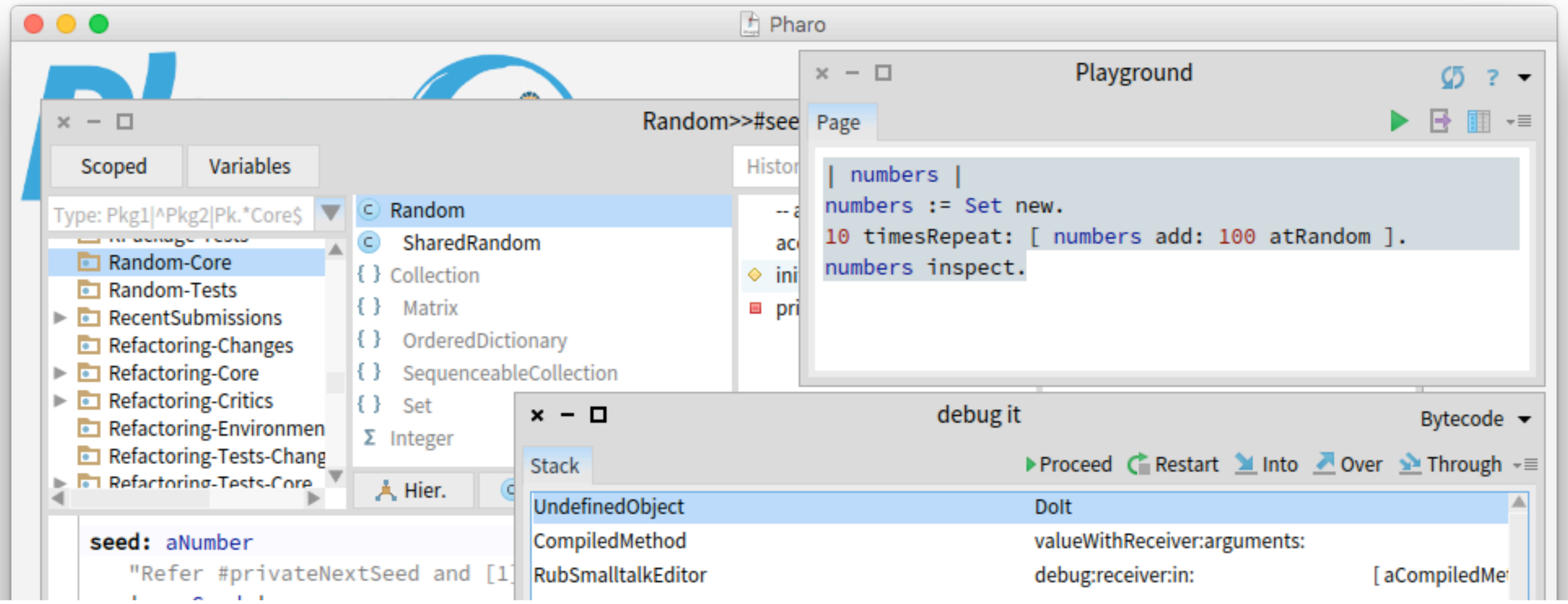
ILLIMANI Memory Profiler at Work: Identifying Object Allocation Sites

Sebastian JORDAN MONTAÑO, Guillermo POLITO, Stéphane DUCASSE, Pablo TESONE

Inria, Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL

Name origins

Pharo Programming Language



```
seed: aNumber
	"Refer #privateNextSeed and [1]
```

