

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

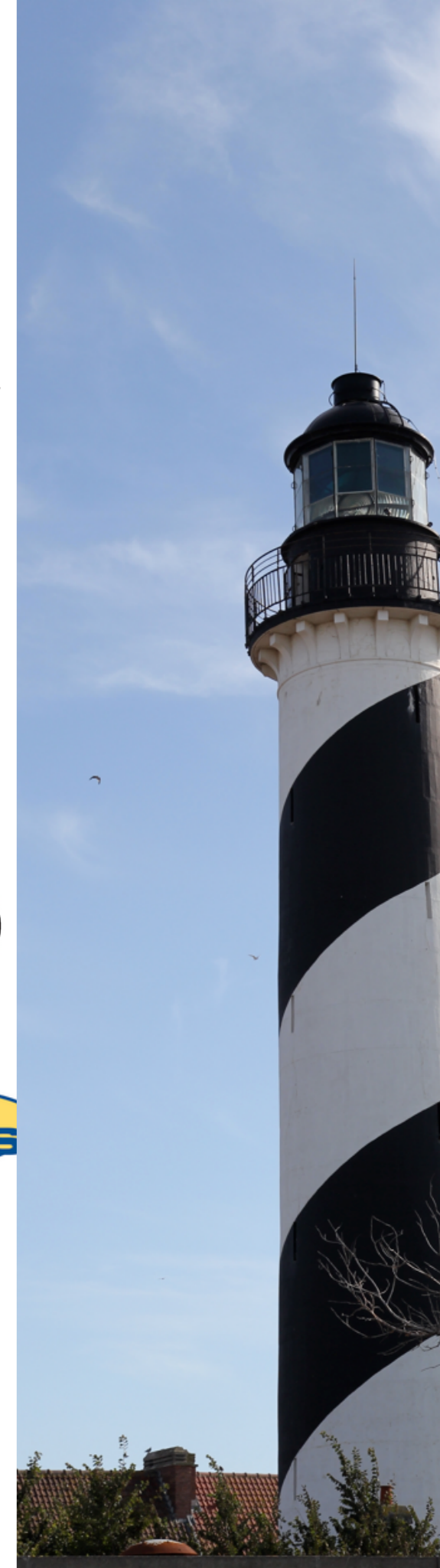
Live tools are developers' best friends

<http://www.pharo.org>
Novi Sad - 2024

Inria

 Université
de Lille

 cnrs



A journey in a live environment and its companion tools

- Pharo in 5 min
- Some advanced features
- Cool tools

Pharo!

- System: **Pure** object language + full IDE
- **Powerful, elegant** and **fun** to program
- **Living** system under your fingers
- Works on 11 oses, 32/64, X86, ARM, M1
- 100% MIT

Pharo in Numbers

17 releases since 2008
Language Core + IDE +
Tools + Frameworks
~ 800 packages (tests
included)

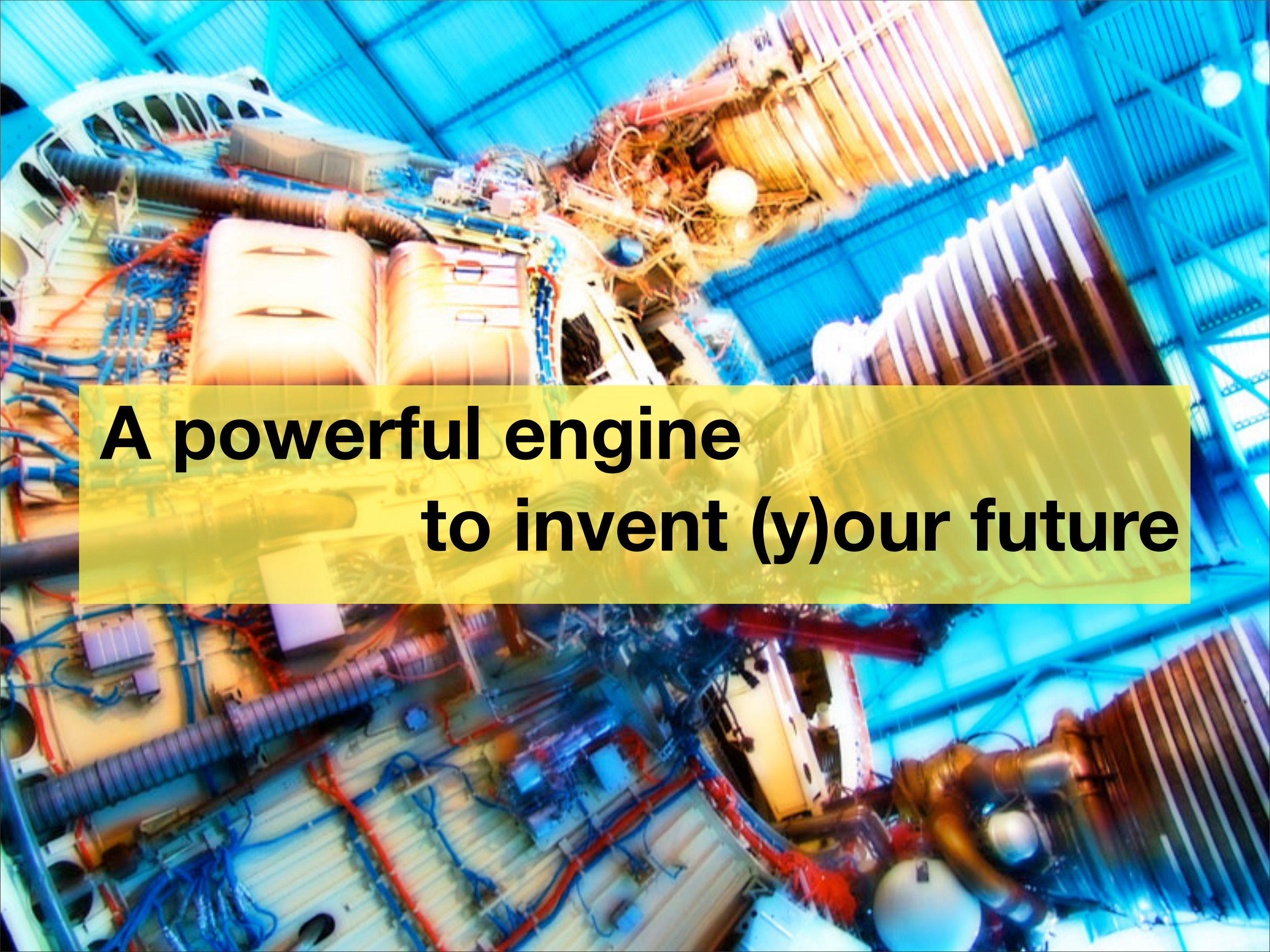
~ **27000 tests**
10k classes
137k methods
61 Mb (64 bits)

<http://github.com/pharo-project/Pharo> (~300
forks, 15/123
international
contributors)


Ecosystem
polymath
pharo-graphics
pharo-gis
pharo-container
pharo-ai



**Pharo is our vehicle
We improve it everyday**

The image shows a large, intricate industrial facility, likely a particle accelerator or fusion reactor. The scene is filled with complex machinery, including large cylindrical components, a dense network of blue and red cables, and various pipes and structural elements. The lighting is a mix of bright blue and yellow, creating a high-tech, futuristic atmosphere. A central yellow semi-transparent box contains the text "A powerful engine to invent (y)our future".

**A powerful engine
to invent (y)our future**



**An ecosystem where
innovation/business bloom**

trentosur

Soluciones móviles para retail y trade marketing

Nos enfocamos en lo que importa del negocio sin perder de vista los detalles de su implementación.

- Primer móvil
- Plataforma Android
- En la nube

PharoCloud

Overview Pricing Blog Login Sign Up

Pharo platform as a Service: put your Smalltalk web-application online at PharoCloud in just 3 clicks

Try it for FREE

Romax TECHNOLOGY

Wind Energy

Pioneering new ways of maximising sustainable wind energy yields. Our products and services optimise asset availability, wind turbine performance and drivetrain reliability. We work with owners, operators, manufacturers, insurers and service providers worldwide.

WEBDRUCK.CH
Web-To-Print Solution

- Design and create individual printed matter
- eShop with credit card payment
- High quality PDF output with Printing Process integration
- Thousands of orders for seven Swiss printing companies

Quuve

Some Success Stories @ pharo.org/

success

Dedicated and cost-effective tools for software evolution

- Dedicated Analyses
- Dedicated tools
- Decision making

NORRIZIK-COM

The world's first online platform fully supporting risk-based test management.

BETTER FASTER CHEAPER

Yesplan is veelzijdige software voor het efficiënt plannen van evenementen.

Yesplan is uiterst gebruiksvriendelijk, flexibel en makkelijk te koppelen met andere software.

CSOB

airflowing

Organize your creative work

Plans and Pricing

Manage your simple way

OBJECT PROFILE

CMSBOX

Das Content Management mit System

100% Inline-Editor

2denker

Continuous API Testing

keep your services under control 24/7

t3

Elegant!

- Full syntax on a postcard
- Simple and powerful objet model

Pharo

```
exampleWithNumber: x

<syntaxOn: #postcard>
"A ""complete"" Pharo syntax"
| y |
true & false not & (nil isNil)
  ifFalse: [ self perform: #add: with: x ].
y := thisContext stack size + super size.
byteArray := #[2 2r100 8r20 16rFF].
{ -42 . #($a #a #'I''m' 'a' 1.0 1.23e2 3.14s2 1) }
do: [ :each |
  | var |
  var := Transcript
  show: each class name;
  show: each printString ].
^ x < y
```

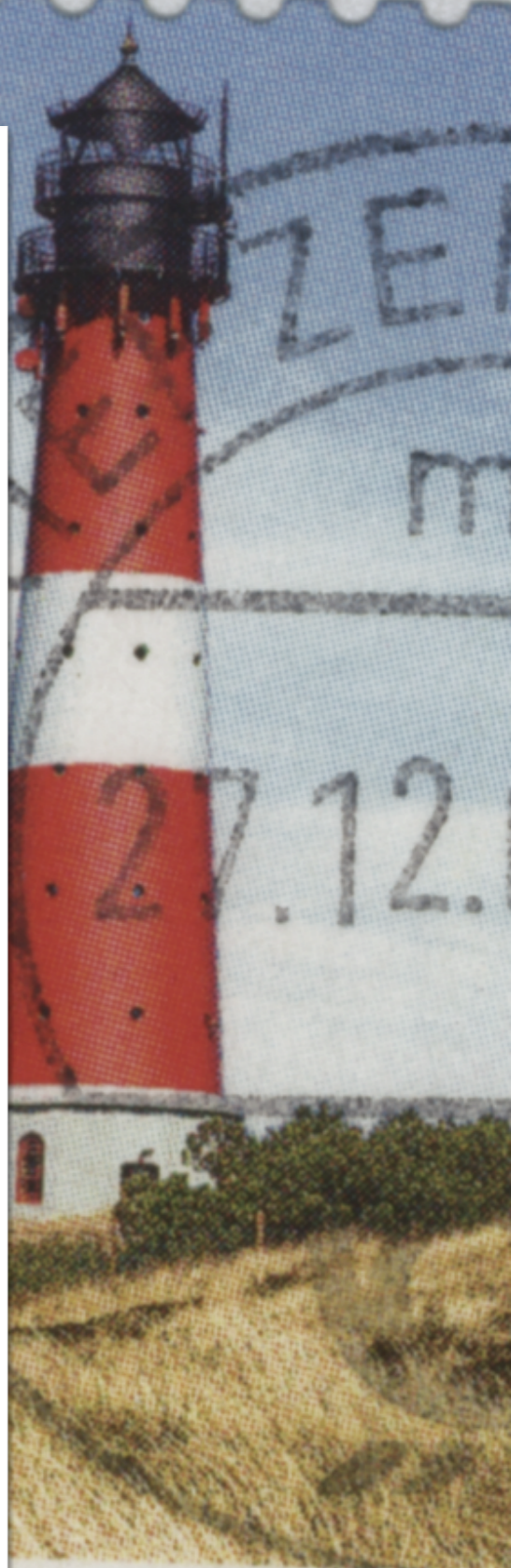
method name, parameter, pragma, comment, local variable, boolean literals, binary message, nil literal, unary message, block, keyword message, pseudo variables, assignment, instance variable, integer literals, byte array, array generated at runtime, literal array, symbols, character, string, scaled decimal, local block variable, block parameter, global variable, cascade, keyword message, return instruction, other method definition examples: unary, + binaryMessageArgument, keyword: arg, keyword: arg1 withTwo: arg2

<https://www.pharo.org>

PLACE
STAMP
HERE

.....
.....
.....
.....

* H Ö R N U M



Yes the full Syntax!

Yes there is nothing else...

Pharo

```
exampleWithNumber: x

<syntaxOn: #postcard>
"A ""complete"" Pharo syntax"
| y |
true & false not & (nil isNil)
  ifFalse: [ self perform: #add: with: x ].
y := thisContext stack size + super size.
byteArray := #[2 2r100 8r20 16rFF].
{ -42 . #($a #a #'I''m' 'a' 1.0 1.23e2 3.14s2 1) }
do: [ :each |
  | var |
  var := Transcript
  show: each class name;
  show: each printString ].
^ x < y
```

method name, parameter, pragma, comment, local variable, boolean literals, binary message, nil literal, unary message, block, keyword message, pseudo variables, assignment, instance variable, integer literals, byte array, array generated at runtime, literal array, symbols, character, string, scaled decimal, local block variable, block parameter, global variable, cascade, keyword message, return instruction

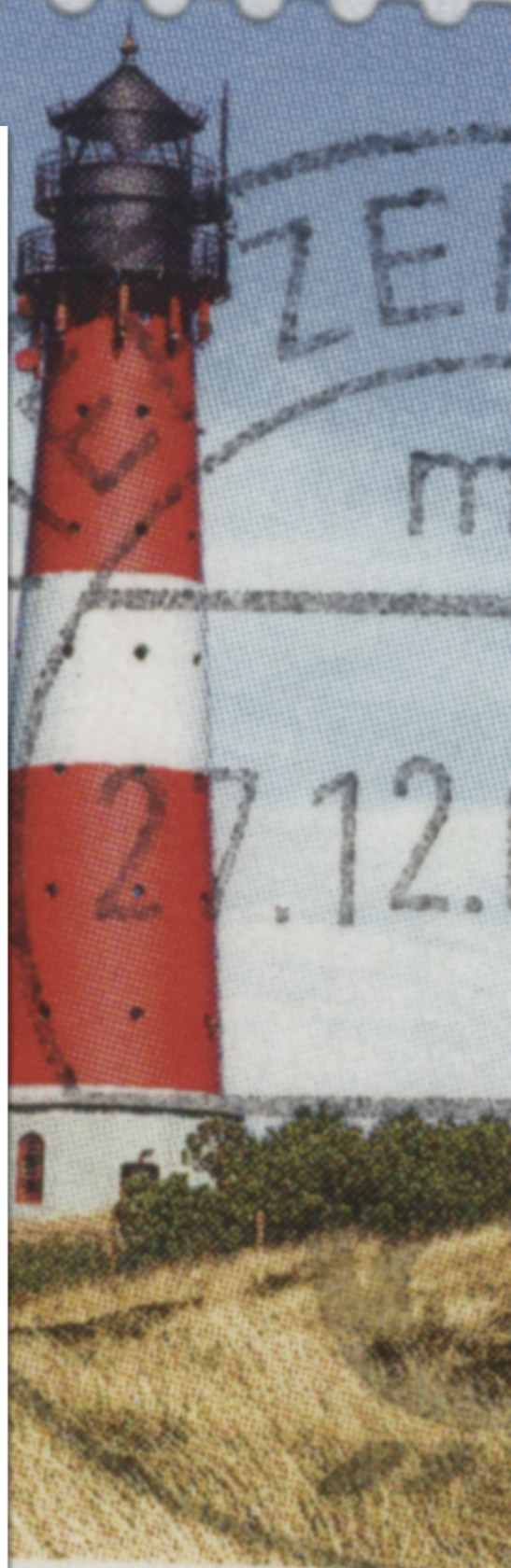
other method definition examples:
unary
+ binaryMessageArgument
keyword: arg
keyword: arg1 withTwo: arg2

<https://www.pharo.org>

PLACE
STAMP
HERE

.....
.....
.....
.....

* H Ö R N U M



exampleWithNumber: x

```
<syntaxOn: #postcard>
```

```
"A ""complete"" Pharo syntax"
```

```
| y |
```

```
true & false not & (nil isNil)
```

```
  ifFalse: [ self perform: #add: with: x ].
```

```
y := thisContext stack size + super size.
```

```
byteArray := #[2 2r100 8r20 16rFF].
```

```
{ -42 . #($a #a #'I' 'm' 'a' 1.0 1.23e2 3.14s2 1) }
```

```
do: [ :each |
```

```
  | var |
```

```
  var := Transcript
```

```
    show: each class name;
```

```
    show: each printString ].
```

```
^ x < y
```

method annotation

exampleWithNumber: **x** parameter

```
<syntaxOn: #postcard>  
"A ""complete"" Pharo syntax" comment
```

```
| y |  
true & false not & (nil isNil)  
ifFalse: [ self perform: #add: with: x ].
```

```
y := thisContext stack size + super size.
```

```
byteArray := #[2 2r100 8r20 16rFF].
```

```
{ -42 . #($a #a #'I' 'm' 'a' 1.0 1.23e2 3.14s2 1) }
```

```
do: [ :each |  
| var |  
var := Transcript  
show: each class name;  
show: each printString ].
```

return

```
^ x < y
```


exampleWithNumber: x **parameter**

<syntaxOn: #postcard>

"A ""complete"" Pharo syntax"

| y |

true & false not & (nil isNil)

ifFalse: [self perform: #add: with: x].

y := thisContext stack size + super size.

byteArray := #[2 2r100 8r20 16rFF].

