
FAMIXClass>>accept:
FAMIXJavaPrintVisitor>>print:
FAMIXJavaPrintVisitorTest>>testPrintClas
FAMIXJavaPrintVisitorTest(TestCase)>>pe
[self setUp. self performTest] in FAMIXJav
BlockClosure>>ensure:
50 / 90
Source
Halt now.
aMethodNode hasClassScope
ifTrue: [stream nextPutAll: 'static '].
aMethodNode declaredType ifNotNil: [stream nextPutAll:
aMethodNode declaredType name , ' '].
stream nextPutAll: aMethodNode name , '('.
aMethodNode parameters
inject: ''
into: [:sep :param |
stream nextPutAll: sep.
param accept: self.
', ' "next separator"].
stream
nextPutAll:
')
'.
self indent.
Type
Variable
Value
_self
a FAMIXJavaPrintVis
_stack top
aClass>> (Method)
_thisContext
FAMIXJavaPrintVisit
parameter
aMethodNode
aClass>> (Method)
attribute
indentation
attribute
stream
a WriteStream
State
Evaluator
Meta
Variable
Value
self
aClass>> (Method)
accesses
a FMMultivalueLink [7 item
annotationInstances
a FMNullMultivalueLink [0 i
comments
a FMNullMultivalueLink [0 i
declaredSourceLanguage
nil
declaredType
void (PrimitiveType)
functions
a FMNullMultivalueLink [0 i
implicitVariables
a FMNullMultivalueLink [0 i
Debugger