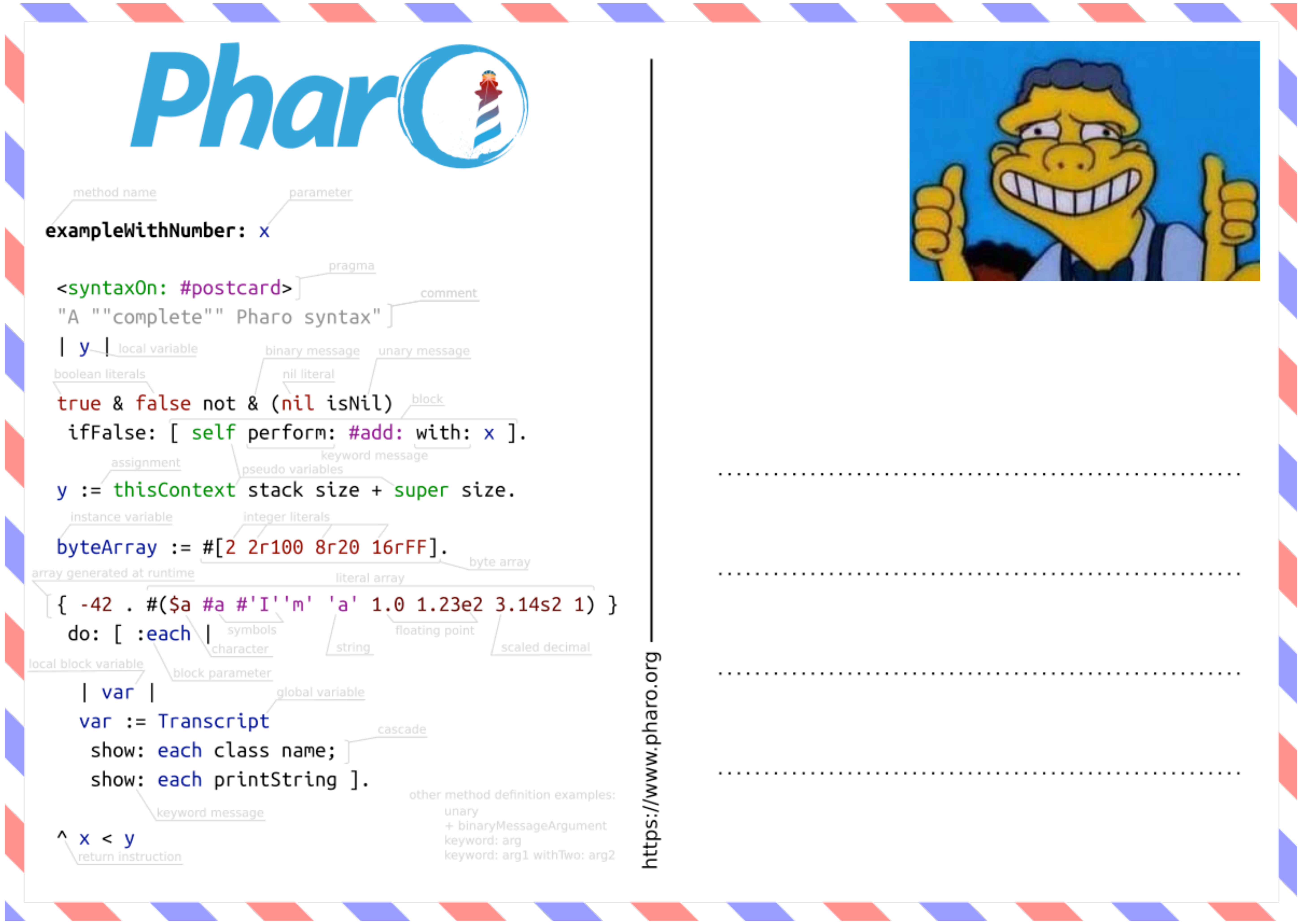
```
| numbers |
numbers := Set new.
10 timesRepeat: [ numbers add: 100 atRandom ].
numbers inspect.
```

```
UndefinedObject        DoIt
CompiledMethod         valueWithReceiver:arguments:
RubSmalltalkEditor     debug:receiver:in:            [ aCompiledMe
```