{ -42 . #(\$a #a #'I' 'm' 'a' 1.0 1.23e2 3.14s2 1) }

do: [:each |

| var |

var := Transcript

show: each class name;

show: each printString].

^ x < y

local var

affectation

instance var

local var

exampleWithNumber: x

```
<syntaxOn: #postcard>
```

```
"A ""complete"" Pharo syntax"
```

```
| y |
```

```
true & false not & (nil isNil)
```

```
  ifFalse: [ self perform: #add: with: x ].
```

closure

```
y := thisContext stack size + super size.
```

```
byteArray := #[2 2r100 8r20 16rFF].
```

```
{ -42 . #($a #a #'I' 'm' 'a' 1.0 1.23e2 3.14s2 1) }
```

```
do: [ :each |
```

arg closure

```
| var |
```

```
var := Transcript
```

```
  show: each class name;
```

```
  show: each printString ].
```

```
^ x < y
```

exampleWithNumber: x

```
<syntaxOn: #postcard>
```

```
"A ""complete"" Pharo syntax"
```

```
| y |
```

binary msg

```
true & false not & (nil isNil)
```

```
ifFalse: [ self perform: #add: with: x ].
```

keyword msg

```
y := thisContext stack size + super size.
```

unary msg

```
byteArray := #[2 2r100 8r20 16rFF].
```

```
{ -42 . #($a #a #'I' 'm' 'a' 1.0 1.23e2 3.14s2 1) }
```

```
do: [ :each |
```

```
| var |
```

```
var := Transcript
```

```
show: each class name;
```

```
show: each printString ].
```

cascade

separator

binary msg

```
^ x < y
```

exampleWithNumber: x

```
<syntaxOn: #postcard>
"A ""complete"" Pharo syntax"
| y |
true & false not & (nil isNil)
  ifFalse: [ self perform: #add: with: x ].
y := thisContext stack size + super size.
byteArray := #[2 2r100 8r20 16rFF].
{ -42 . #($a #a #'I' 'm' 'a' 1.0 1.23e2 3.14s2 1) }
do: [ :each |
  | var |
  var := Transcript
  show: each class name;
  show: each printString ].
```

array character

symbol
string

number

^ x < y

A Pure World of Objects

Only

objects + messages +

closures

mouse, booleans, arrays, numbers, strings, windows, scrollbars, canvas, files, trees, compilers, sound, url, socket, fonts, text, collections, stack, shortcut, streams, ...

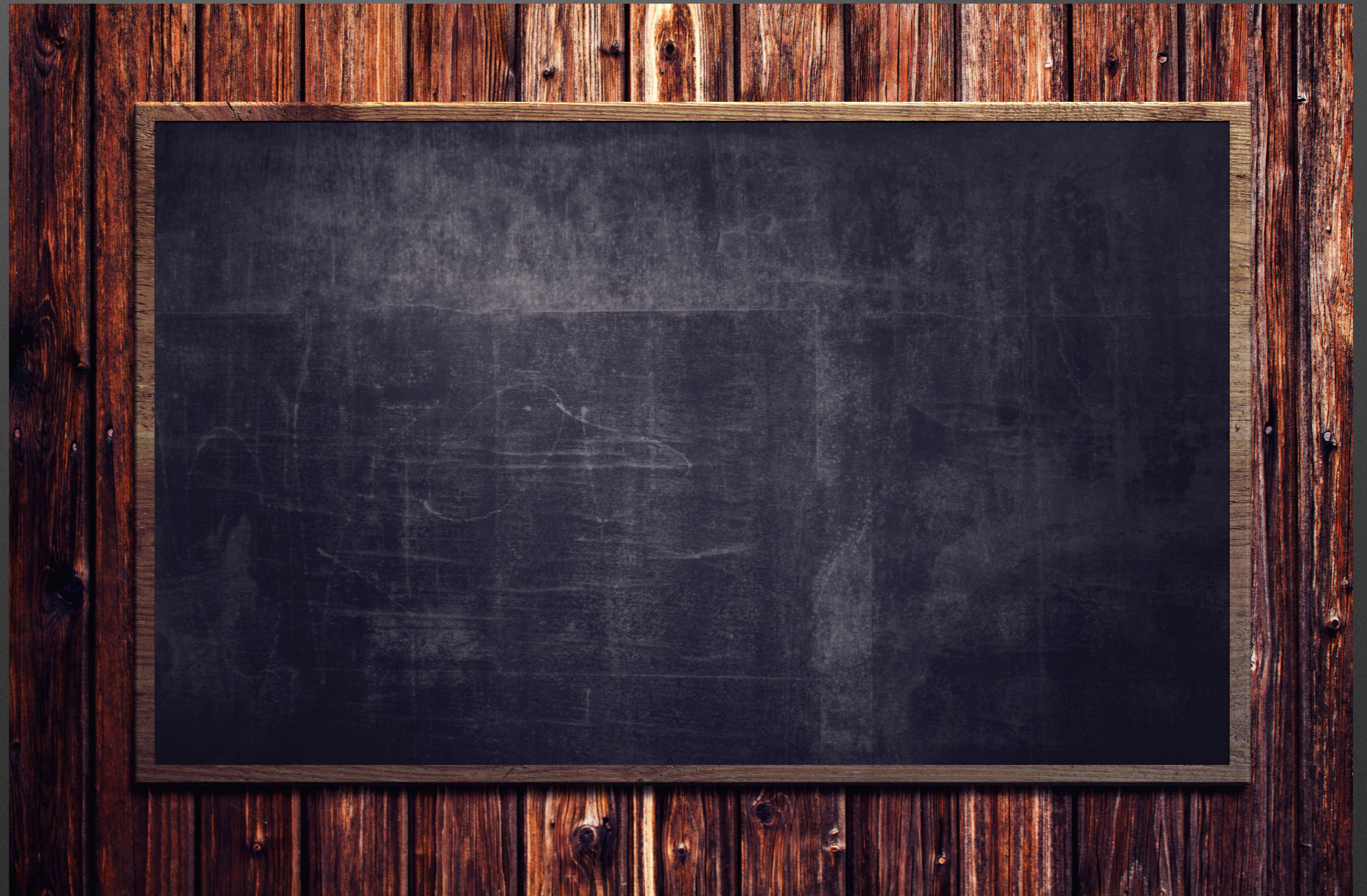
A Fully Uniform Model

- Dynamically typed
- **Everything** is an object instance of a class
- All methods are public **virtual**
- All attributes are **protected**
- **Single** inheritance with **traits**

Pharo is highly immersive

**if a compiler would be an
aquarium**

**compiling a program would
like writing on a black board**



A large aquarium tank filled with water, featuring a large spotted shark swimming in the center. Numerous smaller fish are scattered throughout the tank. In the foreground, the silhouettes of several people are visible, looking at the tank. The text "Immersing..." is overlaid in the top left corner.

Immersing...

Pharo is not a blackbox

Everything is **fully inspectable** and
reflective

**You are immersed and
interacting with objects**



Workspace

```
| elements lay |
```

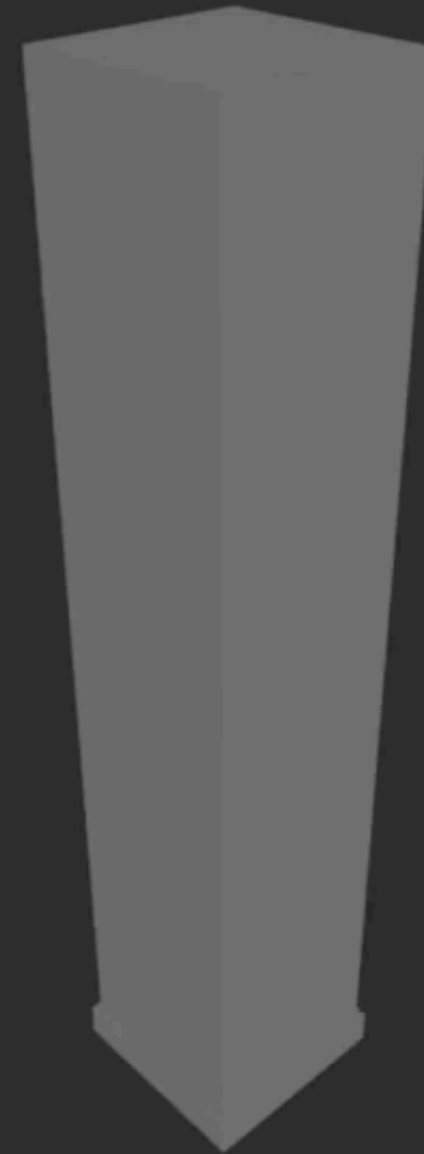
```
elements := (1 to: 5) collect: [ :ob |  
  (R3CubeShape new) elementOn: ob ].
```

```
I
```

```
lay := R3WallLayout new.  
lay on: elements.
```

```
UberPresenter present: elements
```

Uber Presenter



**We can do the same with
web app, sockets, networks,
sensors, living programming....**



**Hackers
scripting live
the coffee
machine**



Selected Infrastructure

Fully Written in Itself



Selected features

- First class instance variables (daemons, relationships...)
- Fast resumable exceptions
- Runtime classes and objects migration
- Customizable compiler
- Serializable and shareable execution stack
- Optional system virtualization
- Fully bootstrapped kernel(s) (down to 200kb)

Advanced reflective layer

- Versatile AST annotations and transformations @ runtime
- Full stack reification (continuations, exceptions...)
- Instance enumeration
- Causally connected “Software as Objects”
- Atomic bulk object swapping
- ... more but no time for that

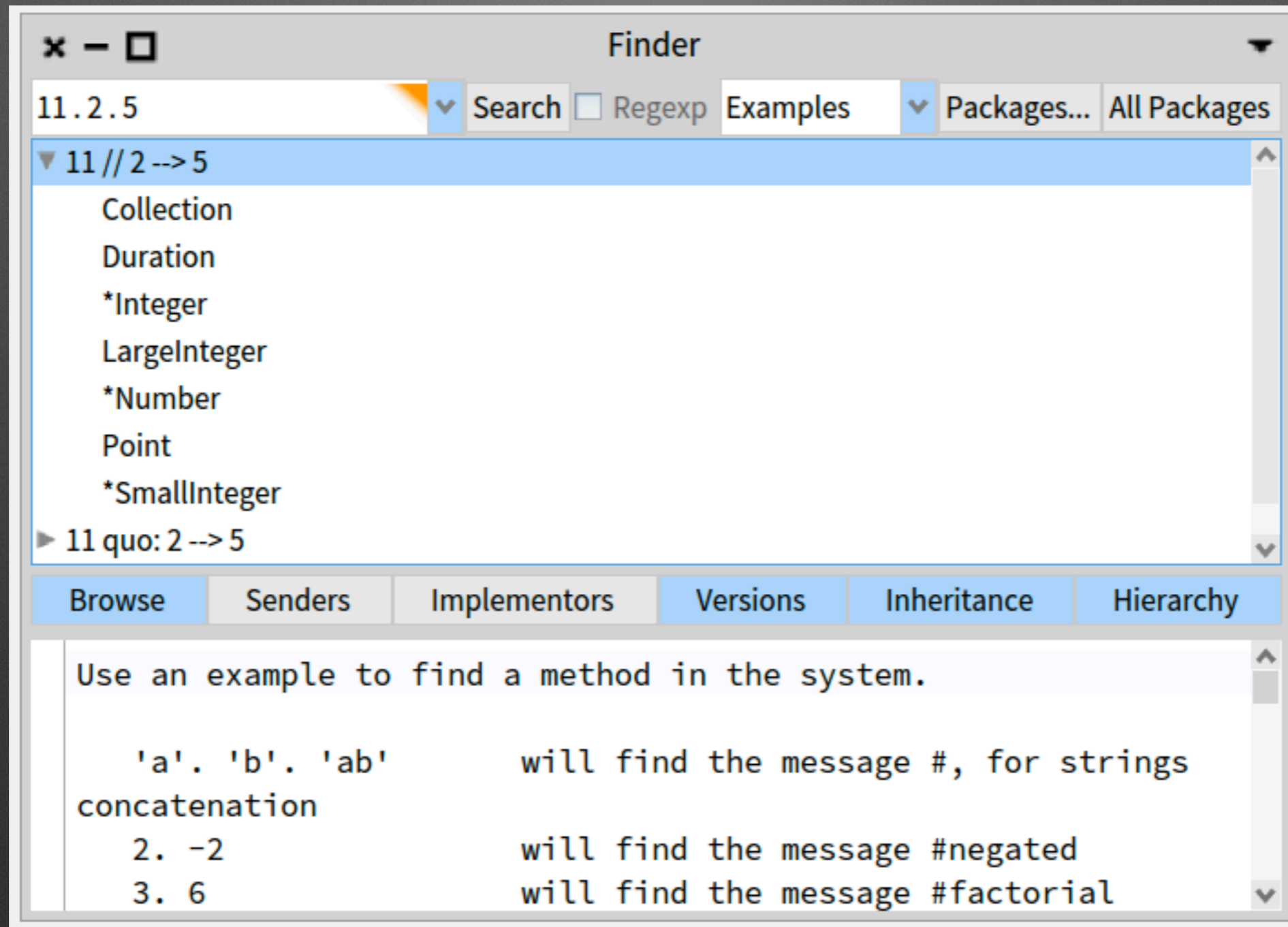


**Tools are our friends
But they shape our mind**

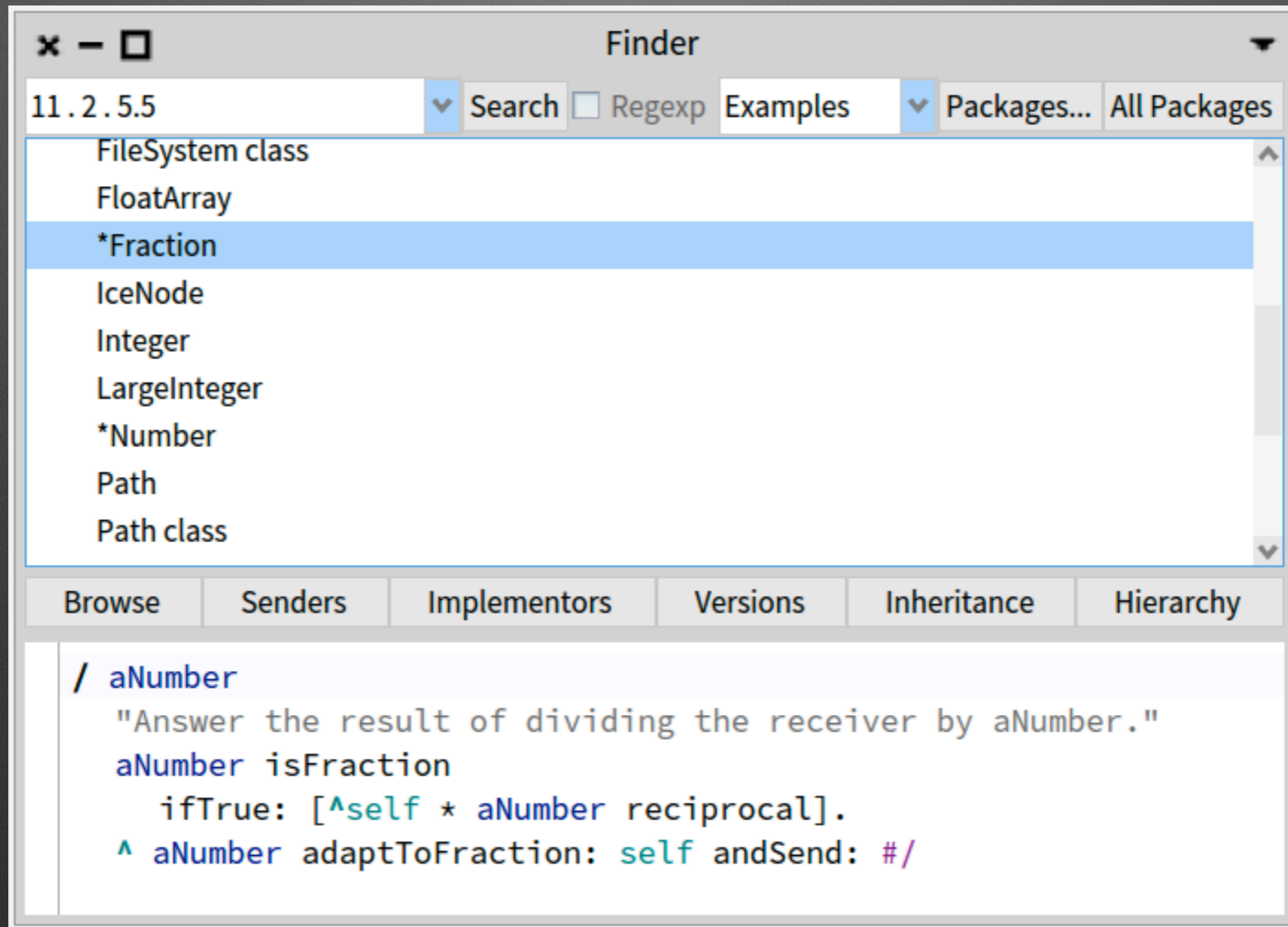
How to find information?

- Libraries are large
- You know what you want
- You do not know how to express it

11 ??? 2 should give 5



11 ??? 2 should give 5.5



The screenshot shows a window titled "Finder" with a search bar containing "11.2.5.5". The search results list several classes, with "*Fraction" selected. Below the list are tabs for "Browse", "Senders", "Implementors", "Versions", "Inheritance", and "Hierarchy". The "Browse" tab is active, displaying the following code snippet:

```
/ aNumber  
"Answer the result of dividing the receiver by aNumber."  
aNumber isFraction  
  ifTrue: [^self * aNumber reciprocal].  
^ aNumber adaptToFraction: self andSend: #/
```

What are the messages send to \$0 that return true

The screenshot shows the Ruby Finder tool with the search query '\$0.isDigit'. The results list several messages that return true for the receiver '\$0'. The 'Character' class is highlighted, and the 'isDigit' method is expanded to show its implementation and test cases.