Everything is an object



Numbers

Closures

Classes

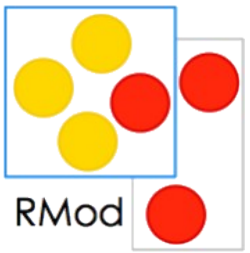
Everything is an object

Methods

Tools

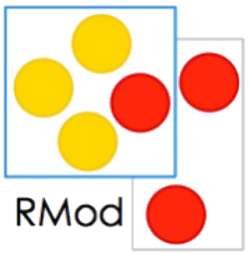
Method calls

Call stack



Everything happens by sending messages to objects

- 2 + 3
- Nautilus open
- Object subclass: #MyClass …
- 4 timesRepeat: …



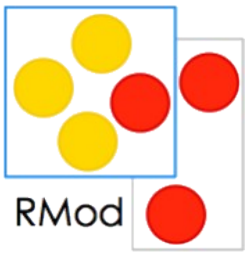
Simple syntax

- Variable declaration

```
| x toto aCollection |
```

- Variable assignment

```
x := 5
```



Simple syntax

- Unary message

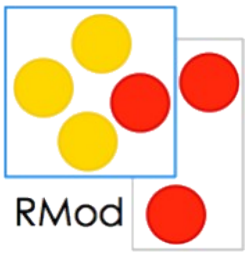
```
5 factorial
```

- Binary message

```
3 + 4
```

- Keywords message

```
3 raisedTo: 10 modulo: 5
```

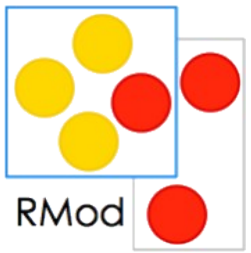


Simple syntax

- Two keywords

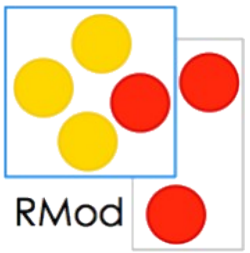
self

super

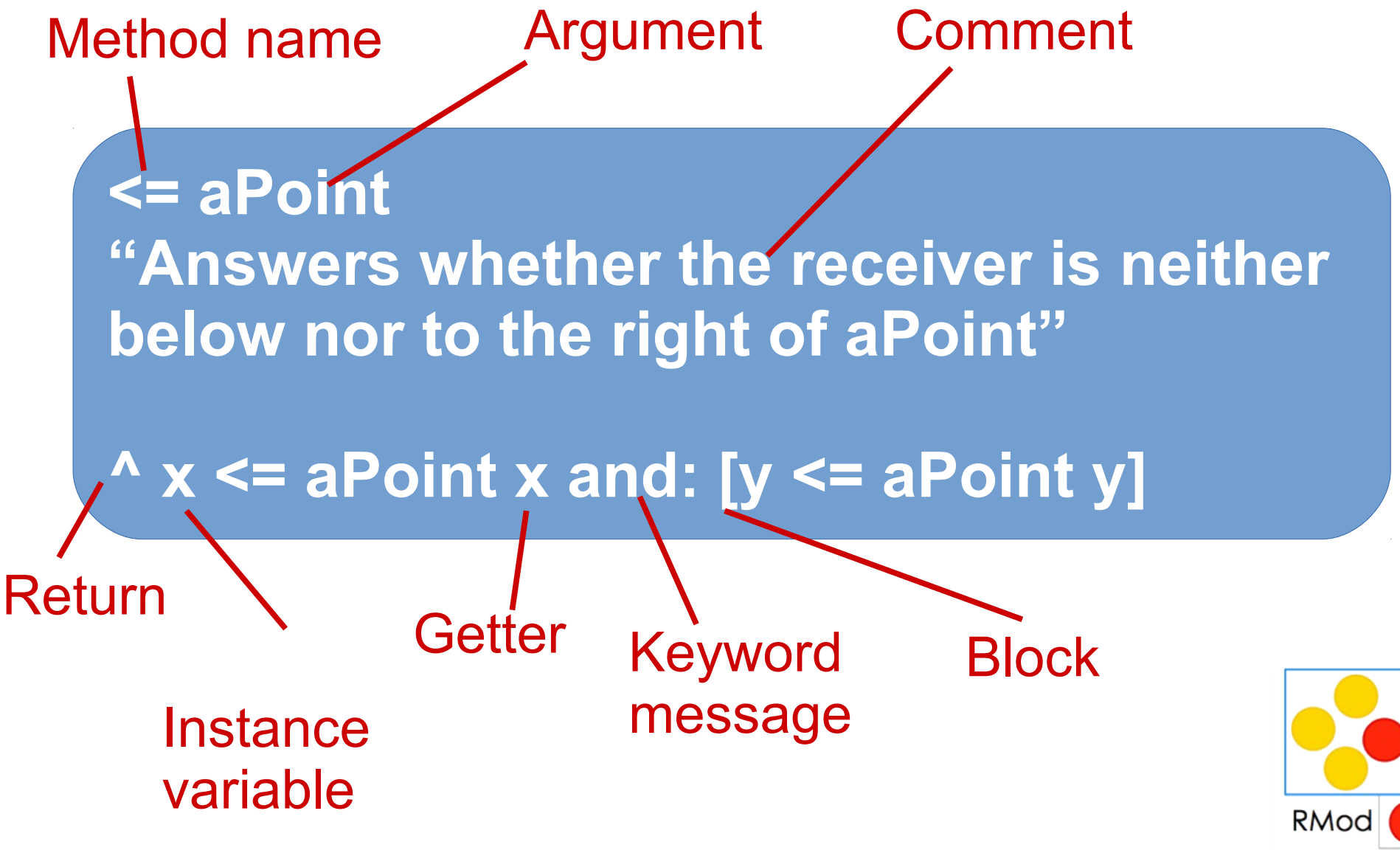


Simple syntax

- Assignment :=
- Return ^
- Comment " "
- String ' '
- Local/block variable | |
- Block [ ]
- Statement separator .
- Cascade ;
- Symbol #
- Character $



Method example (a binary message):



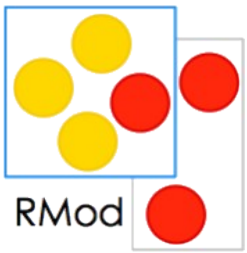
RMod

Control structures realized by message sends

```
3 timesRepeat: [ Transcript show: 'hip !' ]
```

```
(1 to: 10) do: [:i |
    Transcript show: i asString ]
```

```
x < 0
ifTrue: [ Transcript show: 'negative' ]
ifFalse: [ Transcript show: 'positive' ]
```

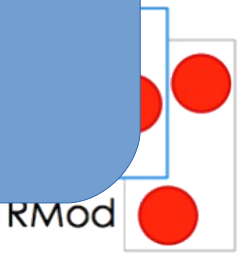


Creating instances by message sends (to the class)

```
Workspace new open
```

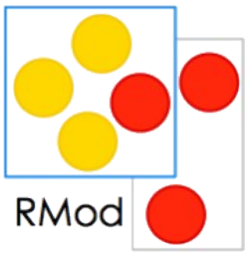
Creating class also (message to the superclass)

```
Number subclass: #Complex
    instanceVariableNames: 'real imaginary'
    classVariableNames: ''
    category: 'ComplexNumbers'
```



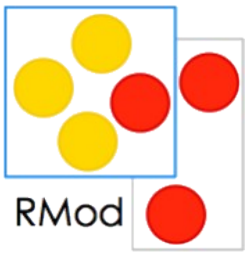
Large API

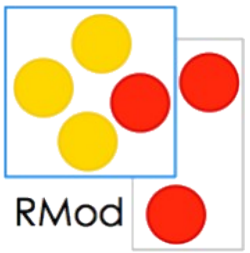
- Collections
  - select:[:anElement | aBooleanExpression]
  - collect:[:anElement | aResult]
  - detect:[:anElement | aBooleanExpression]
- Numbers
  - isPrime
  - isOdd



Exercices

- List numbers from 1 to 1000 ?
- List even numbers between 1 and 1000 ?
- List prime numbers between 1 and 1000 ?
- First prime number superior to 3000 ?



More information

- http://www.pharo-project.org
- http://www.pharobyexample.org
- http://deepintopharo.com
- http://consortium.pharo.org/
- http://forum.world.st/Pharo-Smalltalk-Users-f1310670.html