github.com/pharo-project/pharo

pharo.org

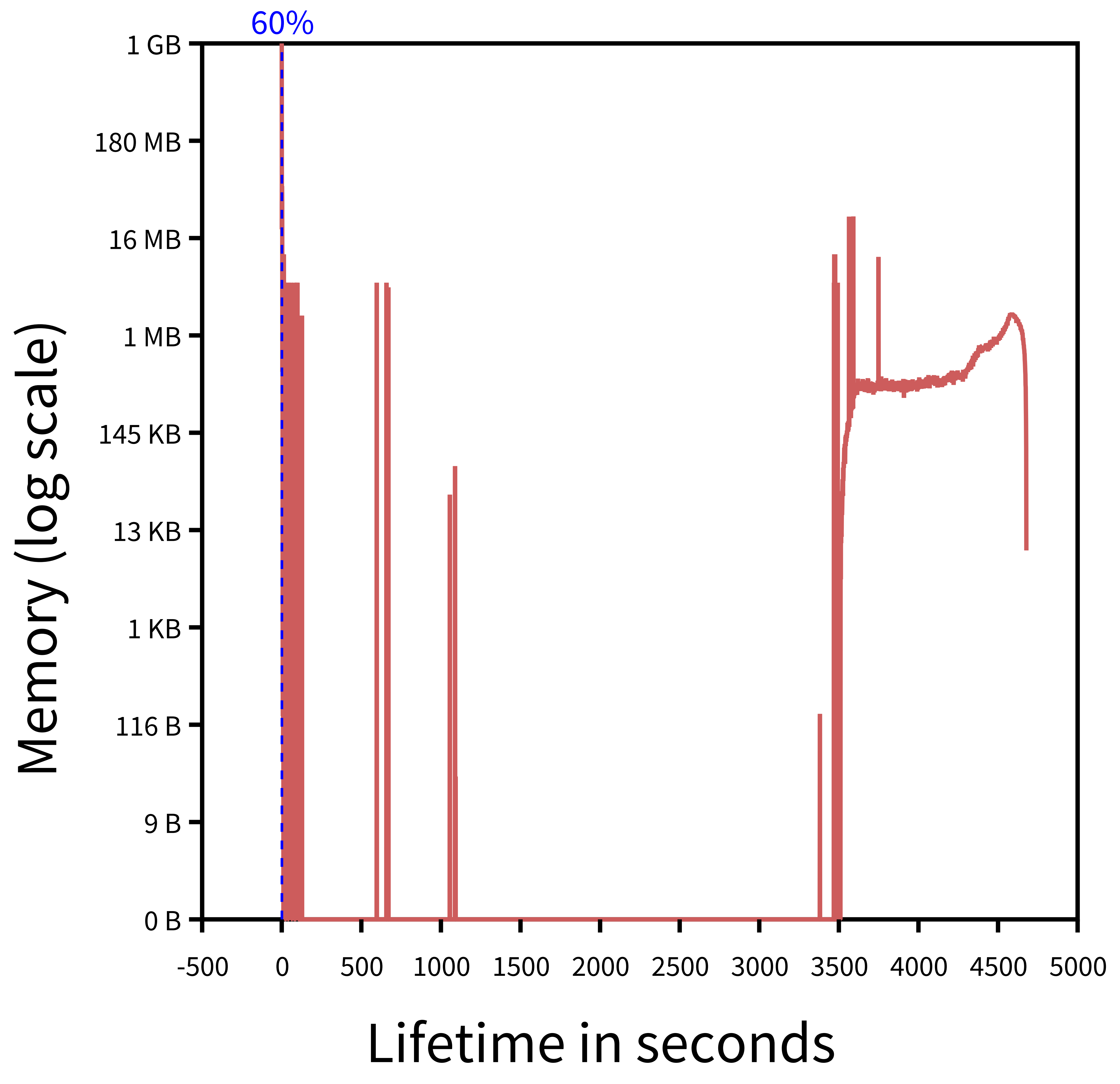
Pharo's Syntax


Pharo
method name
parameter
pragma
comment
local variable
binary message
unary message
boolean literals
nil literal
block
keyword message
assignment
pseudo variables
instance variable
integer literals
byte array
array generated at runtime
literal array
symbols
character
string
floating point
scaled decimal
local block variable
block parameter
global variable
cascade
keyword message
return instruction
https://www.pharo.org

```
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
```