Finder

\$0.isDigit Search Regexp Examples Packages... All Packages

- ▶ \$0.isAlphaNumeric --> true
- ▶ \$0.isCharacter --> true
- ▶ \$0.isCompletionCharacter --> true
- ▶ \$0.isDecimalDigit --> true
- ▼ \$0.isDigit --> true

Character

- ▶ \$0.isLiteral --> true
- ▶ \$0.isOctetCharacter --> true
- ▶ \$0.isSafeForHTTP --> true
- ▶ \$0.shouldBePrintedAsLiteral --> true
- ▶ \$0.tokenish --> true

Browse Senders Implementors Versions Inheritance Hierarchy

isDigit

```
"Return whether the receiver is a digit."  
"$1.isDigit >>> true"  
"$0.isDigit >>> true"  
  
^ self.characterSet.isDigit: self
```

Tools

- Shape our mind....
- Pharo has moldable tools: **you CAN adapt them to you and your process** and not the inverse
- Build fast **your own** tools

Pharo has **amazing**
moldable tools

**Customizable object
interaction/presentations**

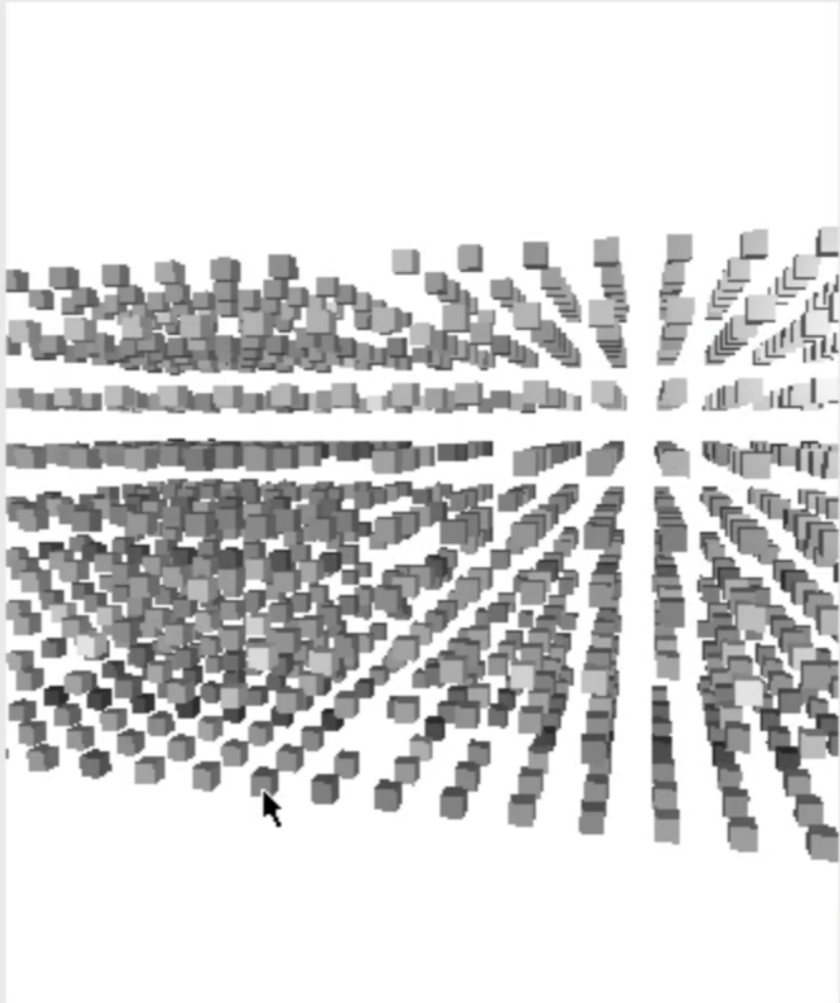
Inspecting live a 3D object

Playground

```
data := TestData new data.  
  
cube := MatrixCube new initWithData: data.  
cube view
```

a RWView

Raw Live Meta



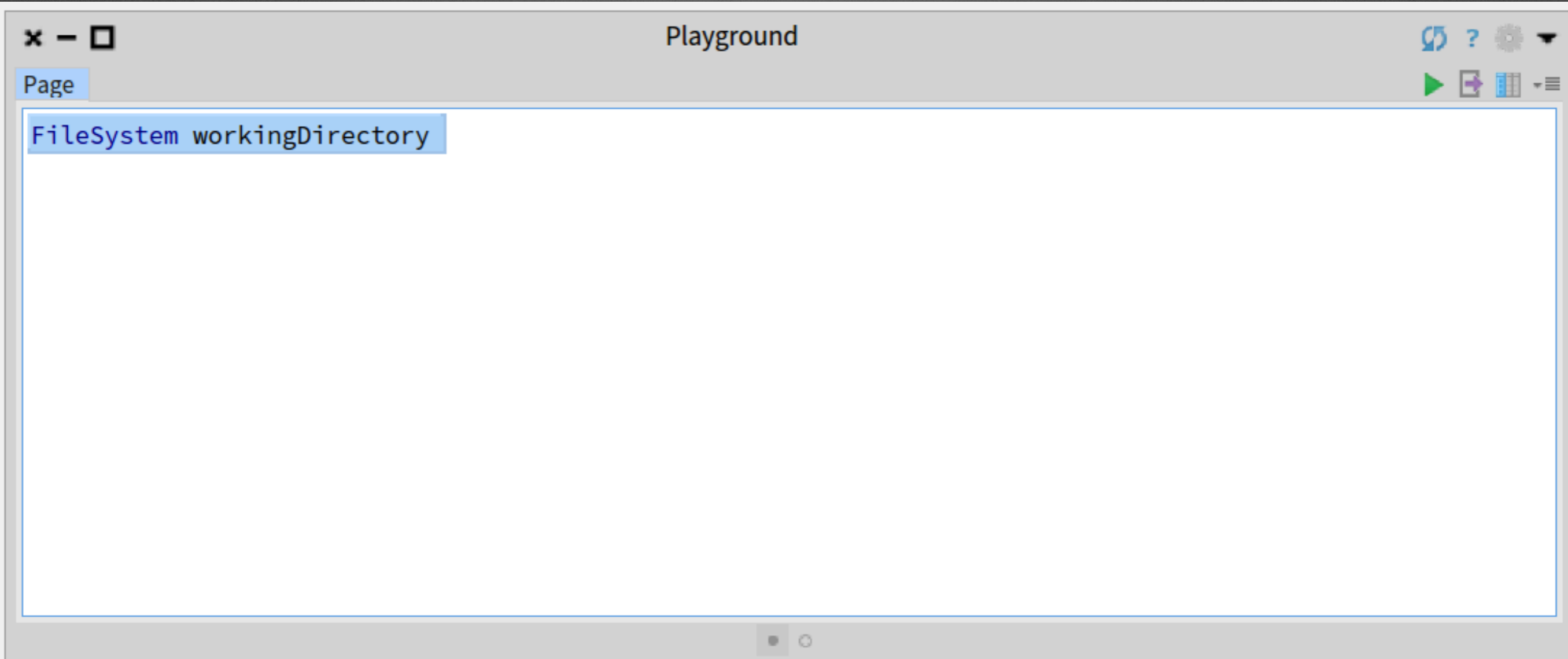
an Array [4 items] ('2000-01-01' 'Malawi' 'France'... x

Index	Item
1	'2000-01-01'
2	'Malawi'
3	'France'
4	'400.0'

enter search query (example: "each = 5")

The views of a file reference

Looking at a file reference



Oh! a file browser in my inspector!

The image shows a web browser window titled "Playground" with a file browser interface. The browser's address bar shows a file reference: `/Users/ducasse/Documents/Pharo/images/F...`. The file browser interface is divided into two main sections: a console on the left and a file listing on the right.

The console on the left displays the text `FileSystem workingDirectory`. The file listing on the right shows a directory structure with columns for Name, Size, and a third column (likely permissions or type). The files listed are:

Name	Size	
..	0 B	2
pharo-local	0 B	2
logo.png	25.82 kB	2
pharo.version	3 B	2
ReadMe.txt	63 B	2
meta-inf.ston	1.17 kB	2
P8-MasterClass.image	70.22 MB	2
P8-MasterClass.changes	1.16 kB	2
Archive.zip	27.24 kB	2
Pharo8.0-32bit-0932da8.sources	37.94 MB	2

But I have a file reference: a dull object

The screenshot shows a Playground IDE window with a code editor on the left and a variable inspector on the right. The code editor contains the text `FileSystem workingDirectory`. The variable inspector shows a `FileReference` object with the following properties:

Variable	Value
self	/Users/ducasse/Documents/Pharo/images/I
filesystem	a FileSystem
path	Path / 'Users' / 'ducasse' / 'Documents' / 'Ph

Below the table, the object's class and parent are shown:

```
"File @  
/Users/ducasse/Documents/Pharo/images/P8-MasterCl  
ass"  
self parent File @ /Users/ducasse/Documents/Pharo/images
```

Quite boring object

The screenshot shows a Squeak Playground window with the following components:

- Code Area:** Contains the code `FileSystem workingDirectory`.
- Inspector:** Shows the object `a FileReference (/Users/ducasse/Documents/Pharo/images/F...`.
- Inspector Table:** A table with two columns: Variable and Value.
- Inspector Text:** Shows the object's class and a specific instance variable.

Variable	Value
self	/Users/ducasse/Documents/Pharo/images/I
filesystem	a FileSystem
path	Path / 'Users' / 'ducasse' / 'Documents' / 'Ph

"File @
/Users/ducasse/Documents/Pharo/images/P8-MasterCl
ass"
self fullName '/Users/ducasse/Documents/Pharo/images/P8-MasterClass'

We can see the png ;)


Playground

a FileReference (/Users/ducasse/Documents/Pharo/images/P8-M...)

Items Raw Meta

Name	Size	
..	0 B	2
pharo-local	0 B	2
logo.png	25.82 kB	2
pharo.version	3 B	2
ReadMe.txt	63 B	2
meta-inf.ston	1.17 kB	2
P8-MasterClass.image	70.22 MB	2
P8-MasterClass.changes	1.16 kB	2
Archive.zip	27.24 kB	2
Pharo8.0-32bit-0932da8.sources	37.94 MB	2

Picture Contents Raw Meta



The image shows a screenshot of a 'Playground' application window. The window title is 'Playground'. The address bar shows a file reference: 'a FileReference (/Users/ducasse/Documents/Pharo/images/P8-M...)'.

On the left, there is a file browser with tabs 'Items', 'Raw', and 'Meta'. The 'Items' tab is active, showing a list of files and folders:

Name	Size	
..	0 B	2
pharo-local	0 B	2
logo.png	25.82 kB	2
pharo.version	3 B	2
ReadMe.txt	63 B	2
meta-inf.ston	1.17 kB	2
P8-MasterClass.image	70.22 MB	2
P8-MasterClass.changes	1.16 kB	2
Archive.zip	27.24 kB	2
Pharo8.0-32bit-0932da8.sources	37.94 MB	2

On the right, there is a preview pane with tabs 'Picture', 'Contents', 'Raw', and 'Meta'. The 'Picture' tab is active, displaying the 'Pharo' logo. The logo consists of the word 'Pharo' in a blue, italicized font, followed by a circular icon containing a stylized lighthouse with a red top and a white body with blue stripes.

Looking inside that PNG file

The image shows a software interface with two main panes. The left pane is a file explorer titled 'Playground' showing a directory listing. The right pane is a hex editor showing the raw bytes of a selected file, with a 'Contents' tab active.

File Explorer (Left Pane):

Name	Size
..	0 B
pharo-local	0 B
logo.png	25.82 kB
pharo.version	3 B
ReadMe.txt	63 B
meta-inf.ston	1.17 kB
P8-MasterClass.image	70.22 MB
P8-MasterClass.changes	1.16 kB
Archive.zip	27.24 kB
Pharo8.0-32bit-0932da8.sources	37.94 MB

Hex Editor (Right Pane):

Contents tab active. The hex data is displayed in a table with line numbers 1 through 7.

Line	Hex	ASCII
1	00000000 89 50 4E 47 0D 0A 1A 0A 00 00 00	
2	0D 49 48 44 52 .PNG.....IHDR	.PNG.....IHDR
3	00000010 00 00 01 77 00 00 00 90 08 06 00	
4	00 00 F3 F6 2B ...w.....+	...w.....+
5	00000020 70 00 00 0A D1 69 43 43 50 49 43	
6	43 20 50 72 6F p....iCCPICC Pro	p....iCCPICC Pro
7	00000030 66 69 6C 65 00 00 48 89 95 97 07	
8	54 53 69 16 C7 file..H....TSi..	file..H....TSi..
9	00000040 BF F7 D2 43 42 4B 08 45 4A E8 4D	
10	90 5E A5 84 1E ...CBK.EJ.M.^...	...CBK.EJ.M.^...
11	00000050 40 41 3A D8 08 49 48 42 09 21 05	
12	15 3B 32 38 02 @A:...IHB.!...;28.	@A:...IHB.!...;28.
13	00000060 23 8A 8A 08 96 01 1D 8A 28 38 16	

But still a file reference!

The screenshot shows a 'Playground' window with two panes. The left pane displays a file browser view with a table of files. The right pane shows the 'Raw' view of a selected file, displaying its metadata and full path.

Name	Size	Count
..	0 B	2
pharo-local	0 B	2
logo.png	25.82 kB	2
pharo.version	3 B	2
ReadMe.txt	63 B	2
meta-inf.ston	1.17 kB	2
P8-MasterClass.image	70.22 MB	2
P8-MasterClass.changes	1.16 kB	2
Archive.zip	27.24 kB	2
Pharo8.0-32bit-0932da8.sources	37.94 MB	2

Variable	Value
self	/Users/ducasse/Documents/Pharo/images/...
filesystem	a FileSystem
path	Path / 'Users' / 'ducasse' / 'Documents' / 'Ph...

"File @
/Users/ducasse/Documents/Pharo/images/P8-MasterCl
ass/logo.png"
self fullName '/Users/ducasse/Documents/Pharo/images/P8-MasterClass/logo.png'

See! an archive '.zip'

The image shows a screenshot of a file explorer window titled "Playground" and a hex editor window. The file explorer window displays a directory listing with columns for Name, Size, and a third column. The files listed are: .. (0 B), pharo-local (0 B), logo.png (25.82 kB), pharo.version (3 B), ReadMe.txt (63 B), meta-inf.ston (1.17 kB), P8-MasterClass.image (70.22 MB), P8-MasterClass.changes (1.16 kB), Archive.zip (27.24 kB), and Pharo8.0-32bit-0932da8.sources (37.94 MB). The "Archive.zip" file is highlighted. The hex editor window shows the raw contents of the selected file, displaying hexadecimal values and their corresponding ASCII characters. The first line of the hex editor shows the ZIP magic number: 50 4B 03 04 14 00 08 00 08 00 E1 AA 16 51 00 00 |PK.....Q..|. The second line shows the file name "go.png" in ASCII: 00 10 00 6C 6F |.....lo|. The third line shows the file name "go.png" in ASCII: 70 41 5F 96 70 |go.pngUX...pA_.p|. The fourth line shows the file name "go.png" in ASCII: 41 B3 25 DA B6 |A_....l.c.hA.%..|. The fifth line shows the file name "go.png" in ASCII: DA B6 6D DB B6 |m..m...m..y..m..|. The sixth line shows the file name "go.png" in ASCII: 8F 8A 55 99 2B |m[. |...y.....U.+|. The seventh line shows the file name "go.png" in ASCII: 57 46 66 C4 DE 55 19 A1 20 27 0E.

Playground

a FileReference (/Users/ducasse/Documents/Pharo/images/P8-M...)

Items Raw Meta

Name	Size	
..	0 B	2
pharo-local	0 B	2
logo.png	25.82 kB	2
pharo.version	3 B	2
ReadMe.txt	63 B	2
meta-inf.ston	1.17 kB	2
P8-MasterClass.image	70.22 MB	2
P8-MasterClass.changes	1.16 kB	2
Archive.zip	27.24 kB	2
Pharo8.0-32bit-0932da8.sources	37.94 MB	2

a FileReference (/Users/ducasse/Documents/Pharo/images/F...)

Items Contents Raw Meta

```
1 00000000 50 4B 03 04 14 00 08 00 08 00 E1
AA 16 51 00 00 |PK.....Q..|
2 00000010 00 00 00 00 00 00 00 00 00 00 08
00 10 00 6C 6F |.....lo|
3 00000020 67 6F 2E 70 6E 67 55 58 0C 00 A4
70 41 5F 96 70 |go.pngUX...pA_.p|
4 00000030 41 5F F7 01 14 00 6C B7 63 90 68
41 B3 25 DA B6 |A_....l.c.hA.%..|
5 00000040 6D DB B6 6D BB FB B4 6D DB B6 79
DA B6 6D DB B6 |m..m...m..y..m..|
6 00000050 6D 5B EF 7C F7 DE 99 79 F1 DE D4
8F 8A 55 99 2B |m[. |...y.....U.+|
7 00000060 57 46 66 C4 DE 55 19 A1 20 27 0E
```

Kind of clear...

