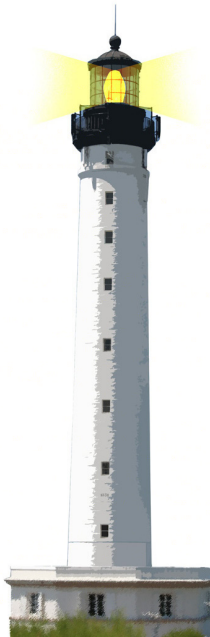
Learning Object-Oriented
Programming and Design with TDD

Learning how to discover information

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



Pharo
http://www.pharo.org

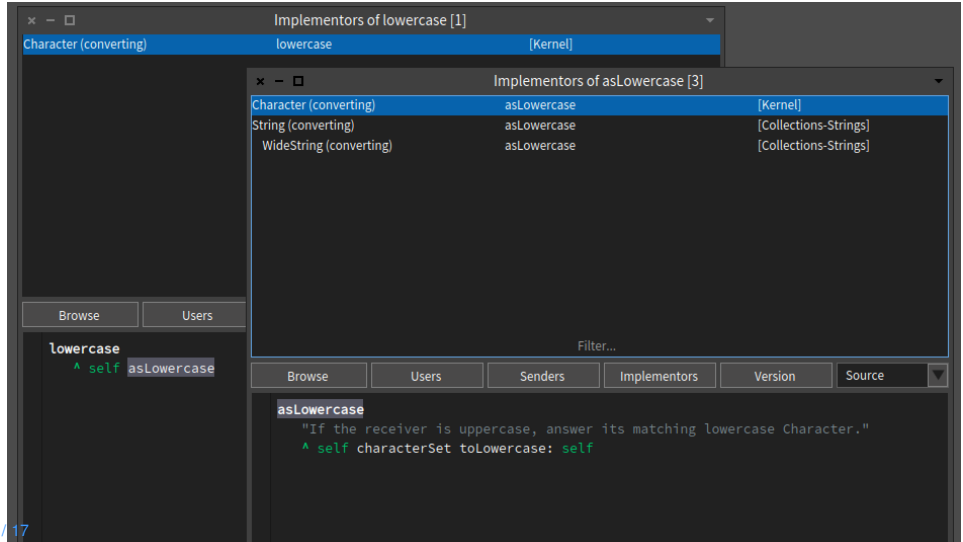
W3S06

Looking for implementors

- Look for possible implementations
- You will get methods and may be one that you can call on (sub)-instances of the class implementing it.

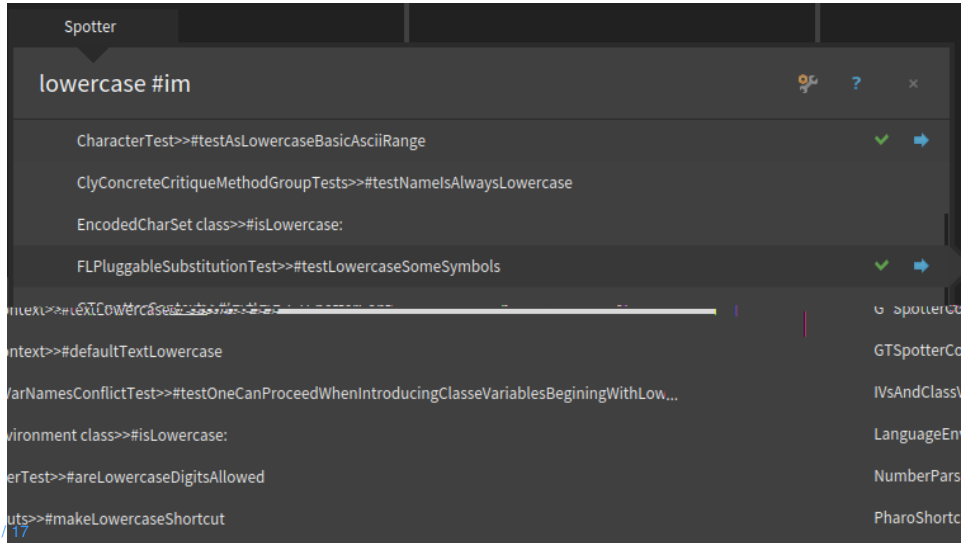
Looking for implementors

- Command-m (for iMpleMentors)



Looking for implementors (Spotter)

- Shift-Enter hex #im (all methods whose selectors contains ...)