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

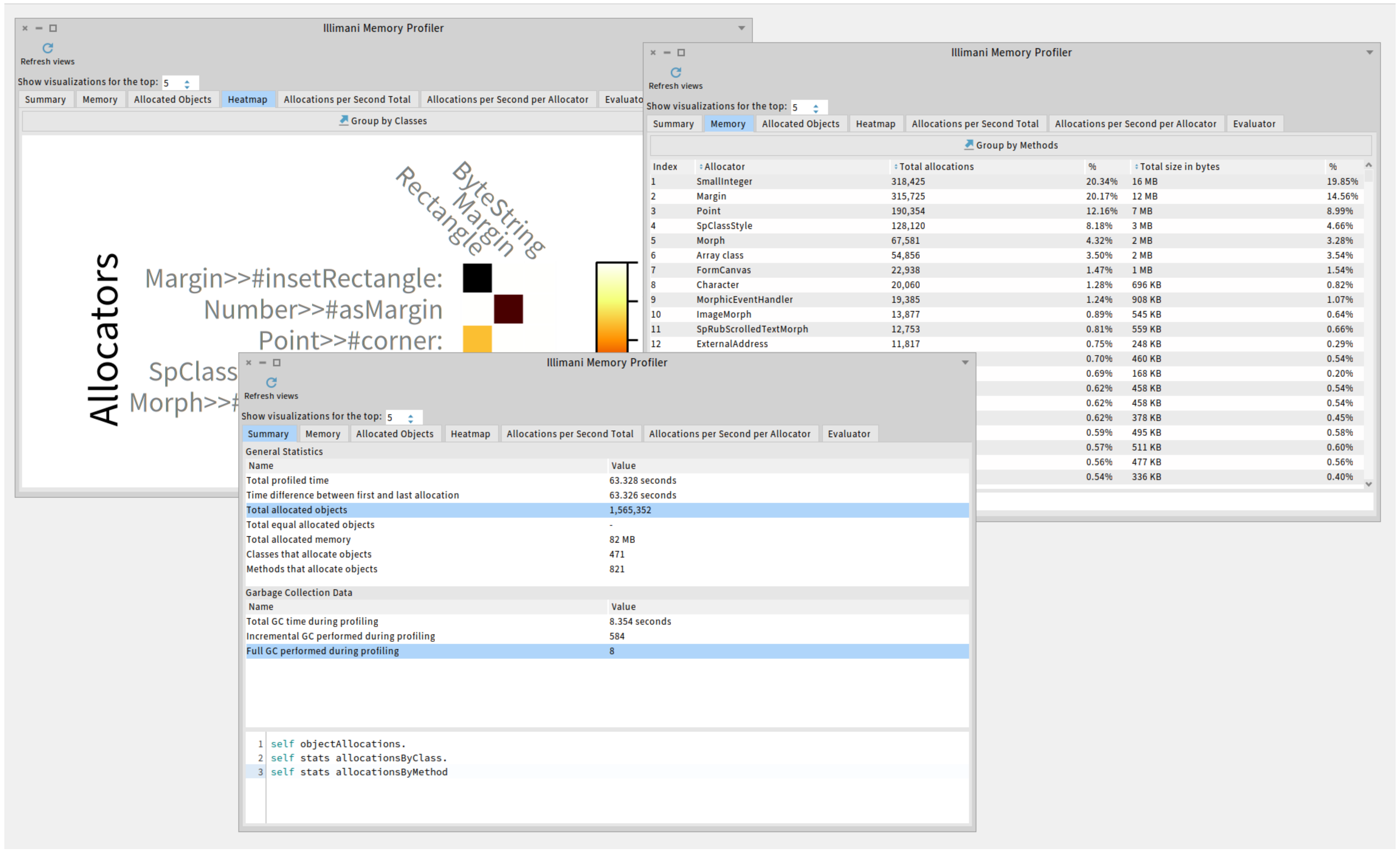
Object Allocation Sites



Object Allocation Sites

- Memory leaks
- Optimize the application
- Optimize the GC

A Memory Profiler for Pharo



Index	Allocator	Total allocations	%	Total size in bytes	%
1	SmallInteger	318,425	20.34%	16 MB	19.85%
2	Margin	315,725	20.17%	12 MB	14.56%
3	Point	190,354	12.16%	7 MB	8.99%
4	SpClassStyle	128,120	8.18%	3 MB	4.66%
5	Morph	67,581	4.32%	2 MB	3.28%
6	Array class	54,856	3.50%	2 MB	3.54%
7	FormCanvas	22,938	1.47%	1 MB	1.54%
8	Character	20,060	1.28%	696 KB	0.82%
9	MorphicEventHandler	19,385	1.24%	908 KB	1.07%
10	ImageMorph	13,877	0.89%	545 KB	0.64%
11	SpRubScrolledTextMorph	12,753	0.81%	559 KB	0.66%
12	ExternalAddress	11,817	0.75%	248 KB	0.29%
			0.70%	460 KB	0.54%
			0.69%	168 KB	0.20%
			0.62%	458 KB	0.54%
			0.62%	458 KB	0.54%
			0.62%	378 KB	0.45%
			0.59%	495 KB	0.58%
			0.57%	511 KB	0.60%
			0.56%	477 KB	0.56%
			0.54%	336 KB	0.40%