The image shows a screenshot of a file explorer window titled "Playground". The window is split into two panels, each displaying a directory listing. The left panel shows a directory with the following files and sizes:

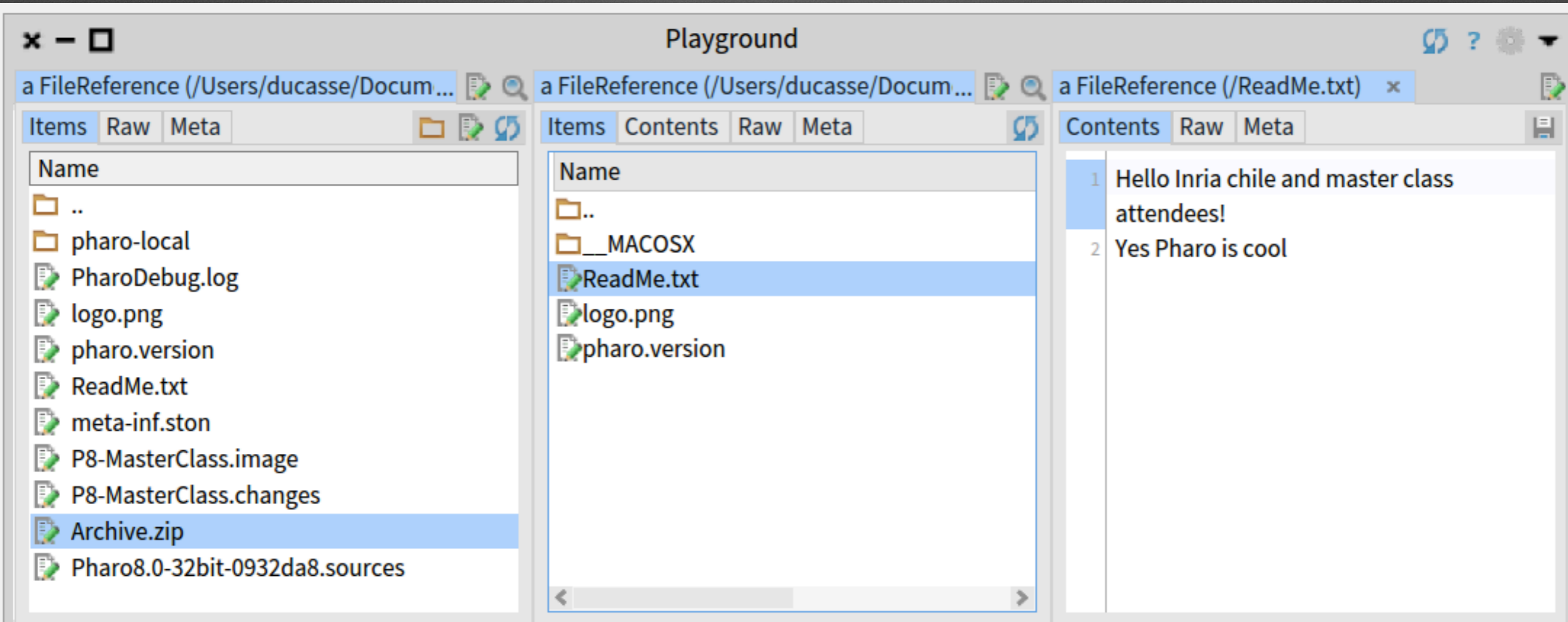
Name	Size
..	0 B
pharo-local	0 B
logo.png	25.82 kB
pharo.version	3 B
ReadMe.txt	63 B
meta-inf.ston	1.17 kB
P8-MasterClass.image	70.22 MB
P8-MasterClass.changes	1.16 kB
Archive.zip	27.24 kB
Pharo8.0-32bit-0932da8.sources	37.94 MB

The right panel shows a directory with the following files and sizes:

Name	Size
..	0 B
__MACOSX	0 B
ReadMe.txt	0 B
logo.png	0 B
pharo.version	0 B

An object can expose multiple interactive views!

- You can use the best view for your task!
- You can add views to your domain objects



It is cool but it is not magic
You can define your own

Implementing a pane!

The screenshot shows the Smalltalk IDE interface for the method `inspectionJpeg` in the class `AbstractFileReference`. The interface is divided into several panes:

- Left Pane:** A package browser showing a tree of packages under `Public`, including `Extensions`, `FileSystem-Core-Tests`, `FileSystem-Disk`, `FileSystem-Disk-Tests`, `FileSystem-Memory`, `FileSystem-Memory-Tests`, `FileSystem-Path`, `FileSystem-Tests-Attributes`, `FileSystem-Zip`, `Files`, `Files-Prompt`, and `Files-Taste`.
- Middle Pane:** A class browser showing the hierarchy of `AbstractFileReference`, including subclasses like `FileLocator`, `FileReference`, `FileSystem`, `FileSystemDirectoryEntry`, `DiskDirectoryEntry`, `DiskSymlinkDirectoryEntry`, `MemoryDirectoryEntry`, `FileSystemPermission`, and `FileVersionner`.
- Right Pane:** A list of extensions for the `inspectionJpeg` method, including `inspectionCompressedItems`, `inspectionCompressedItemsContext`, `inspectionContents`, `inspectionContentsContext`, `inspectionFuelContext`, `inspectionGif`, `inspectionGifContext`, `inspectionItems`, `inspectionItemsContext`, `inspectionJpeg` (highlighted), `inspectionJpegContext`, `inspectionPng`, `inspectionPngContext`, and `inspectionStScript`.
- Bottom Pane:** A code editor showing the implementation of the `inspectionJpeg` method:

```
inspectionJpeg
  <inspectorPresentationOrder: 0 title: 'Picture'>

  ^ SpImagePresenter new
    image: (self binaryReadStreamDo: [ :stream | ImageReadWrite formFromStream: stream ]);
    yourself
```

At the bottom of the IDE, there is a status bar showing the current package `NewTools-Inspector-Extensions`, a checked checkbox for `extension`, and a keyboard shortcut `F +L W`.

Files are boring...
What about
inside
the
system?



A class is an object we can inspect!

The screenshot shows a Python Playground window with the following components:

- Page:** A text area containing the word "Point".
- Inspector:** A table showing the internal state of the "a Point class (Point)" object.

Variable	Value
self	Point
superclass	Object
methodDict	a MethodDictionary [103 items] (size 103)
format	65538
layout	a FixedLayout
organization	a ClassOrganization
subclasses	nil
name	#Point
classPool	a Dictionary [0 items] /

Below the table, the object's string representation is shown as "Point" and the value of the self variable is shown as self.

“A class has a method dictionary”
they said... let us verify

The screenshot shows a Ruby Playground window with two panes. The left pane displays the class structure of `Point`, and the right pane displays the contents of its `methodDict`.

Left Pane: a Point class (Point)

Variable	Value
self	Point
superclass	Object
methodDict	a MethodDictionary [103 items] (size 103)
format	65538
layout	a FixedLayout
organization	a ClassOrganization
subclasses	nil
name	#Point
classPool	a Dictionary [0 items] (size 0)

Below the table, the source code for the `Point` class is shown:

```
"Point"  
self
```

Right Pane: a MethodDictionary [103 items] (size 103)

Key	Value
#reflectedAbout:	Point>>#reflectedAbout:
#rotateBy:centerAt:	Point>>#rotateBy:centerAt:
#adaptToNumber:andSend:	Point>>#adaptToNumber:andSend:
#squaredDistanceTo:	Point>>#squaredDistanceTo:
#adaptToCollection:andSend:	Point>>#adaptToCollection:andSend:
#theta	Point>>#theta
#transposed	Point>>#transposed
#-	Point>>#-
#fourDirections	Point>>#fourDirections
#crossProduct:	Point>>#crossProduct:
#scaleFrom:to:	Point>>#scaleFrom:to:
#veryDeepCopyWith:	Point>>#veryDeepCopyWith:

Dissecting one method object

The screenshot shows the Ruby Inspector tool window titled "Inspector on a CompiledMethod (Point>>#degrees)". The main content area displays the internal structure of the method object, organized into a table with "Variable" and "Value" columns. The variables listed are self, literal1 through literal8, and bc 89. The values are Point>>#degrees, 90.0, 270.0, #asFloat, #arcTan, #radiansToDegrees, 360.0, 180.0, #ifTrue:ifFalse:, and 0. Below the table, the source code for the method is shown, starting with "Point>>#degrees" and "self".

Variable	Value
{ } self	Point>>#degrees
▶ Σ literal1	90.0
▶ Σ literal2	270.0
▶ ¶ literal3	#asFloat
▶ ¶ literal4	#arcTan
▶ ¶ literal5	#radiansToDegrees
▶ Σ literal6	360.0
▶ Σ literal7	180.0
▶ ¶ literal8	#ifTrue:ifFalse:
▶ Σ bc 89	0

```
"Point>>#degrees"  
self
```

I do not want to be a compiler!

Inspector on a CompiledMethod (Point>>#degrees)

a CompiledMethod (Point>>#degrees)

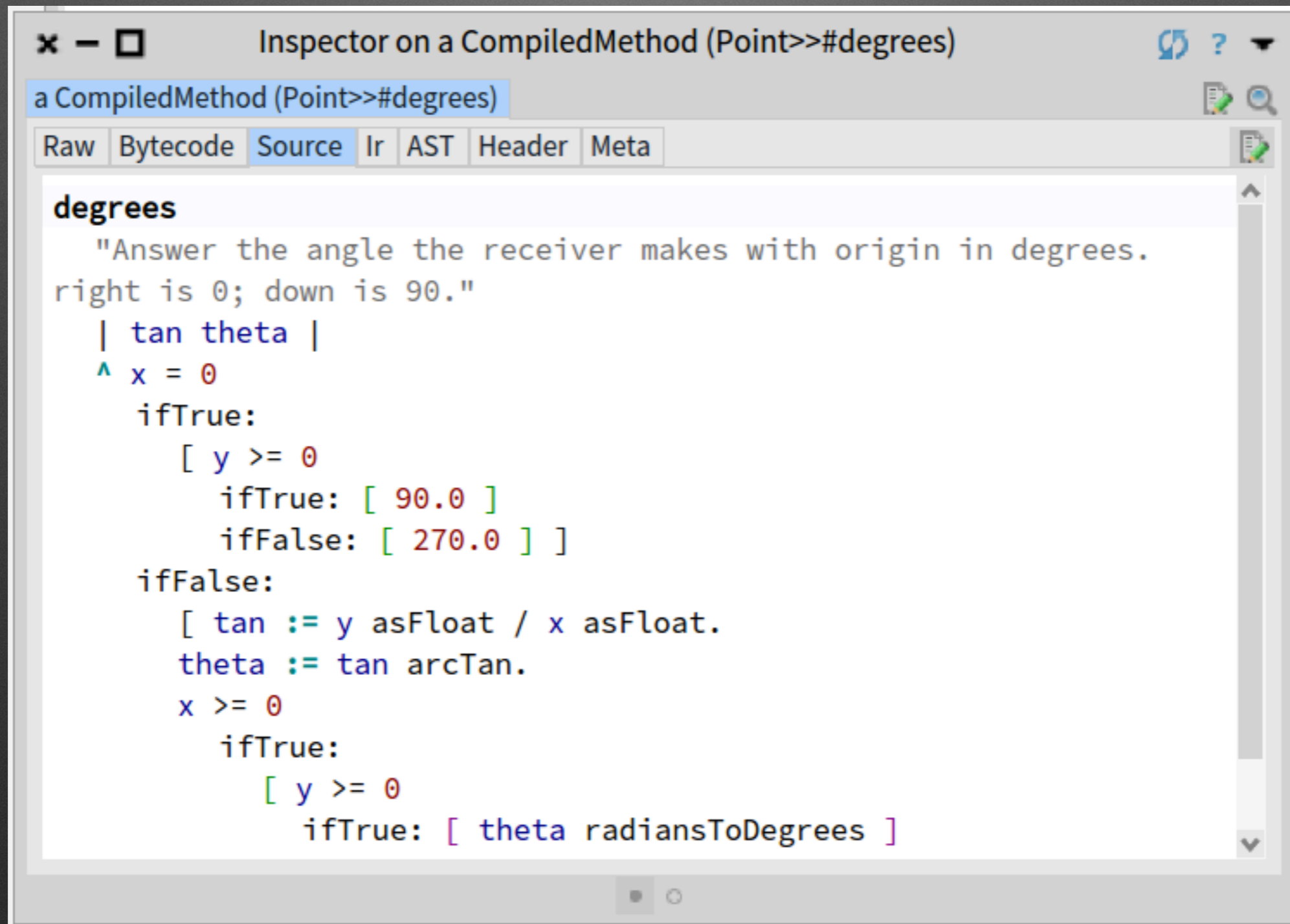
Raw Bytecode Source Ir AST Header Meta

Variable	Value
▶ Σ bc 89	0
▶ Σ bc 90	117
▶ Σ bc 91	182
▶ Σ bc 92	172
▶ Σ bc 93	9
▶ Σ bc 94	1
▶ Σ bc 95	117
▶ Σ bc 96	181
▶ Σ bc 97	153
▶ Σ bc 98	32

"Point>>#degrees"

```
self
```

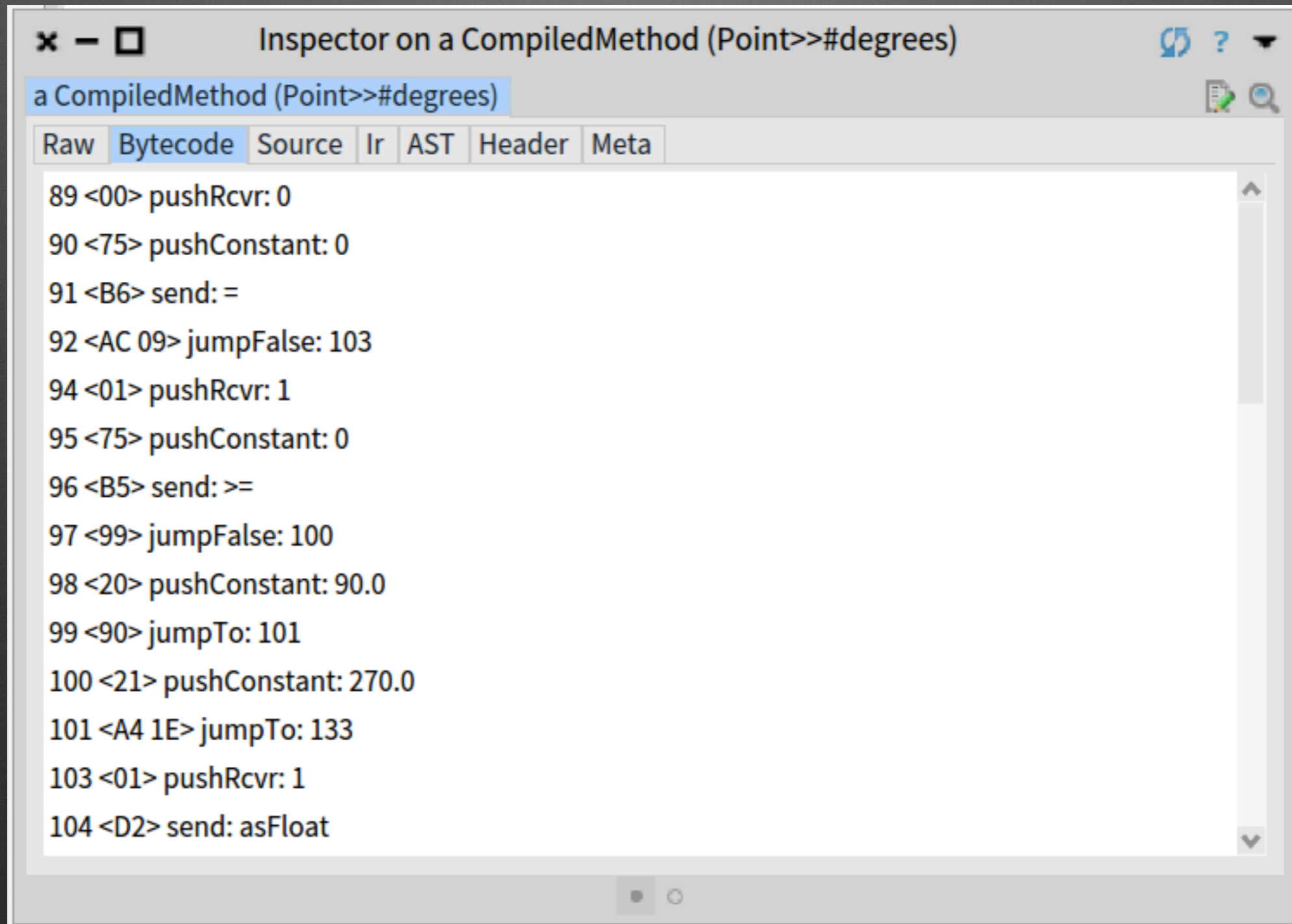
It looks like a method



The screenshot shows an IDE window titled "Inspector on a CompiledMethod (Point>>#degrees)". The window contains a tabbed interface with "Source" selected. The source code for the "degrees" method is displayed, featuring a docstring and several conditional branches based on the sign of the x and y coordinates.

```
degrees
  "Answer the angle the receiver makes with origin in degrees.
  right is 0; down is 90."
  | tan theta |
  ^ x = 0
  ifTrue:
    [ y >= 0
      ifTrue: [ 90.0 ]
      ifFalse: [ 270.0 ] ]
  ifFalse:
    [ tan := y asFloat / x asFloat.
      theta := tan arcTan.
      x >= 0
        ifTrue:
          [ y >= 0
            ifTrue: [ theta radiansToDegrees ]
```

Numbers are not that obscure



The screenshot shows a debugger window titled "Inspector on a CompiledMethod (Point>>#degrees)". The window displays the bytecode for a method named "degrees". The bytecode is organized into a table with columns for "Raw", "Bytecode", "Source", "Ir", "AST", "Header", and "Meta". The "Bytecode" column is currently selected, showing the following instructions:

```
89 <00> pushRcvr: 0
90 <75> pushConstant: 0
91 <B6> send: =
92 <AC 09> jumpFalse: 103
94 <01> pushRcvr: 1
95 <75> pushConstant: 0
96 <B5> send: >=
97 <99> jumpFalse: 100
98 <20> pushConstant: 90.0
99 <90> jumpTo: 101
100 <21> pushConstant: 270.0
101 <A4 1E> jumpTo: 133
103 <01> pushRcvr: 1
104 <D2> send: asFloat
```

And mapping them to the good abstraction helps

The image shows a screenshot of an IDE's AST inspector for a compiled method. The left pane shows the AST tree for `a CompiledMethod (Point>>#degrees)`. The right pane shows the source code for `a RbMessageNode (RbMessageNode(y >= 0))`.

