



**Learning Object-Oriented
Programming and Design with TDD**

Learning how to discover information

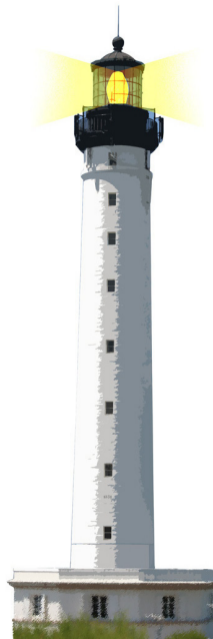
Stéphane Ducasse

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



<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)

The screenshot shows an IDE interface with two windows. The background window is titled 'Implementors of lowercase [1]' and contains a table with one row: 'Character (converting)' implementing 'lowercase' from the '[Kernel]' package. The foreground window is titled 'Implementors of asLowercase [3]' and contains a table with three rows: 'Character (converting)' implementing 'asLowercase' from the '[Kernel]' package, 'String (converting)' implementing 'asLowercase' from the '[Collections-Strings]' package, and 'WideString (converting)' implementing 'asLowercase' from the '[Collections-Strings]' package. Below the tables are buttons for 'Browse', 'Users', 'Senders', 'Implementors', 'Version', and 'Source'. The 'Implementors' button is currently selected. Below the foreground window, the source code for the 'asLowercase' method is visible, showing a comment and a call to 'characterSet.toLowercase: self'.

Implementor	Method	Package
Character (converting)	lowercase	[Kernel]

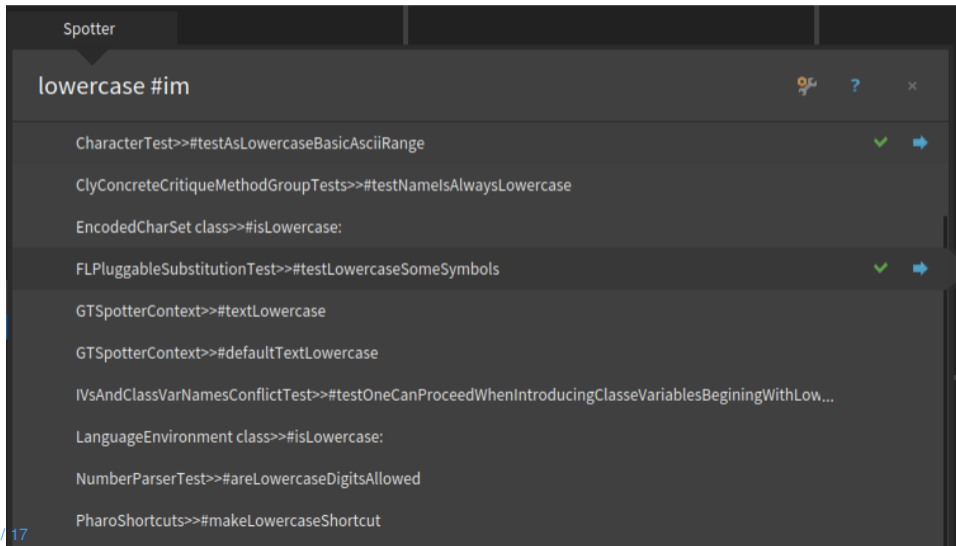
Implementor	Method	Package
Character (converting)	asLowercase	[Kernel]
String (converting)	asLowercase	[Collections-Strings]
WideString (converting)	asLowercase	[Collections-Strings]

```
lowercase
^ self asLowercase
```

```
asLowercase
    "If the receiver is uppercase, answer its matching lowercase Character."
    ^ self characterSet toLowercase: self
```

Looking for implementors (Spotter)

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



The screenshot shows the Spotter tool interface with a search query 'lowercase #im'. The results list several methods and classes:

- CharacterTest>>#testAsLowercaseBasicAsciiRange (checked)
- ClyConcreteCritiqueMethodGroupTests>>#testNamelsAlwaysLowercase
- EncodedCharSet class>>#isLowercase:
- FLPluggableSubstitutionTest>>#testLowercaseSomeSymbols (checked)
- GTPotterContext>>#textLowercase
- GTPotterContext>>#defaultTextLowercase
- IVsAndClassVarNamesConflictTest>>#testOneCanProceedWhenIntroducingClasseVariablesBeginingWithLow...
- LanguageEnvironment class>>#isLowercase:
- NumberParserTest>>#areLowercaseDigitsAllowed
- PharoShortcuts>>#makeLowercaseShortcut