General Statistics

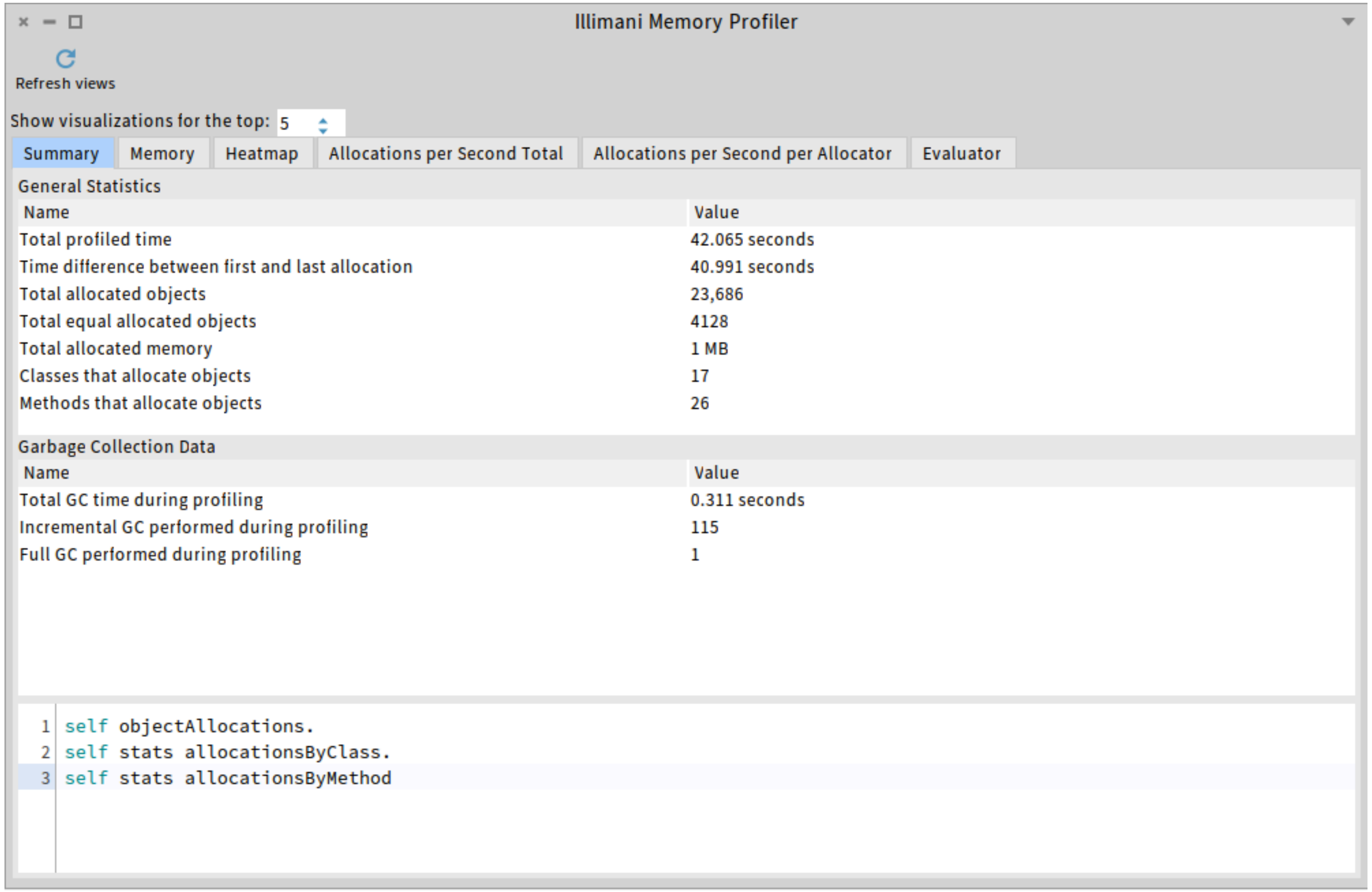
Name	Value
Total profiled time	63.328 seconds
Time difference between first and last allocation	63.326 seconds
Total allocated objects	1,565,352
Total equal allocated objects	-
Total allocated memory	82 MB
Classes that allocate objects	471
Methods that allocate objects	821