Inspector on a CompiledMethod (Point>>#degrees)

a CompiledMethod (Point>>#degrees)

- Raw
- Bytecode
- Source
- Ir
- AST
- Header
- Meta

▼ RBMethodNode(degrees "Answer the angle the receiver makes with")

- ▼ RBSequenceNode(| tan theta | ^ x = 0 ifTrue: [y >= 0 ifTrue:)

 - RBTemporaryNode(tan)
 - RBTemporaryNode(theta)

▼ RBReturnNode(^ x = 0 ifTrue: [y >= 0 ifTrue: [90.0]

- ▼ RBMessageNode(x = 0 ifTrue: [y >= 0 ifTrue: [90.0]

 - ▼ RBMessageNode(x = 0)

 - RBInstanceVariableNode(x)
 - RBLiteralValueNode(0)

▼ RBBlockNode([y >= 0 ifTrue: [90.0] ifFalse: [270.0]])

- ▼ RBSequenceNode(y >= 0 ifTrue: [90.0] ifFalse: [270.0])

 - ▼ RBMessageNode(y >= 0 ifTrue: [90.0] ifFalse: [270.0])

 - ▶ RBMessageNode(y >= 0)
 - ▶ RBBlockNode([90.0])
 - ▶ RBBlockNode([270.0])

a RbMessageNode (RbMessageNode(y >= 0))

- Raw
- Tree
- Scopes
- Source cc...
- AST Dump
- Meta

degrees

"Answer the angle the receiver makes with origin in degrees. right is 0; down is 90."

```
| tan theta |  
^ x = 0  
ifTrue:  
  [ y >= 0  
    ifTrue: [ 90.0 ]  
    ifFalse: [ 270.0 ] ]  
ifFalse:  
  [ tan := y asFloat / x asFloat.  
    theta := tan arcTan.  
    x >= 0  
      ifTrue:  
        [ y >= 0  
          ifTrue: [ theta radiansToDegrees ]  
          ifFalse: [ 360.0 + theta  
                    radiansToDegrees ] ]
```


Yes pushRcvr: 1 means the second field!

The image shows a screenshot of the Inspector on a CompiledMethod (Point>>#degrees) in a development environment. The window is split into two panes. The left pane shows the bytecode for the method, and the right pane shows the symbolic bytecode.

Inspector on a CompiledMethod (Point>>#degrees)

a CompiledMethod (Point>>#degrees)

Raw	Bytecode	Source	Ir	AST	Header	Meta
89	<00>	pushRcvr: 0				
90	<75>	pushConstant: 0				
91	<B6>	send: =				
92	<AC 09>	jumpFalse: 103				
94	<01>	pushRcvr: 1				
95	<75>	pushConstant: 0				
96	<B5>	send: >=				
97	<99>	jumpFalse: 100				
98	<20>	pushConstant: 90.0				
99	<90>	jumpTo: 101				
100	<21>	pushConstant: 270.0				
101	<A4 1E>	jumpTo: 133				
103	<01>	pushRcvr: 1				
104	<D2>	send: asFloat				
105	<00>	pushRcvr: 0				
106	<D2>	send: asFloat				

a SymbolicBytecode (94 <01> pushRcvr: 1)

```
origin in degrees. right is 0; down is 90."
| tan theta |
^ x = 0
ifTrue:
  [ y >= 0
    ifTrue: [ 90.0 ]
    ifFalse: [ 270.0 ] ]
ifFalse:
  [ tan := y asFloat / x asFloat.
    theta := tan arcTan.
    x >= 0
      ifTrue:
        [ y >= 0
          ifTrue: [ theta radiansToDegrees ]
          ifFalse: [ 360.0 + theta
radiansToDegrees ] ]
      ifFalse: [ 180.0 + theta
radiansToDegrees ] ]
```

Pharo Pro devs do XtremeTDD

- Get **productivity boost**
- Xtreme TDD
 - write test, test fails and
 - **code in debugger**



Counter



Counter

Filter...

All Packages Scoped View | Inst. side Class side

? Comment x

+ New class x



```
Object subclass: #NameOfSubclass
instanceVariableNames: ''
classVariableNames: ''
package: 'Counter'
```

Slots



CounterTest



Counter

CounterTest

instance side

Counter

Filter...

All Packages Scoped View | Flat Hier. | Inst. side Class side | Methods Vars |

+ New class x

? Comment x

CounterTest x

setUp x

+ Inst. side meth x

TestCase subclass: #CounterTest

instanceVariableNames: ''

classVariableNames: ''

package: 'Counter'

⚠ Test class not in a package with name ending with '-Tests' ✖ ?



CounterTest



Counter	<input type="radio"/> CounterTest	<input checked="" type="radio"/> instance side	
Counter	Filter...		

All Packages Scoped View | Flat Hier. | Inst. side Class side | Methods Vars |

? Comment x CounterTest x setUp x **+ *Inst. side met x**

```
testSetAndGetCounter
  self assert: (Counter new count: 22) count equals: 22|
```

CounterTest

Unknown variable: Counter please correct, or cancel:

- Leave variable undeclared
- Define new class
- Declare new global
- Declare new class variable
- Define new trait
- CounterIconStyler
- ExecutionCounter
- BlockLocalTempCounter
- SUnitTestsCounter
- PRNotFoundCounter
- PRCascadingCounter
- CounterTest
- ComposablePresenter
- ContainerPresenter
- CompletionSorter
- Cancel

Cancel

Class side | Methods | Vars |

de met x

```
equals: 22
```

2/2 [55] tests extension

CounterTest

Counter CounterTest instance side

Counter

All Packages S


Comment x

testSetAndGet
self assert

Vars |

2/2 [55] tests extension

Information Required

 Edit class definition:

```
Object subclass: #Counter
  instanceVariableNames: ""
  classVariableNames: ""
  category: 'Counter'
```

OK Cancel

Counter	Counter ! CounterTest	instance side tests	testSetAndGetCounter
---------	--------------------------	------------------------	----------------------

All Packages Scoped View | Flat Hier. | Inst. side Class side | Methods Vars | C

? Comment x CounterTest x setUp x testSetAndGet x + Inst. side meth x + Inst. side n

```
testSetAndGetCounter  
  self assert: (Counter new count: 22) count equals: 22
```




Instance of Counter did not understand #count:

Bytecode GT ▾

Stack

+ Create ▶ Proceed ↺ Restart ↪ Step into ↪ Step over ↪ Step through ▾

Class	Method	Other	Package
CounterTest	testSetAndGetCounter		Counter
CounterTest(TestCase)	performTest		SUnit-Core
CounterTest(TestCase)	runCase	[self setUp. self performTest	SUnit-Core
FullBlockClosure(BlockClosure)	ensure:		Kernel

Source

🔍 Where is? 📄 Browse

testSetAndGetCounter

```
self assert: (Counter new count: 22) count equals: 22
```

Variables Evaluator

Type	Variable	Value
implicit	self	CounterTest>>#testSetAndGetCounter
attribute	expectedFails	an Array [0 items] ()
attribute	testSelector	#testSetAndGetCounter
implicit	thisContext	CounterTest>>testSetAndGetCounter

Create the missing class or method in the user prompted class, and restart the debugger at the location where it can be edited.



Instance of Counter d

Bytecode GT

Stack

+ Create ▶ Proceed ↺ Restart ↻ Step into ↴ Step over ↵ Step through ⋮

Class	Method	Other	Package
CounterTest	testSetAndGetCounter		Counter
CounterTest(TestCase)	performTest		SUnit-Core
CounterTest(TestCase)	runCase	[self setUp. self performTest	SUnit-Core
FullBlockClosure(BlockClosure)	ensure:		Kernel

Source

Where is? Browse

testSetAndGetCounter

```
self assert: (Counter new count: 22) count equals: 22
```

Variables Evaluator

Type	Variable	Value
implicit	self	CounterTest>>#testSetAndGetCounter
attribute	expectedFails	an Array [0 items] ()
attribute	testSelector	#testSetAndGetCounter
implicit	thisContext	CounterTest>>testSetAndGetCounter



Instance of Counter did not understand #count:

Bytecode GT

Stack

Proceed Restart Step into Step over Step through

Class	Method	Other	Package
Counter	count:		Counter
CounterTest	testSetAndGetCounter		Counter
CounterTest(TestCase)	performTest		SUnit-Core
CounterTest(TestCase)	runCase	[self setUp. self performTest	SUnit-Core

Source

Where is? Browse

```
count: anInteger  
self shouldBeImplemented.
```

Variables Evaluator

Type	Variable	Value
implicit	self	a Counter
parameter	anInteger	22
implicit	thisContext	Counter>>count:
implicit	stack top	22



Instance of Counter did not understand #count:

Bytecode GT

Stack

Proceed Restart Step into Step over Step through

Class	Method	Other	Package
Counter	count:		Counter
CounterTest	testSetAndGetCounter		Counter
CounterTest(TestCase)	performTest		SUnit-Core
CounterTest(TestCase)	runCase	[self setUp. self performTest	SUnit-Core

Source

Where is? Browse

```
count: anInteger  
  count := anInteger
```

Variables

Type	Variable	Value
implicit	self	a Counter
parameter	anInteger	22
implicit	thisContext	Counter>>count:
implicit	stack top	22

Unknown variable: count please correct, or cancel:

- Declare new temporary variable
- Declare new instance variable
- Cancel

Cancel

Restart Step into Step over Step through

Other	Package
	Counter
	Counter
	SUnit-Core
[self setUp. self performTest SUnit-Core	

Source

Where is? Browse

```
count: anInteger
count := anInteger
```

Variables Evaluator

Type	Variable	Value
implicit	self	a Counter
parameter	anInteger	22
implicit	thisContext	Counter>>count:
implicit	stack top	22



Instance of Counter did not understand #count:

Bytecode GT

Stack

Proceed Restart Step into Step over Step through

Class	Method	Other	Package
Counter	count:		Counter
CounterTest	testSetAndGetCounter		Counter
CounterTest(TestCase)	performTest		SUnit-Core
CounterTest(TestCase)	runCase	[self setUp. self performTest	SUnit-Core

Source

Where is? Browse

```
count: anInteger  
  count := anInteger
```

Variables Evaluator

Type	Variable	Value
implicit	self	a Counter
parameter	anInteger	22
attribute	count	nil
implicit	thisContext	Counter>>count:

Relinquish debugger control and proceed execution from the current point of debugger control.cmd+r

Instance of Counter did not un... Bytecode GT

Stack Proceed Restart Step into Step over Step through

Class	Method	Other	Package
Counter	count:		Counter
CounterTest	testSetAndGetCounter		Counter
CounterTest(TestCase)	performTest		SUnit-Core
CounterTest(TestCase)	runCase	[self setUp. self performTest	SUnit-Core

Source Where is? Browse

```
count: anInteger  
  count := anInteger
```

Variables Evaluator

Type	Variable	Value
implicit	self	a Counter
parameter	anInteger	22
attribute	count	nil
implicit	thisContext	Counter>>count:



Instance of Counter did not understand #count

Bytecode GT

Stack

+ Create ▶ Proceed ↺ Restart ↵ Step into ↗ Step over ↘ Step through

Class	Method	Other	Package
CounterTest	testSetAndGetCounter		Counter
CounterTest(TestCase)	performTest		SUnit-Cor
CounterTest(TestCase)	runCase	[self setUp. self performTest	SUnit-Cor
FullBlockClosure(BlockClosure)	ensure*		Kernel

Source

Where is? Browse

testSetAndGetCounter

```
self assert: (Counter new count: 22) count equals: 22
```

Variables Evaluator

Type	Variable	Value
implicit	self	CounterTest>>#testSetAndGetCounter
attribute	expectedFails	an Array [0 items] ()
attribute	testSelector	#testSetAndGetCounter
implicit	thisContext	CounterTest>>testSetAndGetCounter



Instance of Counter did not understand #count

Bytecode GT

Stack

Proceed Restart Step into Step over Step through

Class	Method	Other	Package
Counter	count		Counter
CounterTest	testSetAndGetCounter		Counter
CounterTest(TestCase)	performTest		SUnit-Cor
CounterTest(TestCase)	runCase	[self setUp. self performTest	SUnit-Cor

Source

Where is? Browse

```
count
  ^ count
```

Variables Evaluator

Type	Variable	Value
implicit	self	a Counter
attribute	count	22
implicit	thisContext	Counter>>count
implicit	stack top	nil



CounterTest>>testSetAndGetCounter

Counter	Counter ! CounterTest	instance side tests	testSetAndGetCounter
---------	--------------------------	------------------------	----------------------

All Packages Scoped View | Flat Hier. | Inst. side Class side | Methods Vars | C

? Comment x CounterTest x setUp x testSetAndGet x + Inst. side met x + Inst. side n

testSetAndGetCounter

```
self assert: (Counter new count: 22) count equals: 22
```

1/2 [1]

**Micro commits are
your friends**

Micro commits?

- Record all your activities
- Support the replay, query and reload of actions

Going back in time

Recent versions of Counter>>#increment

Browse References Senders Implementors Revert

Side By Side

Location	Selector	Package	Timestamp
Counter (accessing)	increment	[]	1901-01-01T00:00:00+00:00
Counter (accessing)	increment	[]	1901-01-01T00:00:00+00:00
Counter (accessing)	increment	[]	1901-01-01T00:00:00+00:00
Counter (accessing)	increment	[]	1901-01-01T00:00:00+00:00
Counter (accessing)	increment	[]	1901-01-01T00:00:00+00:00

Pretty print

increment

```
self halt.  
count := count + 2
```

increment

```
count := count + 1
```

Navigating/Querying past activities

The screenshot displays the Epicea - Code Changes interface. On the left, a table lists file modifications:

	Modified	File
○	*	Pharo12.0-S...D4497C1B8C8
○	Yesterday	Pharo12.0-S...847E1D69550
○	2 days ago	Pharo12.0-S...447640D1228
○	3 days ago	Pharo12.0-S...5FC8697FD30

The main area shows a filtered list of code changes. A search filter is applied, resulting in 0 found items out of 57 total events. The filtered list includes:

- <> Counter » increment
- <> Counter » increment
- <> Counter » decrement
- <> Counter » decrement
- + Counter protocol API
- + Counter » decrement
- + Counter protocol as yet unclassified
- + CounterTest » testDecrement
- <> Counter » increment
- <> Counter » increment
- + Counter » increment
- + Counter protocol as yet unclassified
- + CounterTest » testIncrement
- <> Counter » count
- + Counter » count
- + Counter protocol as yet unclassified

At the bottom, two side-by-side code snippets are shown, illustrating a change in the increment operation:

```
"protocol: #accessing"
increment
count := count + 1
```

```
"protocol: #accessing"
increment
count := count + 2
```

Selective Replay

The screenshot shows the Epicea - Apply Changes window. At the top, there is a search bar labeled "Filter..." and a "Find" button. Below this is a list of events with the following items:

- <> Counter » increment
- <> Counter » decrement (highlighted)
- + CounterTest » testDecrement
- + CounterTest » testIncrement
- <> Counter » count
- <> Counter » count:
- + Counter
- <> CounterTest » testSetAndGet
- + CounterTest
- + Tag Uncategorized in package Counter
- + Package Counter

At the bottom of the list, it says "11 events | 0 found | 11 filtered". To the right of this is a button labeled "Confirm All Changes".

Below the list, there are two side-by-side code snippets for comparison:

```
"protocol: #API"  
decrement  
self shouldBeImplemented.
```

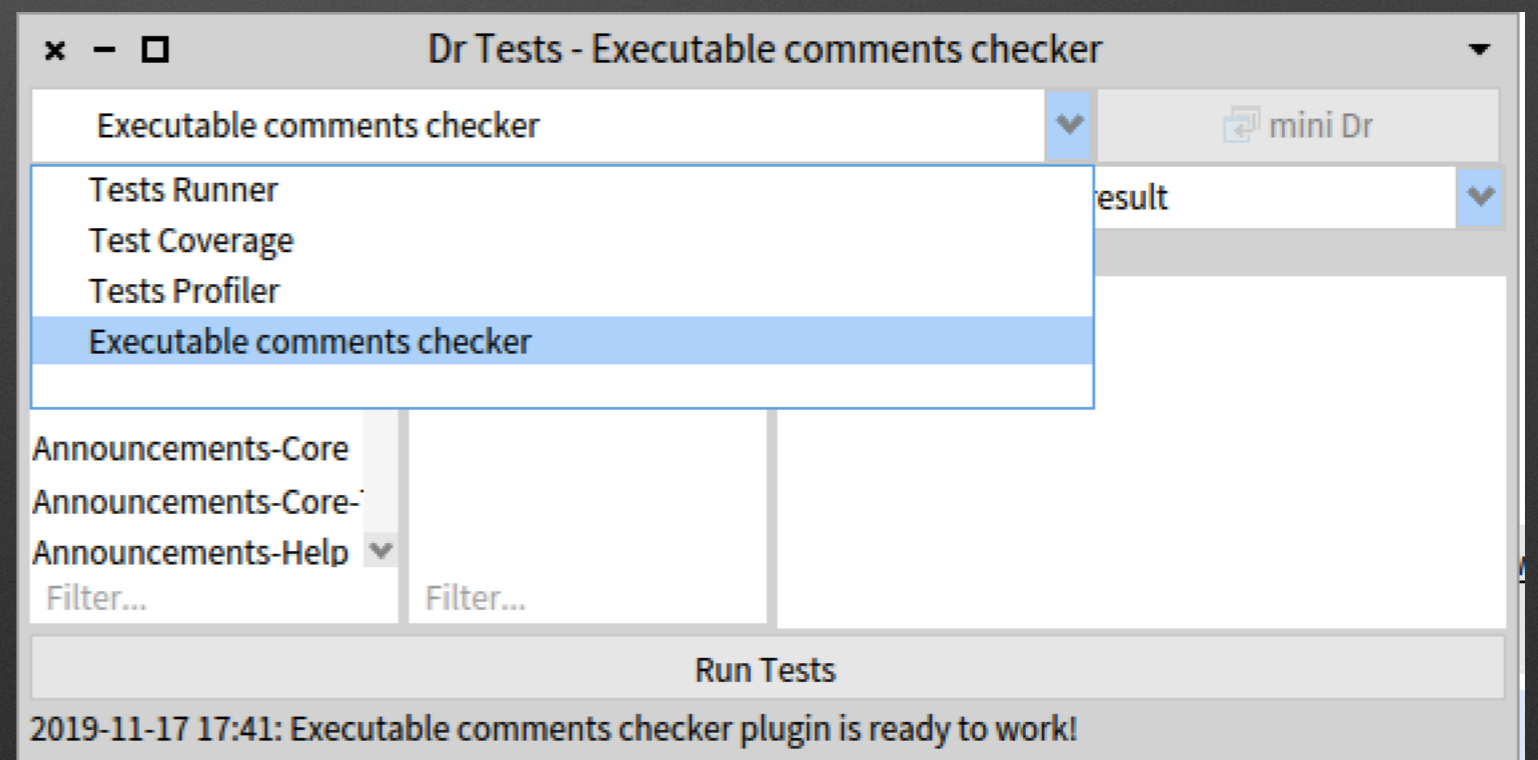
```
"protocol: #API"  
decrement  
count := count - 1
```

What about tests?