Looking for senders

- Look how a given message is used
- Always interesting to understand typical arguments to be passed

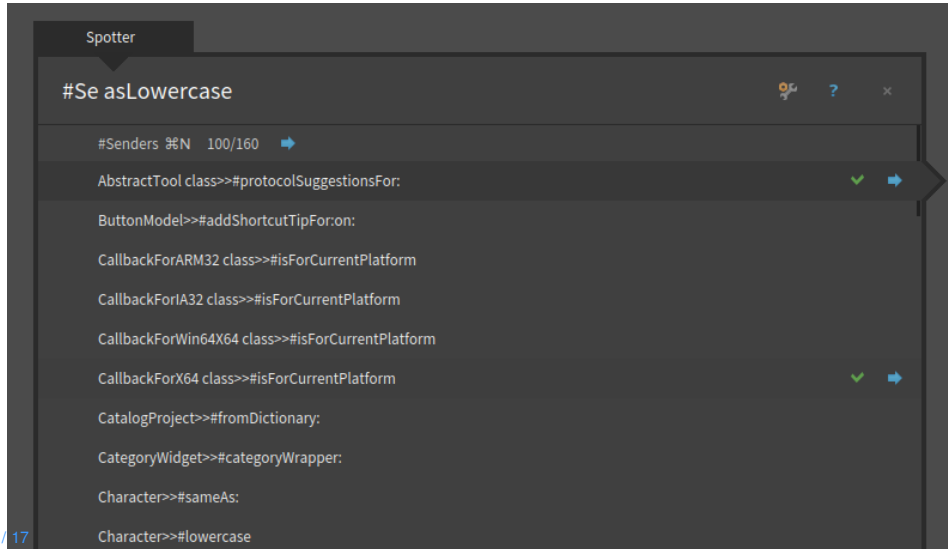
Looking for senders

- Command-n (for seNders)

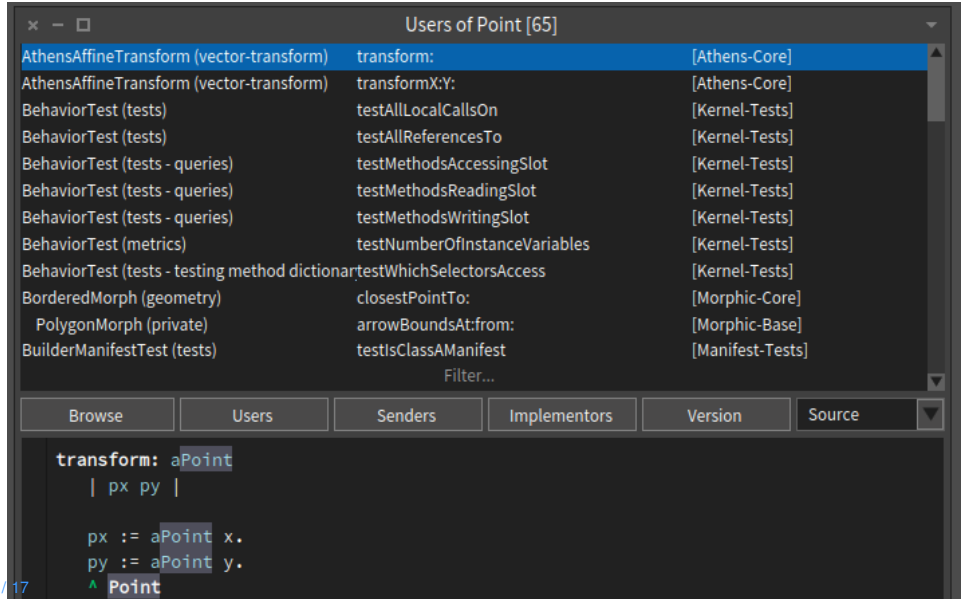
There is no senders of lowercase

Looking for senders (Spotter)