Garbage Collection Data

Name	Value
Total GC time during profiling	8.354 seconds
Incremental GC performed during profiling	584
Full GC performed during profiling	8

```
self objectAllocations.
self stats allocationsByClass.
self stats allocationsByMethod
```

A Memory Profiler for Pharo



Show visualizations for the top: 5

General Statistics

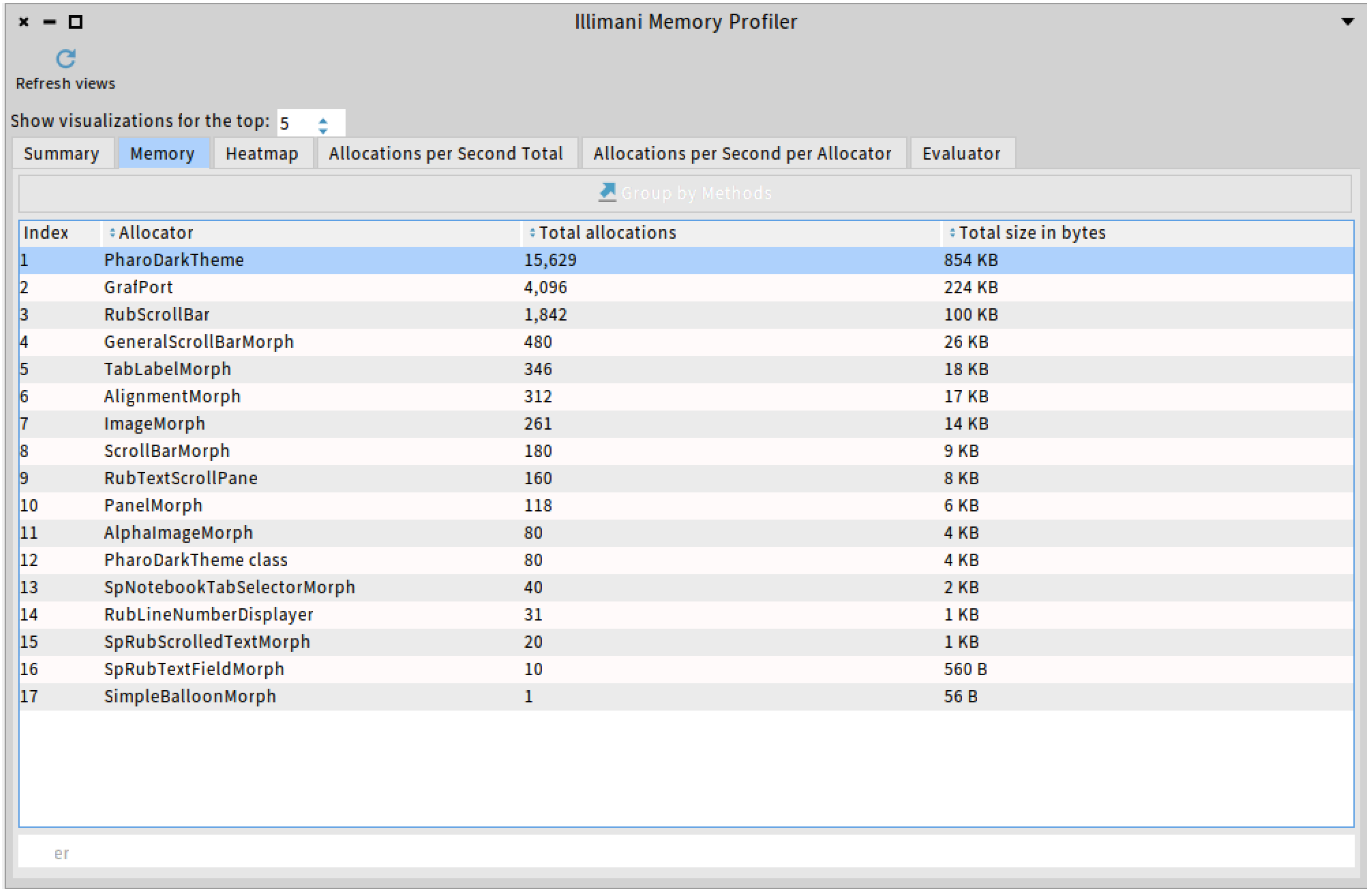
Name	Value
Total profiled time	42.065 seconds
Time difference between first and last allocation	40.991 seconds
Total allocated objects	23,686
Total equal allocated objects	4128
Total allocated memory	1 MB
Classes that allocate objects	17
Methods that allocate objects	26