Tests are also our best friends

- DrTests
 - Plugin for Rotten Green tests
 - Executable comments
- Mutalk: Mutation analyser framework

DrTests: a plugin-based architecture to plug test analyses



**We validated our 27000
tests for Rotten Green
Tests (ICSE'19)**

Hot update on the fly
customizable debugger

Halt

Bytecode

Stack

Proceed Restart Into Over Through

PDFCellElement	getSubElementsWith:styleSheet:
PDFCellElement(PDFComposite)	generateCodeSegmentsCollectionWi
PDFCellElement(PDFComposite)	generateCodeSegmentWith:styleShe
PDFDataTableElement(PDFComposite)	generateCodeSegmentsCollectionWi [:aSubElement aSubElement generateCodeSe
Array(SequenceableCollection)	collect:

Source

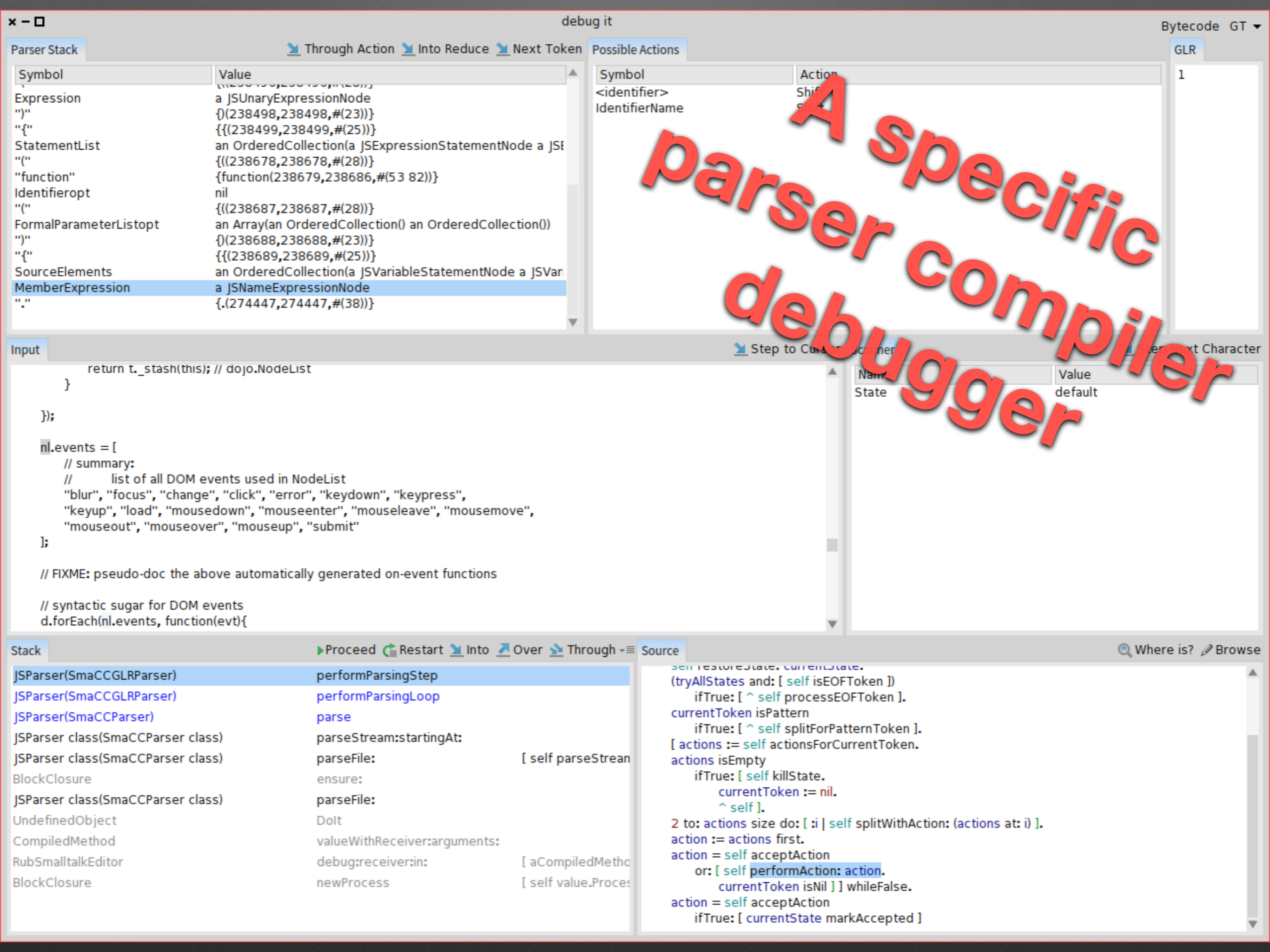
Where is? Browse

```
generateCodeSegmentsCollectionWith: aPDFGenerator styleSheet: compositeStyleSheet format: aFormat
^ (self getSubElementsWith: aPDFGenerator styleSheet: compositeStyleSheet)
  collect: [ :aSubElement |
    aSubElement
      generateCodeSegmentWith: aPDFGenerator
      styleSheet: (aSubElement buildCompositeStyleSheetFrom: compositeStyleSheet)
      format: aFormat ]
```

Variables

Type	Variable	Value
implicit	self	a PDFCellElement
parameter	aFormat	a PDFA4Format
parameter	aPDFGenerator	a PDFGenerator
parameter	compositeStyleSheet	a StyleSheet

dimension: 80 mm @ 20 mm;



A specific
parser compiler
debugger

Symbol	Value
Expression	a JSUnaryExpressionNode
"{"	{(238498,238498,#(23))}
"{"	{{(238499,238499,#(25))}
StatementList	an OrderedCollection(a JSExpressionStatementNode a JS
"{"	{{(238678,238678,#(28))}
"function"	{function(238679,238686,#(53 82))}
Identifieropt	nil
"{"	{{(238687,238687,#(28))}
FormalParameterListopt	an Array(an OrderedCollection() an OrderedCollection())
"{"	{(238688,238688,#(23))}
"{"	{{(238689,238689,#(25))}
SourceElements	an OrderedCollection(a JSVariableStatementNode a JSVari
MemberExpression	a JSNameExpressionNode
"."	.(274447,274447,#(38))

Symbol	Action
<identifier>	Shift
IdentifierName	Shift

```
return t._stash(this); // dojo.NodeList
}
});
nl.events = [
  // summary:
  // list of all DOM events used in NodeList
  "blur", "focus", "change", "click", "error", "keydown", "keypress",
  "keyup", "load", "mousedown", "mouseenter", "mouseleave", "mousemove",
  "mouseout", "mouseover", "mouseup", "submit"
];
// FIXME: pseudo-doc the above automatically generated on-event functions
// syntactic sugar for DOM events
d.forEach(nl.events, function(evt){
```

Name	Value
State	default

Stack	Source
JSParser(SmaCCGLRParser)	performParsingStep
JSParser(SmaCCGLRParser)	performParsingLoop
JSParser(SmaCCParser)	parse
JSParser class(SmaCCParser class)	parseStream:startingAt:
JSParser class(SmaCCParser class)	parseFile: [self parseStream
BlockClosure	ensure:
JSParser class(SmaCCParser class)	parseFile:
UndefinedObject	Dolt
CompiledMethod	valueWithReceiver:arguments:
RubSmalltalkEditor	debug:receiver:in: [aCompiledMetho
BlockClosure	newProcess [self value.Proces

```
self restoreState: currentState.
(tryAllStates and: [ self isEOFToken ])
  ifTrue: [ ^ self processEOFToken ].
currentToken isPattern
  ifTrue: [ ^ self splitForPatternToken ].
[ actions := self actionsForCurrentToken.
actions isEmpty
  ifTrue: [ self killState.
            currentToken := nil.
            ^ self ].
2 to: actions size do: [ :i | self splitWithAction: (actions at: i) ].
action := actions first.
action = self acceptAction
  or: [ self performAction: action.
        currentToken isNil ] ] whileFalse.
action = self acceptAction
  ifTrue: [ currentState markAccepted ]
```

Halt in OCDBox>>name:

Type	Target	Method
<input type="checkbox"/>	Breakpoint self	Reflecting
<input checked="" type="checkbox"/>	Breakpoint self	OCDBox>>initialize
<input checked="" type="checkbox"/>	Breakpoint self	OCDBox>>name:
<input checked="" type="checkbox"/>	Breakpoint self	OCDBox>>name:
<input checked="" type="checkbox"/>	Halt OCDBox	OCDBox>>name:
<input type="checkbox"/>	Breakpoint OCDBox	OCDBox>>name:
<input type="checkbox"/>	Breakpoint self	OCDBox>>removeElement:
<input type="checkbox"/>	Breakpoint self	OrderedCollection>>remove
<input type="checkbox"/>	Halt StHaltCacheTest	StHaltCacheTest>>testInitial
<input checked="" type="checkbox"/>	Halt StHaltCacheTest	StHaltCacheTest>>testInitial

```

1 name: anObject
2 self halt.
3 name := anObject
  
```

Specific view

Receiver in: a StDebuggerContext (OCDBox>>name:)

Variable	Value
[arg] anObject	'i'
self	an OCDBox
elements	an OrderedCollection [0 items] ()
name	''
Temps	a Dictionary [1 item] (#anObject->'i')
anObject	'i'
stackTop	an OCDBox

Type	Target	Method
<input checked="" type="checkbox"/>	Halt OCDBox	OCDBox>>name:

```

1 "an OCDBox"
2 self
  
```

Bytecode Breakpoints Sindarin

```

33 <4C> self
34 <80> send: halt
35 <D8> pop
36 <40> pushTemp: 0
37 <C9> popIntoRcvr: 1
38 <58> returnSelf
  
```

Variable	Value
stackTop	an OCDBox
0 [anObject]	'i'
rcvr: 0 [elements]	an OrderedCollection [0 items] ()
rcvr: 1 [name]	''

object centric +

adv scripting (AST + call stack)

Receiver in: a StDebuggerContext (OCDBox>>name:)

Variable	Value
[arg] anObject	'i'

Type	Target	Method
<input checked="" type="checkbox"/>	Halt OCDBox	OCDBox>>name:

Variable	Value
stackTop	an OCDBox
0 [anObject]	'i'
rcvr: 0 [elements]	an OrderedCollection [0 items] ()
rcvr: 1 [name]	''

Debug actions

- Since Pharo 12
- Toggle set of stop points:
 - toggle off + run the tests + toggle on and code
- Chain stop points: only stop here if you stopped there



Debug Point Browser

(De)activate all

Search by name



Type	Target	Name	Scope
<input checked="" type="checkbox"/> Breakpoint	RBInlineMethodRefactoring>>#pr	Breakpoint	class RBI

- enabled: (de)activates debug point
- Condition: Hit when the condition evaluates to true

1	

- Test Environment Only: Hits only when executing tests
- Chain: Each debug point is hit once in sequential order
- Counter: Tracks how many times the debug point was reached
- Once: Deactivates debug point after one hit
- Script: Executes a script at each hit

1	

- Transcript: Logs to transcript at each hit
- Break

```

1 privateTransform
2   self
3   renameConflictingTemporaries;
4   insertInlinedMethod;
5   compileMethod

```

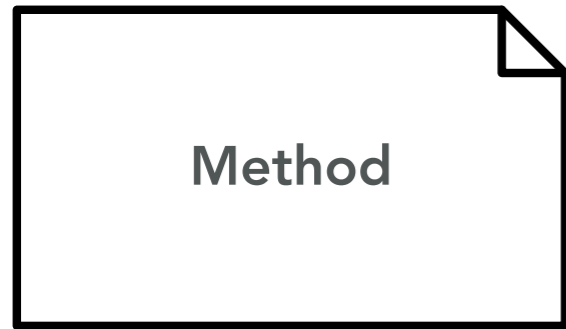
**Dynamically rewriting
deprecated calls @
runtime**

How to support migration to new versions

We deprecate API

How to help our users to migrate?

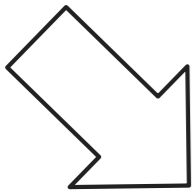
Deprewriter



Annotation (1)

Declaration

Execution

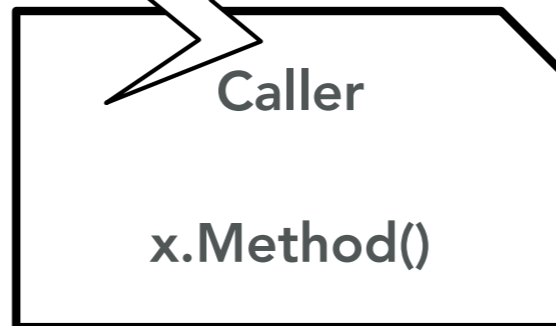


Exception handling (3)

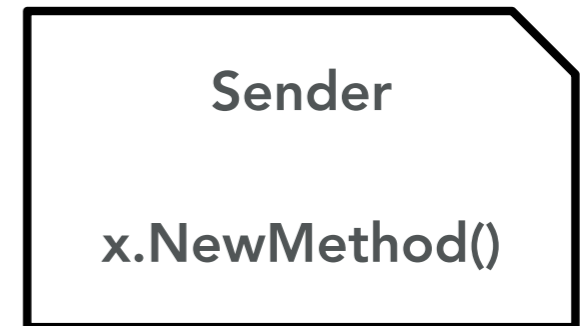
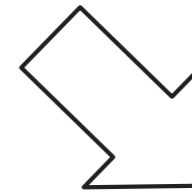
Warning

Caller rewriting (4)

Execution continues (5)



Execution (2)



Rewriting deprecation

`crLog: aString`

`self`

`deprecated: 'Please use trace* methods instead.'`

`transformWith:`

`'@receiver crLog: `@statements1'`

`-> '@receiver crTrace: `@statements1'.`

`self crTrace: aString`

**Run your tests.
Now your code and
tests use the new
API!**

At notification time:
Walk the stack
Get caller AST
If should be rewritten
Rewrite it and proceed
execution

transform

| node rewriteRule aMethod |

aMethod := self contextOfSender method.

node := self contextOfSender sourceNodeExecuted.

rewriteRule := self rewriterClass new

replace: rule key with: rule value.

(rewriteRule executeTree: node)

ifFalse: [^ self signal].

node replaceWith: rewriteRule tree.