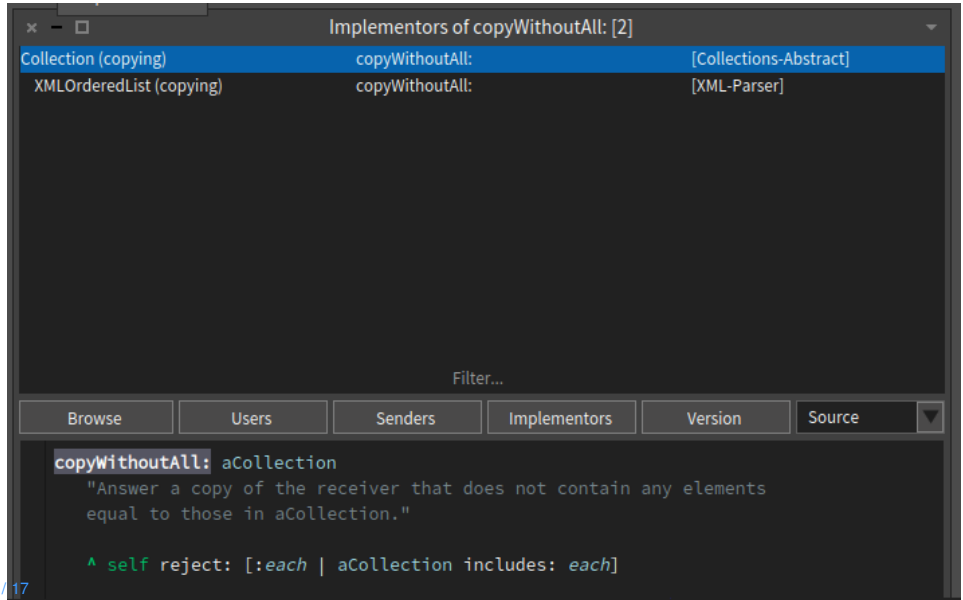
- Shit-Enter asLowercase #se



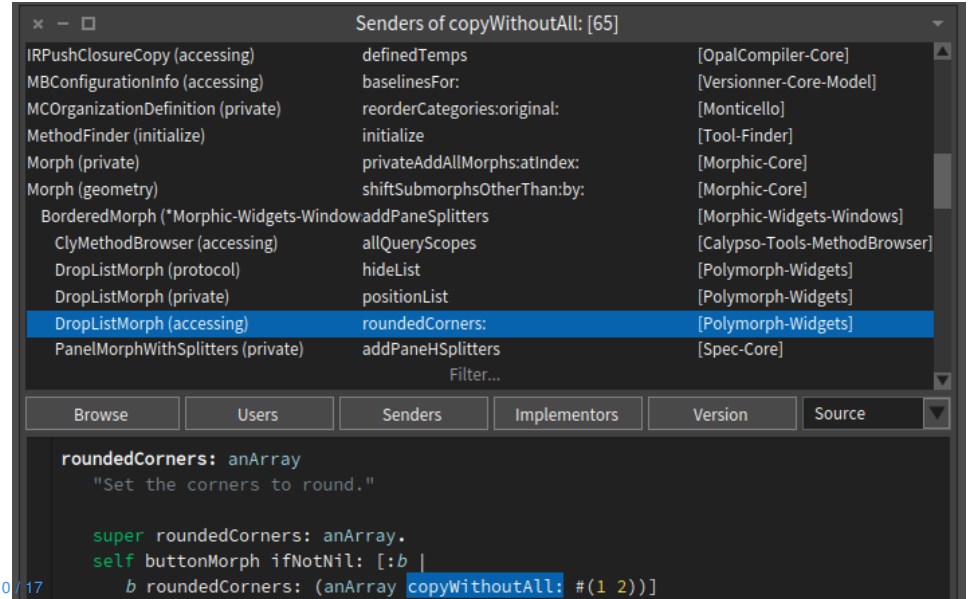
Looking for class references



Example: copyWithoutAll: implementors



Example: copyWithoutAll: senders



Looking for hex

```
testCopyEmptyWithoutAll
  "self debug: #testCopyEmptyWithoutAll"
  | res |
  res := self empty copyWithoutAll: self collectionWithElementsToRemove.
  self assert: res size = self empty size.
  self collectionWithElementsToRemove do: [ :each | self deny: (res includes: each) ]
```

Grabing instances and asking them

Too often we try to imagine objects

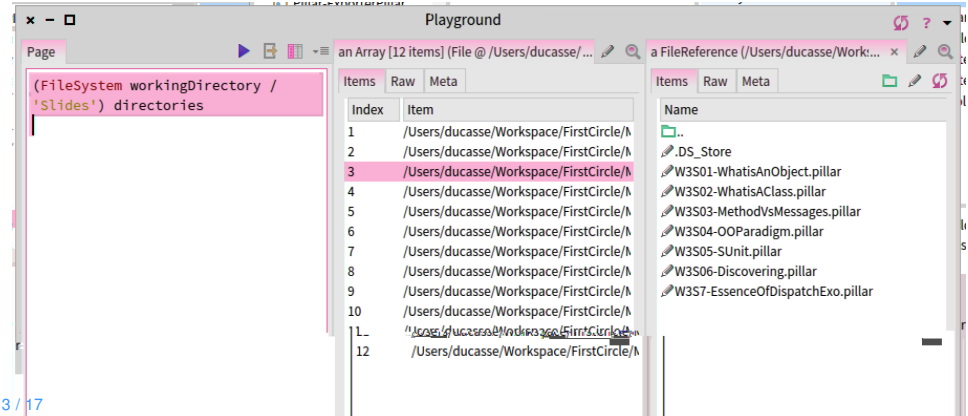
- Grab one as early as possible and
- Send message to this object
- Inspect it, talk to it

All the folders...

Find all the folder of this lectures containing pillar files

```
FileSystem workingDirectory / 'Slides'
```

```
(FileSystem workingDirectory / 'Slides') inspect
```



All the folders...

```
(FileSystem workingDirectory / 'Slides') directories
```

```
(FileSystem workingDirectory / 'Slides') directories
  first children first
```

```
(FileSystem workingDirectory / 'Slides') directories
  first children inspect
```

All the folders...

```
(FileSystem workingDirectory / 'Slides') directories
    select: [ :each | each children anySatisfy: [ :achil | achil extension = 'pillar']]
```

Summary

Looking for

- implementors
- senders
- direct references to classes

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