Garbage Collection Data

Name	Value
Total GC time during profiling	0.311 seconds
Incremental GC performed during profiling	115
Full GC performed during profiling	1

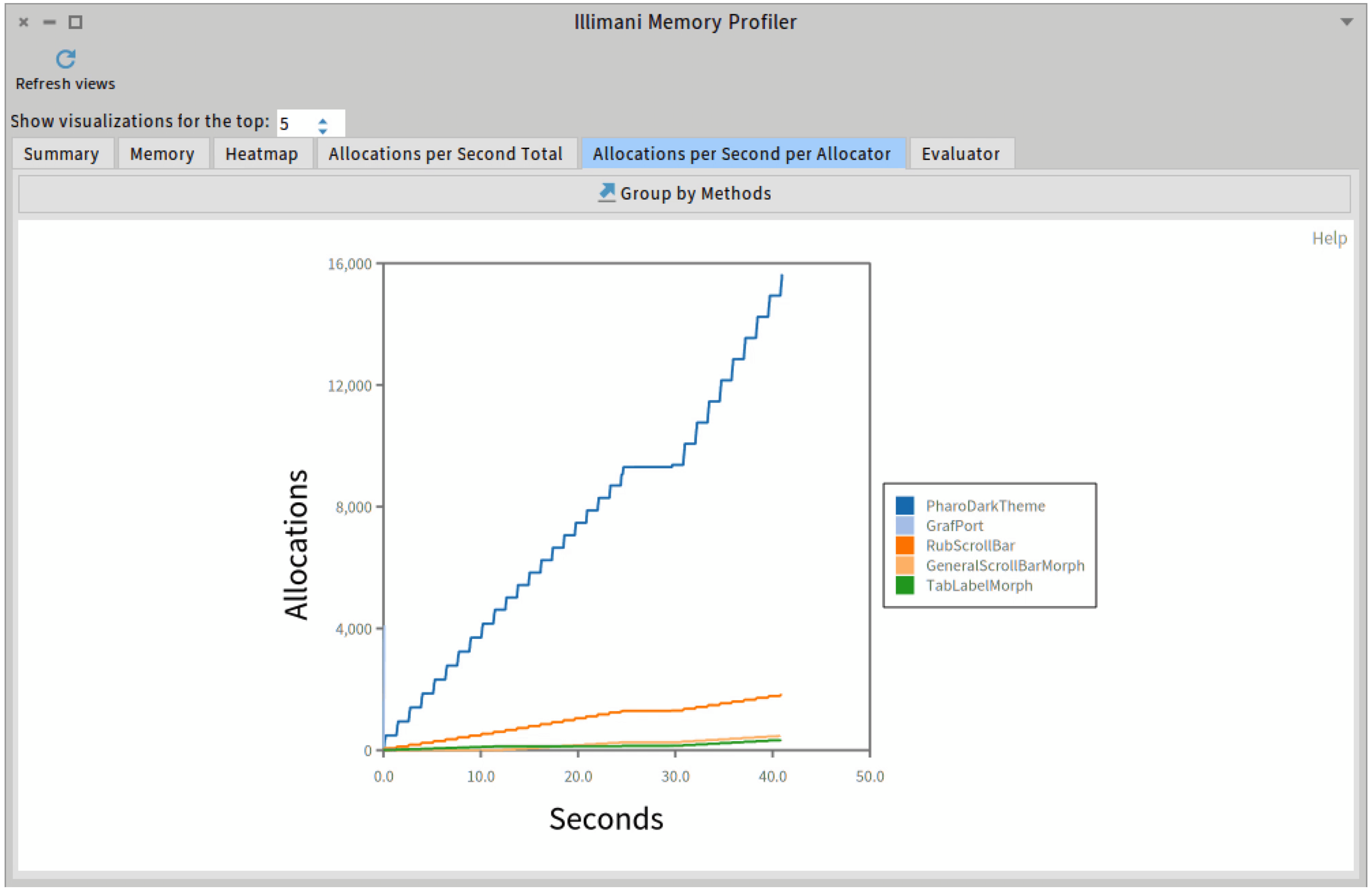
```
self objectAllocations.
self stats allocationsByClass.
self stats allocationsByMethod
```