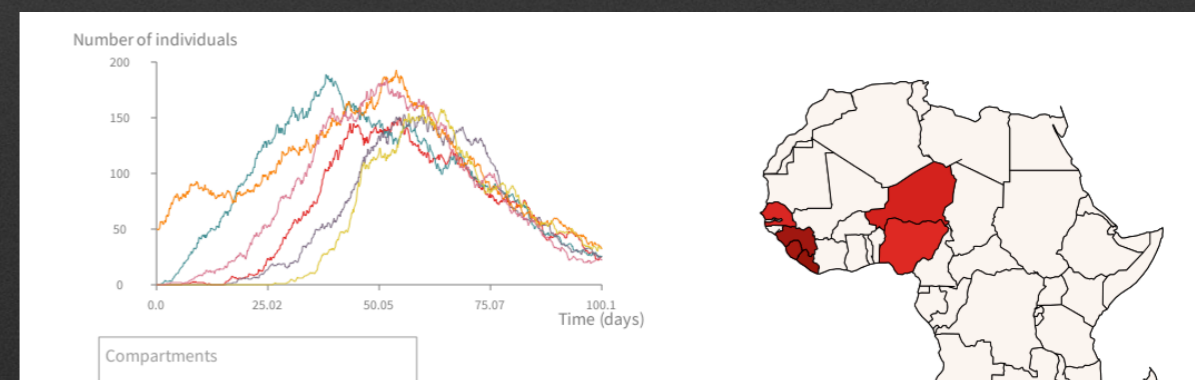
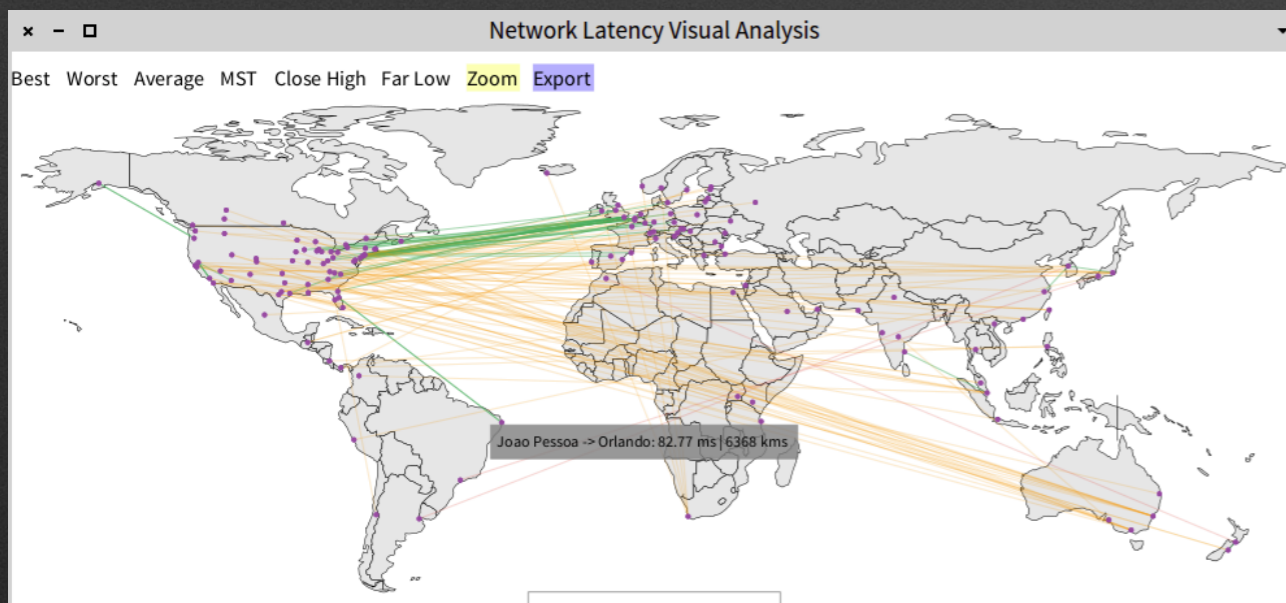
Live visualisation scripting

- Kind of another level
- Roassal 3.0 by Prof. A. Bergel
- Simply gorgeous
- Check <http://agilevisualization.com>

Includes a DSL for Scripting visualisations

```
b := RTMondrian new.  
  b shape rectangle  
    withBorder;  
  width: [ :cls | cls numberOfVariables * 5 ];  
  height: [ :cls | cls numberOfMethods ].
```

```
b nodes: Collection withAllSubclasses.  
b edges connectToAll: [ :cls | cls subclasses ].  
b layout tree.  
b normalizer  
  normalizeColorAsGray: [ :cls |  
cls numberOfLinesOfCode ].  
b
```



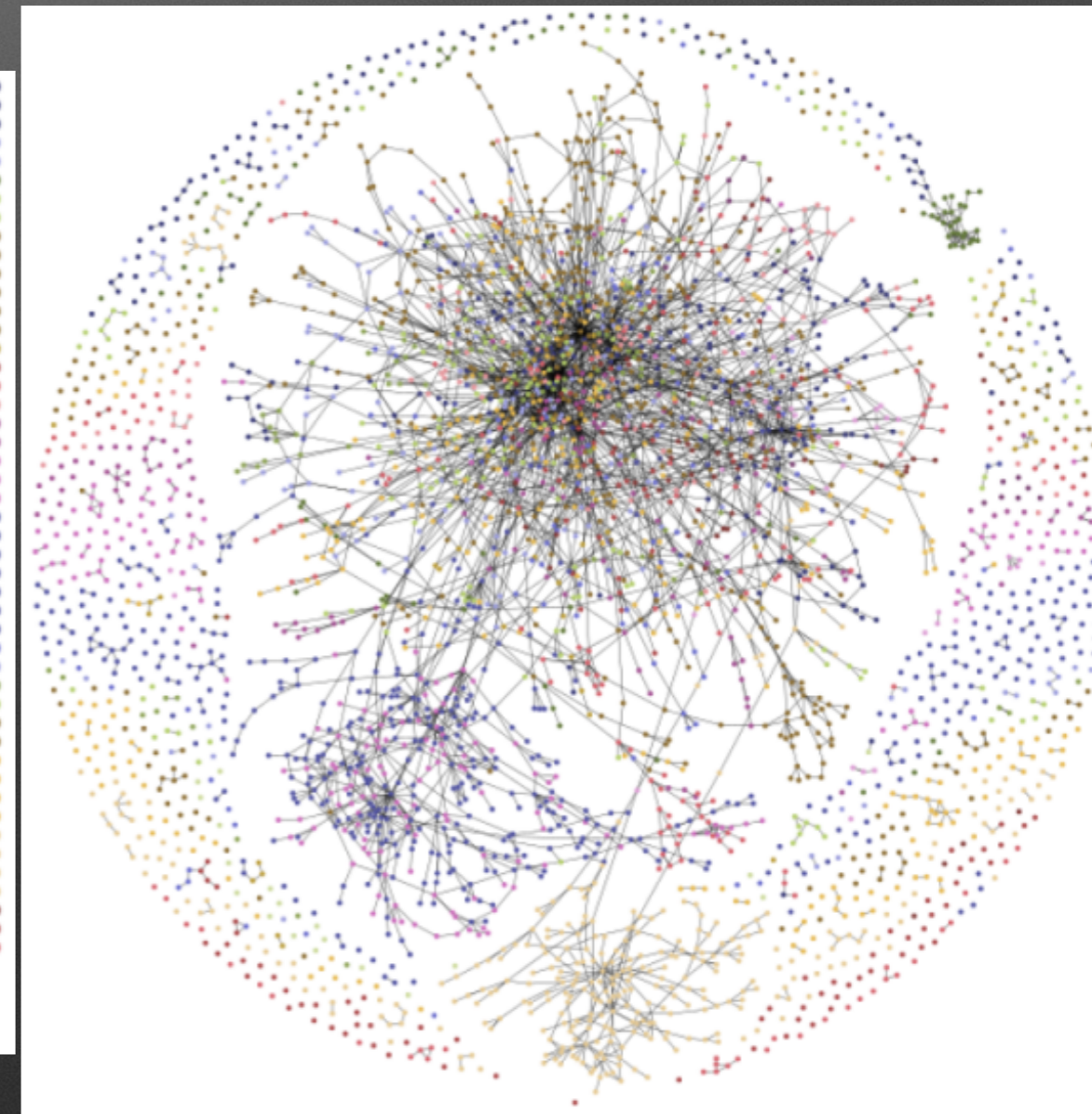
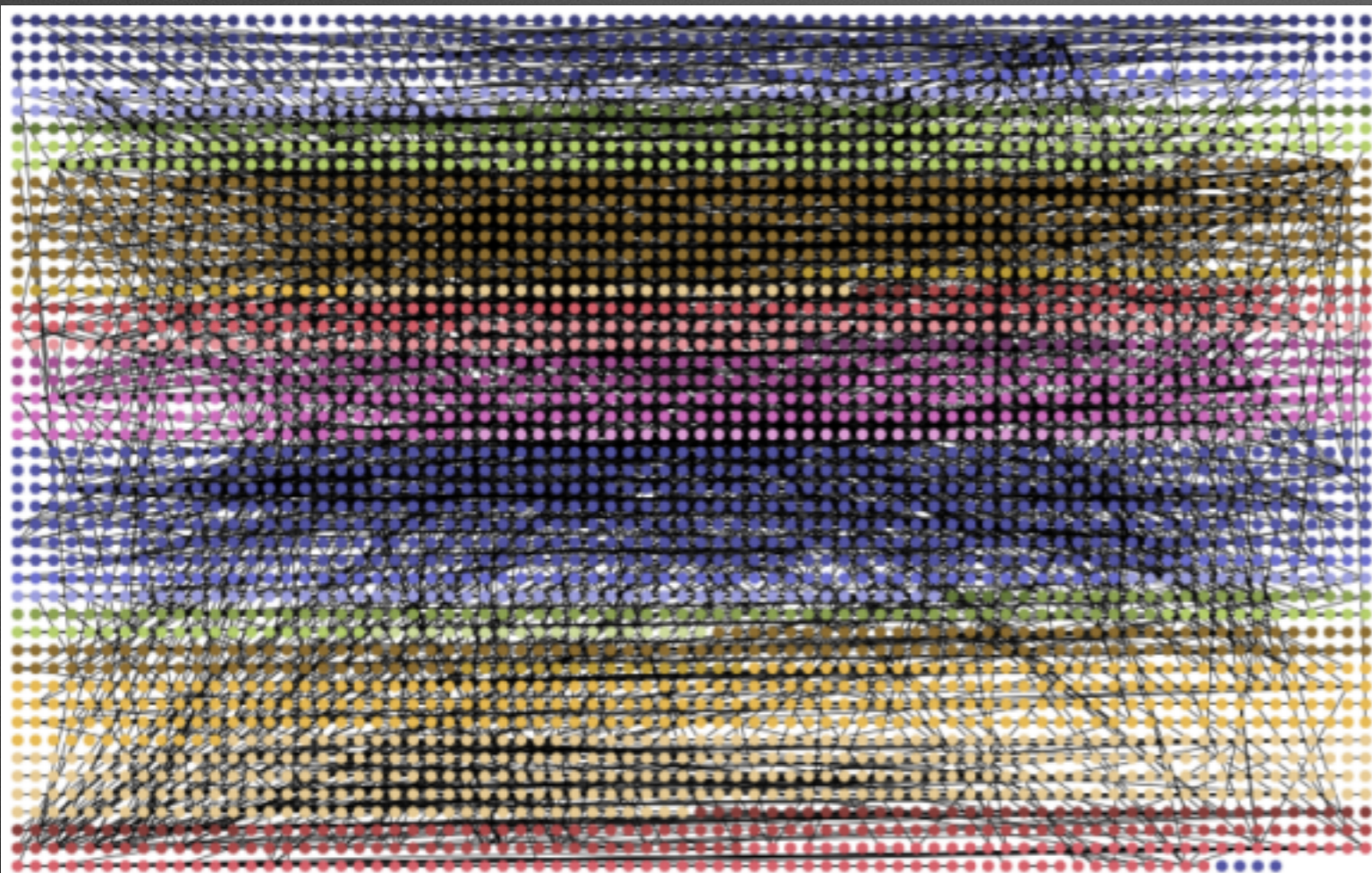
One hour about Basic

COPYRIGHT 1975 BY BILL GATES AND PAUL ALLEN

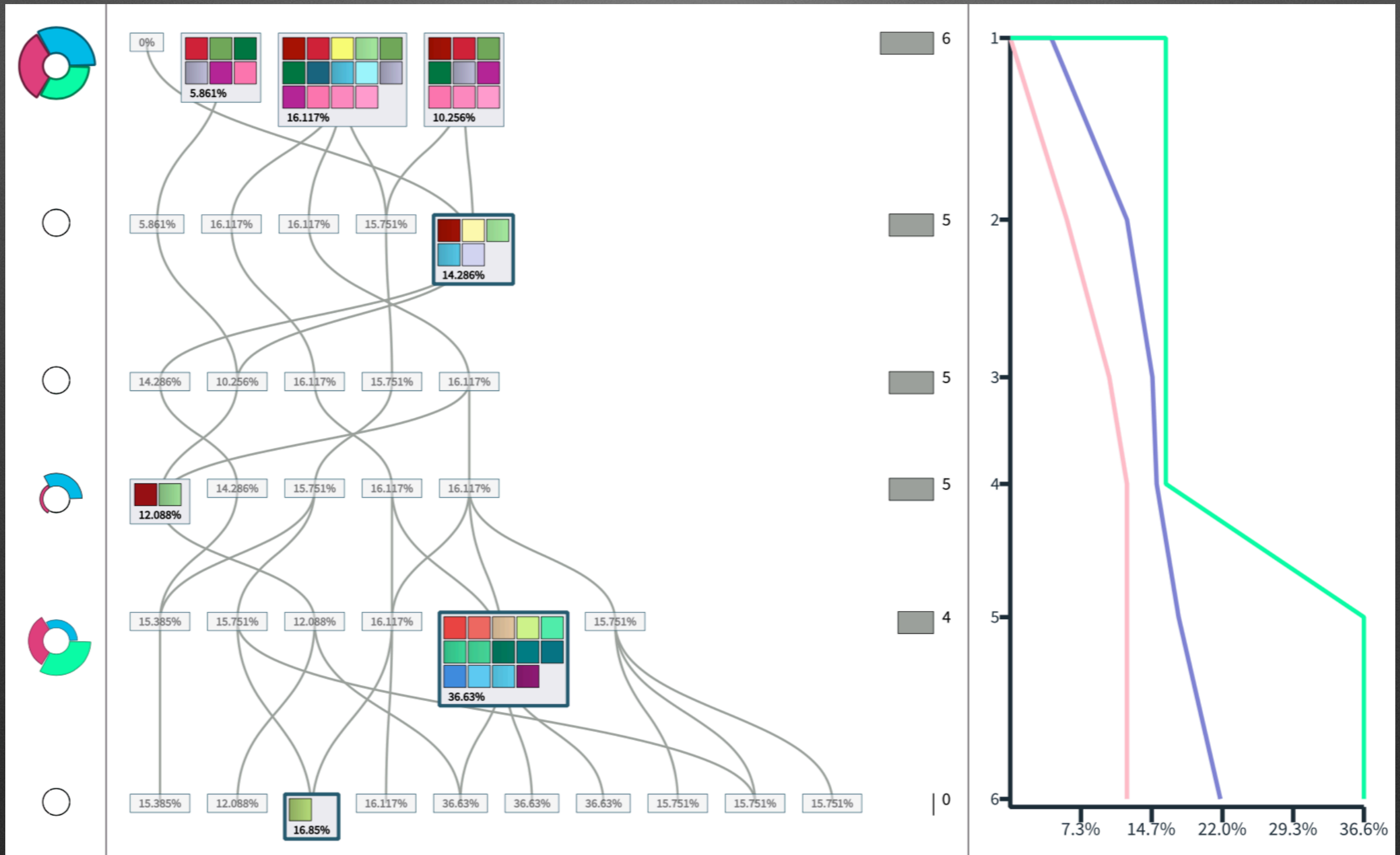
ORIGINALLY WRITTEN ON THE PDP-10 FROM
FEBRUARY 9 TO APRIL 9 1975

BILL GATES WROTE A LOT OF STUFF.
PAUL ALLEN WROTE A LOT OF OTHER STUFF AND FAST CODE.
MONTE DAVIDOFF WROTE THE MATH PACKAGE (F4I.MAC).

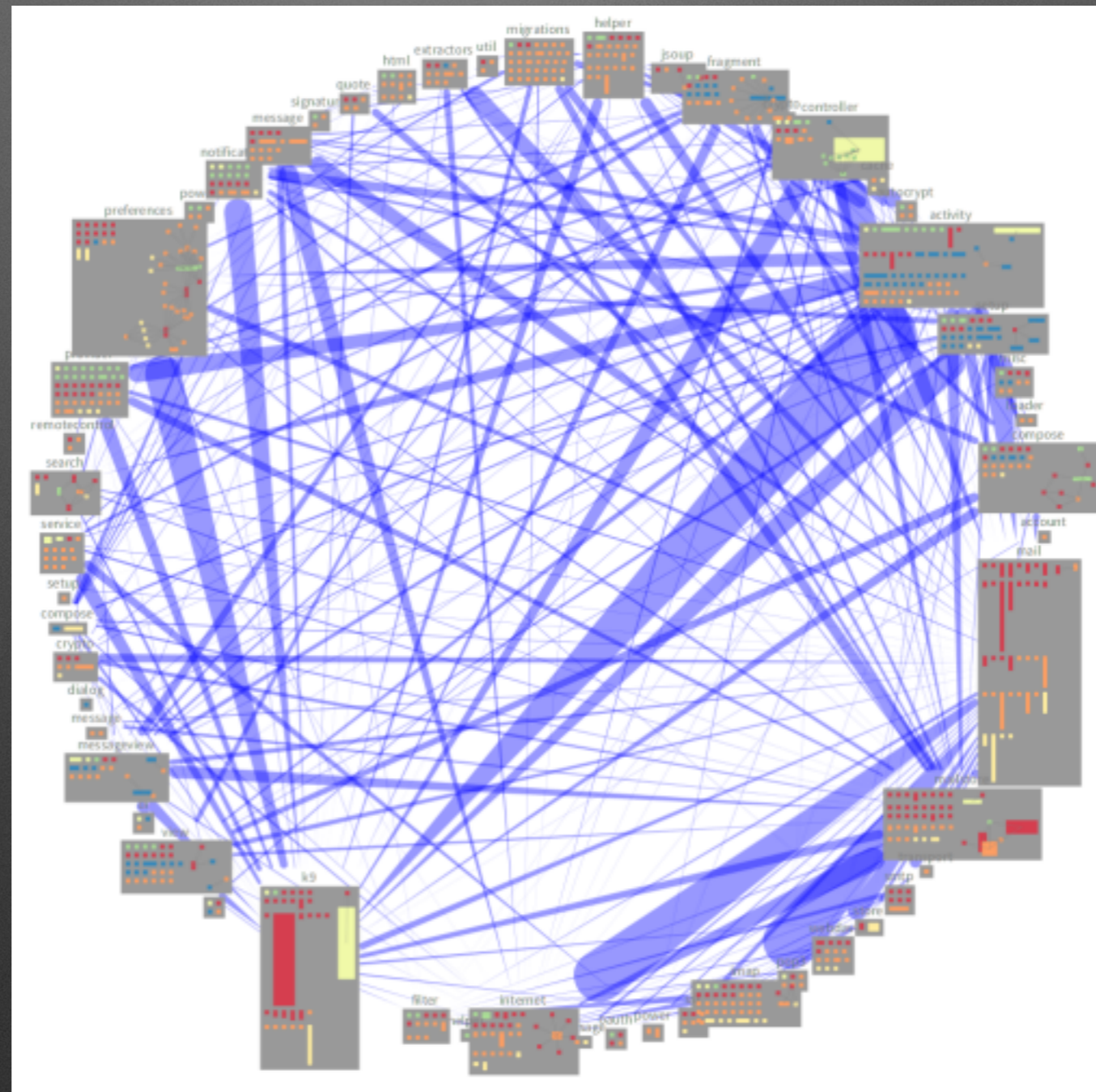
<https://pharoweekly.wordpress.com/2020/05/24/roassal-1-hour-xp-assembly-code-of-gwbasic/>