Looking for senders

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



Looking for senders

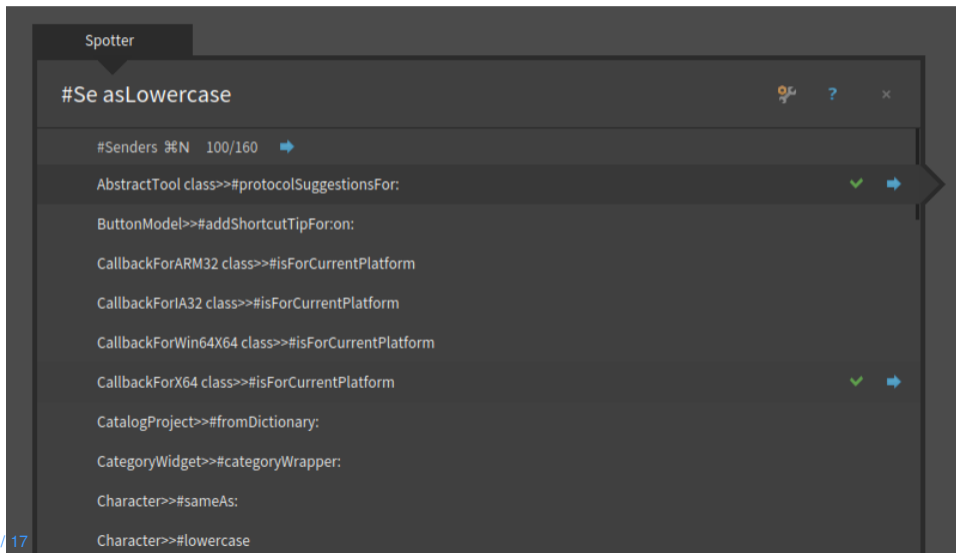
- Command-n (for seNders)

There is no senders of lowercase



Looking for senders (Spotter)

- Shit-Enter asLowercase #se



The screenshot shows the Spotter tool interface. At the top, there is a tab labeled 'Spotter'. Below it, the search query '#Se asLowercase' is entered. The results are displayed in a list format. The first result is 'AbstractTool class>>#protocolSuggestionsFor:' with a green checkmark and a blue arrow icon to its right. The second result is 'ButtonModel>>#addShortcutTipFor:on:'. The third result is 'CallbackForARM32 class>>#isForCurrentPlatform'. The fourth result is 'CallbackForIA32 class>>#isForCurrentPlatform'. The fifth result is 'CallbackForWin64X64 class>>#isForCurrentPlatform'. The sixth result is 'CallbackForX64 class>>#isForCurrentPlatform' with a green checkmark and a blue arrow icon to its right. The seventh result is 'CatalogProject>>#fromDictionary:'. The eighth result is 'CategoryWidget>>#categoryWrapper:'. The ninth result is 'Character>>#sameAs:'. The tenth result is 'Character>>#lowercase'.

```
Spotter
#Se asLowercase
#Senders ##N 100/160
AbstractTool class>>#protocolSuggestionsFor:
ButtonModel>>#addShortcutTipFor:on:
CallbackForARM32 class>>#isForCurrentPlatform
CallbackForIA32 class>>#isForCurrentPlatform
CallbackForWin64X64 class>>#isForCurrentPlatform
CallbackForX64 class>>#isForCurrentPlatform
CatalogProject>>#fromDictionary:
CategoryWidget>>#categoryWrapper:
Character>>#sameAs:
Character>>#lowercase
```

Looking for class references

The screenshot shows an IDE window titled "Users of Point [65]". The window contains a list of classes and their methods that use the `Point` class. The list is as follows:

Class	Method	Package
AthensAffineTransform (vector-transform)	transform:	[Athens-Core]
AthensAffineTransform (vector-transform)	transformX:Y:	[Athens-Core]
BehaviorTest (tests)	testAllLocalCallsOn	[Kernel-Tests]
BehaviorTest (tests)	testAllReferencesTo	[Kernel-Tests]
BehaviorTest (tests - queries)	testMethodsAccessingSlot	[Kernel-Tests]
BehaviorTest (tests - queries)	testMethodsReadingSlot	[Kernel-Tests]
BehaviorTest (tests - queries)	testMethodsWritingSlot	[Kernel-Tests]
BehaviorTest (metrics)	testNumberOfInstanceVariables	[Kernel-Tests]
BehaviorTest (tests - testing method dictionary)	testWhichSelectorsAccess	[Kernel-Tests]
BorderedMorph (geometry)	closestPointTo:	[Morphic-Core]
PolygonMorph (private)	arrowBoundsAt:from:	[Morphic-Base]
BuilderManifestTest (tests)	testIsClassAManifest	[Manifest-Tests]