Execution of IA generating tests



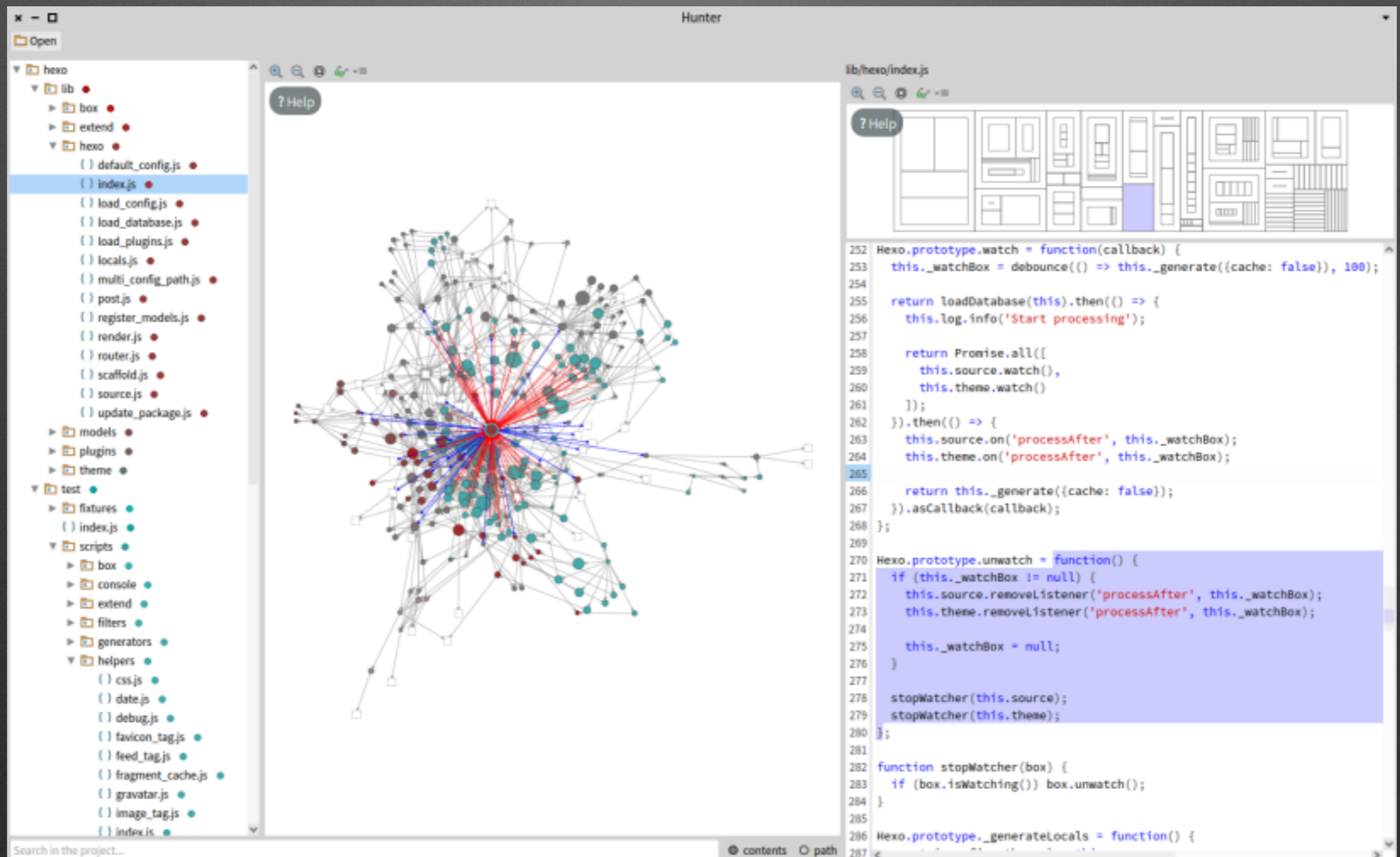
Analysis Android application



**Often developers
write their own tools**

Building your own tool

- Example Javascript analysis



The screenshot displays the Hunter tool interface, which is used for JavaScript analysis. The interface is divided into three main sections:

- File Explorer (Left):** Shows a hierarchical view of the project files. The 'hexo' directory is expanded, revealing sub-directories like 'lib', 'test', and 'helpers', along with numerous JavaScript files such as 'index.js', 'load_config.js', and 'update_package.js'.
- Network Graph (Center):** A complex network graph representing the relationships between the analyzed JavaScript files. Nodes are represented by colored circles (red, blue, green, grey) and are interconnected by lines (edges). A central node is highlighted with a red circle, and several edges radiating from it are colored red, indicating a high degree of connectivity or a central role in the codebase.
- Code Editor (Right):** Displays the source code of the selected file, 'lib/hexo/index.js'. The code is shown in a dark theme with syntax highlighting. The function `Hexo.prototype.watch` is highlighted in blue, and the function `Hexo.prototype.unwatch` is highlighted in purple. The code includes logic for watching and unwatching files, using `debounce`, `Promise.all`, and `stopWatcher`.

Probabilistic Data Structure

- <https://github.com/osoco/PharoPDS>
- Defined new data structure
- And the **analysis tools**

The screenshot displays the 'PDS Algorithms Viewer' interface. The main window is titled 'PDS Algorithms Viewer' and contains several tabs: '+ New Bloom Filter', 'Analysis', 'Profiling', 'Benchmarking', and 'Reset'. The 'Analysis' tab is active, showing a 'Bloom Filter' configuration.

Add elements
Write the text you want to add to the filter.
e.g. Madrid
london added to filter!

Membership test
Check if a given text is a member of the filter.
E.g. London

Bloom Filter BitSet
In Out Center

a PDSBloomFilter
Param... FPP Ci... Storag... Storag... Raw _GT Meta

Name	Value
'Target Elements (n)'	100
'Target FPP'	0.03
'Number of hashes (k)'	6
'Current Elements'	0
'Current FPP'	0.0

False-Positive Probability Curve
In Out Center

HTTP traffic analysis

- <http://youtu.be/rIBbeMdFCys>

The screenshot shows the Pharo IDE interface. At the top left is the Pharo logo. In the center, a 'Monticello Browser' window displays a repository of packages from 'http://smalltalkhub.com/mc/SvenVanCaekenberghe/'. The packages listed include 'ConfigurationOfHP35', 'HP35-Calculator', and 'HP35-Seaside-Calculator'. Below this, a 'Playground' window shows a table of network events. The selected event is a 'ZnResponseReadEvent' with the following details:

Variable	Value
self	2014-10-15 15:27:46 028 Response Read a ZnResponse(200 OK text/plain 1197B)
clientid	nil
duration	0
id	28
response	a ZnResponse(200 OK text/plain 1197B)
timestamp	2014-10-15T15:27:46.873225+02:00

Below the table, a log entry reads: "2014-10-15 15:27:46 028 Response Read a ZnResponse(200 OK text/plain 1197B) 0ms" self.

AOT compiler CFG

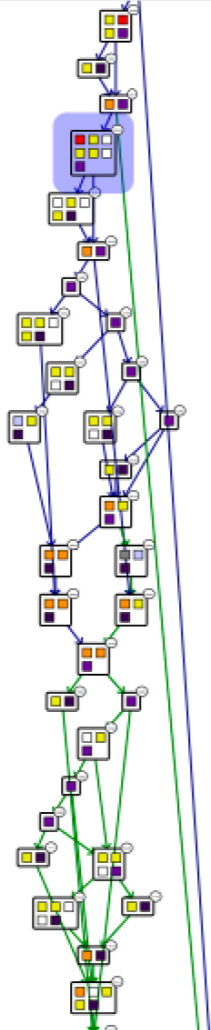
Inspector on a DRPrimitiveControlFlowGraph

a DRPrimitiveControlFlowGraph

a DRBasicBlock (DRBasicBloc...

CFG Raw Breakpoints Meta

Instructions Source code Raw Breakpoints Meta



```
R4333 := Arg0Reg
R351 := RightShift R4333 3
R370 := LOAD 8 [ReceiverResultReg]
R420 := RightShift R370 24
R446 := BitAnd R420 31
R508 := LOAD 1 [ReceiverResultReg, offset=7]
Jump if R508 EqualsThan 255 -> 919 if false -> 918
```

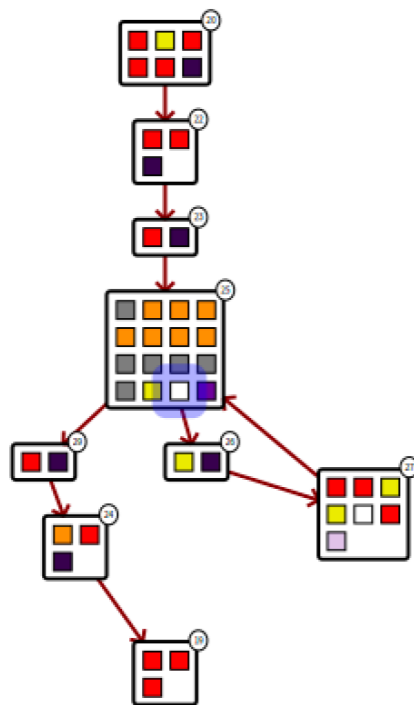
Inspector on a DRControlFlowGraph

a DRControlFlowGraph

a DRGetConditionCode(R29 :=...

CFG Raw Breakpoints Meta

Source Dependency Graph Raw Breakpoints Meta



9 blocks

44 instructions

0 paths

```
1 to: stop do: aBlock
2 "Normally compiled in-line, and therefore not overridable.
3 Evaluate aBlock for each element of the interval (self to: stop by: 1)."
4 | nextValue |
5 nextValue := self.
6 [nextValue <= stop]
7   whileTrue:
8     [aBlock value: nextValue.
9     nextValue := nextValue + 1]
```

A little binary stepper

VM Debugger

Address	ASM	Bytes						
16r1000C0	tst x23, #0x7	#['16rFF' '16rA'	lr		'16r143C000'	SP	16r143DFF8	16r123B000
16r1000C4	b.ne #760	#['16rC1' '16r1	pc		'16r10000C0'	FP	16r143E000	16r0
16r1000C8	mov x1, #1	#['16r21' '16r0'	sp		'16r143BFC0'		16r143E008	16r0
16r1000CC	mov x22, x3	#['16rF6' '16r3'	fp		'16r143E000'		16r143E010	16r0
16r1000D0	ands x1, x1, x22	#['16r21' '16r0'	x28	vmStackPointer	'16r143DFF8'		16r143E018	16r0
16r1000D4	cmp x1, #0	#['16r3F' '16r0'	x0		'16r0'		16r143E020	16r0
16r1000D8	b.eq #12	#['16r60' '16r0'	x1		'16r7FFFFFFFFFFFF'		16r143E028	16r0
16r1000DC	mov x22, #0	#['16r16' '16r0'	x2		'16r0'		16r143E030	16r0
16r1000E0	b.al #8	#['16r4E' '16r0'	x3		'16r9'		16r143E038	16r0
16r1000E4	mov x22, #1	#['16r36' '16r0'	x4		'16r0'		16r143E040	16r0
16r1000E8	cmp x1, #0	#['16r3F' '16r0'	x5		'16r0'		16r143E048	16r0
16r1000EC	b.eq #12	#['16r60' '16r0'	x6		'16r0'		16r143E050	16r0
16r1000F0	mov x22, #0	#['16r16' '16r0'	x7		'16r0'		16r143E058	16r0
16r1000F4	b.al #4	#['16r2E' '16r0'	x8		'16r0'		16r143E060	16r0
16r1000F8	cmp x22, #0	#['16rDF' '16r2	x9		'16r0'		16r143E068	16r0
16r1000FC	b.ne #704	#['16r1' '16r16'	x10		'16r0'		16r143E070	16r0
16r1000100	mov x22, x3	#['16rF6' '16r3'	x11		'16r0'		16r143E078	16r0
16r1000104	asr x22, x22, #3	#['16rD6' '16rF	x12		'16r0'		16r143E080	16r0
16r1000108	ldr x1, [x23]	#['16rE1' '16r2'	x16		'16r143BFF8'		16r143E088	16r0
16r100010C	mov x25, x1	#['16rF9' '16r3'	x19		'16r0'		16r143E090	16r0
16r1000110	asr x25, x25, #24	#['16r39' '16rF	x20		'16r0'		16r143E098	16r0
16r1000114	ands x25, x25, #	#['16r39' '16r1:	x21		'16r0'		16r143E0A0	16r0
16r1000118	ldurb w19, [x23,	#['16rF3' '16r7:	x22	classRegister	'16r0'		16r143E0A8	16r0
16r100011C	ands x19, x19, #	#['16r73' '16r1	x23	receiverRegister	'16r10B0B60'		16r143E0B0	16r0

Jump to Step Disassemble at PC

Moments of grace...

**I want my halt to only
stop when called from
THAT test called testMe!**

mycode

self haltlf: #testMe

...

Use stack reification

Walk it

Halt if needed

(in 5 lines)

```
haltIf: aSelector  
| cntxt |  
cntxt := thisContext.  
[ cntxt isNil ] whileFalse: [  
    cntxt selector = aSelector  
    ifTrue: [ self halt ].  
    cntxt := cntxt sender ]
```


**Pharo is
research friendly**

International Research Groups

Lafhis (AR)

SCG (CH)

CAR (FR)

RMOD (FR)

Ummisco (IRD)

Reveal (CH)

Lysic (FR)

ENSTA-Bretagne (FR)

CEA-List (FR)

Ryerson (CAN)

OC (FR)

CCMI-FIT (CZ)

ASERG (BR)

Pleiad (CL)

Macau (UNO)

Cirad (FR)

USTH (Vietnam)

Soft-Qual (Serbia)

Uni. Quilmes (AR)

ENIT (FR)

CS (Bo)

Maroua (CAM)

ETS (CAN)

**We can be
your guinea
pigs,...**

(well kind of.... we have real users)

Was actively supported research e.g., SCG from Uni. Bern

The screenshot shows the Pharo IDE's Spotter search interface. At the top, there is a search bar with the text "Search" and icons for settings, help, and close. Below the search bar, there is a list of search results. The first result is "#Menu 15/62" with a blue arrow icon. The second result is "System Browser" with a green checkmark and a blue arrow icon. Other results include "Playground", "Test Runner", "Spotter", "Iceberg", and "Monticello Browser". At the bottom of the interface, there is a pink notification box with the text: "No diagnostic and usage data is being sent. Would you like to send diagnostic and usage data to help us improve Pharo?" and a "Go to settings" button.

The screenshot shows the Pharo IDE's class browser for the "NumberParser" class. The browser is divided into two main sections: "Scoped" and "Variables". The "Scoped" section shows a tree view of the class hierarchy, with "AST-Core" selected. The "Variables" section shows a list of variables, including "ASTCache", "ASTCacheReset", "ManifestASTCore", "NumberParser", "RBComment", and "RBParseTreeRule". The "NumberParser" class is highlighted in blue. To the right of the class browser is a "History Navigator" panel with a list of methods: "-- all --", "accessing", "error", "initialize-release", "parsing-large int", "parsing-private", and "parsing-public". Below the class browser, there is a text area showing the class's metadata: "Object subclass: #NumberParser", "instanceVariableNames: 'sourceStream base neg integerPart fraction nDigits lastNonZero requestor failBlock'", "classVariableNames: ''", and "package: 'AST-Core-Parser'". At the bottom of the interface, there is a pink notification box with the text: "No diagnostic and usage data is being sent. Would you like to send diagnostic and usage data to help us improve Pharo?"

The screenshot shows the Pharo IDE's "Utility methods" panel. It features a title bar with "1/9 [1]" and "Format as you read" options. The main area contains a warning icon and the text "Utility methods" followed by help and close icons. At the bottom right, there is a "Helpful?" button with thumbs up and thumbs down icons.

Best Paper Award @ ICPC'19

**Kubelka, Bergel, Robbes,
“Live Programming and Software Evolution: Questions
during a Programming Change Task”**

Map reduce debugging

by M. Marra and Prof. E. Gonzales Boix from Vrije
Universiteit Brussel

Empowering is the right word

The immersive programming experience

Pharo is a pure object-oriented programming language *and* a powerful environment, focused on simplicity and immediate feedback (think IDE and OS rolled into one).

- Pharo is an energizing and creative environment
- Moldable tools are powerful
- Tried to share my feeling
- But “The idea of experience does not replace experience.” Alain

Discover

Learn more about Pharo's key features and elegant design

Download

Download latest version (8.0)!
Read more about [here](#)

Learn

Access the Pharo Moot!
3000 people registered and follow the Pharo Moot. You can find it [here](#).

Fun with Us

- 1 hour from Paris, 1:30 from London, 35 min from Brussels
- Internships 3 to 6 months (right now we have 8 interns)
- Google Summer of Code
- PhDs / co supervision (e.g., B. Sarenac)
- Engineer position
- Visitors (Magagascar, Chile, Montreal, ...)