Below the list is a "Filter..." input field. At the bottom of the window are several tabs: "Browse", "Users", "Senders", "Implementors", "Version", and "Source". The "Users" tab is currently selected.

The code snippet shown below the tabs is:

```
transform: aPoint
| px py |

px := aPoint x.
py := aPoint y.
^ Point
```


Example: copyWithoutAll: implementors

Implementors of copyWithoutAll: [2]

Collection (copying)	copyWithoutAll:	[Collections-Abstract]
XMLOrderedList (copying)	copyWithoutAll:	[XML-Parser]

Filter...

Browse Users Senders **Implementors** Version Source

```
copyWithoutAll: aCollection  
"Answer a copy of the receiver that does not contain any elements  
equal to those in aCollection."  
  
^ self reject: [:each | aCollection includes: each]
```

Example: copyWithoutAll: senders

Senders of copyWithoutAll: [65]

IRPushClosureCopy (accessing)	definedTemps	[OpalCompiler-Core]
MBConfigurationInfo (accessing)	baselinesFor:	[Versionner-Core-Model]
MCOrganizationDefinition (private)	reorderCategories:original:	[Monticello]
MethodFinder (initialize)	initialize	[Tool-Finder]
Morph (private)	privateAddAllMorphs:atIndex:	[Morphic-Core]
Morph (geometry)	shiftSubmorphsOtherThan:by:	[Morphic-Core]
BorderedMorph (*Morphic-Widgets-WindowaddPaneSplitters		[Morphic-Widgets-Windows]
ClyMethodBrowser (accessing)	allQueryScopes	[Calypso-Tools-MethodBrowser]
DropListMorph (protocol)	hideList	[Polymorph-Widgets]
DropListMorph (private)	positionList	[Polymorph-Widgets]
DropListMorph (accessing)	roundedCorners:	[Polymorph-Widgets]
PanelMorphWithSplitters (private)	addPaneHSplitters	[Spec-Core]

Filter...

Browse Users Senders Implementors Version Source

```
roundedCorners: anArray
  "Set the corners to round."

  super roundedCorners: anArray.
  self buttonMorph ifNotNil: [:b |
    b roundedCorners: (anArray copyWithoutAll: #(1 2))]
```

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) ]
```

Grabbing 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

The screenshot shows a REPL playground window with the following content:

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

The output is an array of 12 items, each representing a directory path. The third item is highlighted:

Index	Item
1	/Users/ducasse/Workspace/FirstCircle/
2	/Users/ducasse/Workspace/FirstCircle/
3	/Users/ducasse/Workspace/FirstCircle/
4	/Users/ducasse/Workspace/FirstCircle/
5	/Users/ducasse/Workspace/FirstCircle/
6	/Users/ducasse/Workspace/FirstCircle/
7	/Users/ducasse/Workspace/FirstCircle/
8	/Users/ducasse/Workspace/FirstCircle/
9	/Users/ducasse/Workspace/FirstCircle/
10	/Users/ducasse/Workspace/FirstCircle/
11	/Users/ducasse/Workspace/FirstCircle/
12	/Users/ducasse/Workspace/FirstCircle/

The right-hand pane shows a file reference view of the selected directory, listing the following files:

- ..
- .DS_Store
- W3S01-WhatisAnObject.pillar
- W3S02-WhatisAClass.pillar
- W3S03-MethodVsMessages.pillar
- W3S04-OOParadigm.pillar
- W3S05-SUnit.pillar
- W3S06-Discovering.pillar
- W3S7-EssenceOfDispatchExo.pillar

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 Mocc by

Damien Cassou, Stéphane Ducasse, Luc Fabresse
<http://mocc.pharo.org>



Except where otherwise noted, this work is licensed under CC BY-NC-ND 3.0 France
<https://creativecommons.org/licenses/by-nc-nd/3.0/fr/>