A Memory Profiler for Pharo

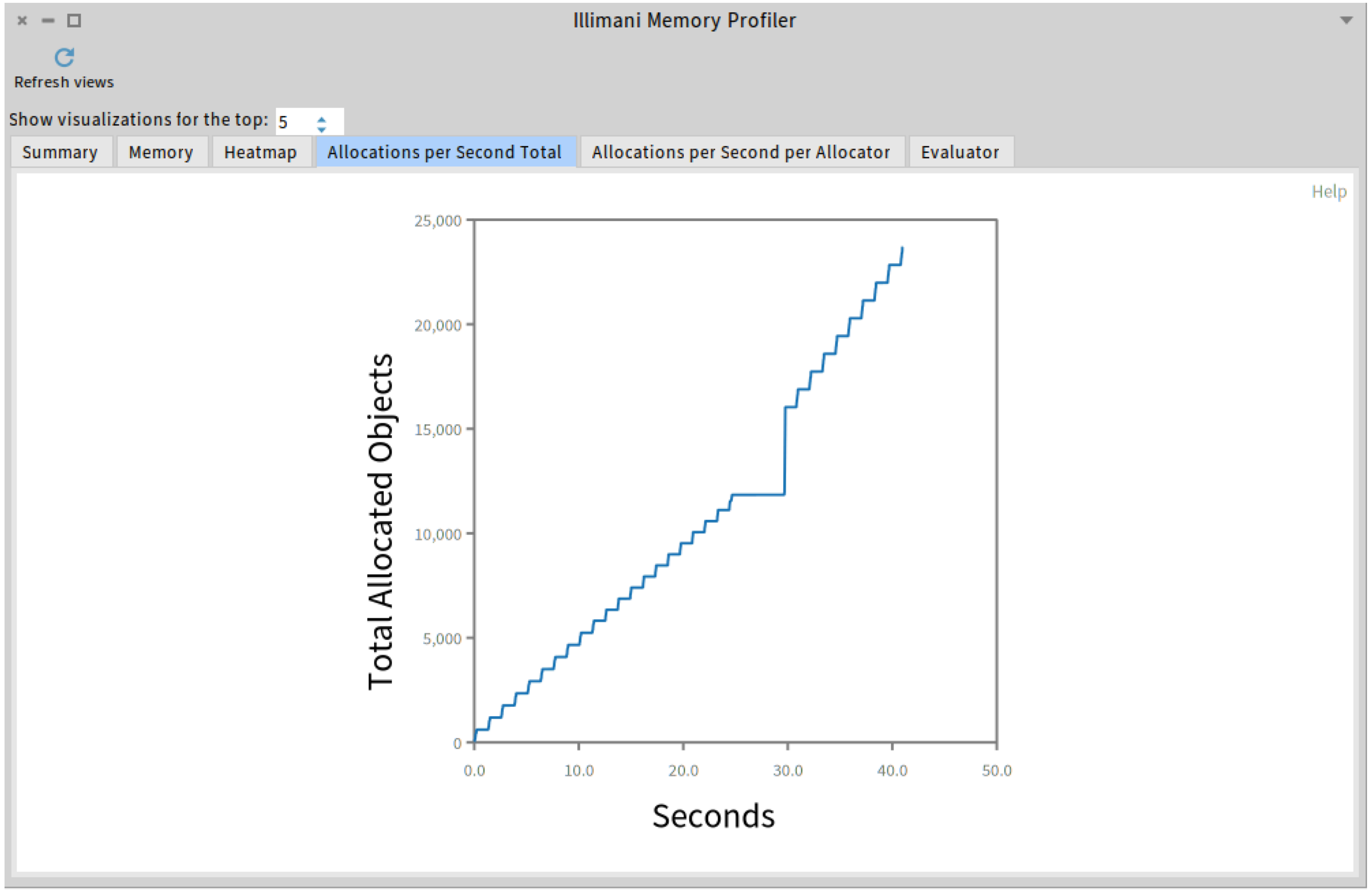

Illimani Memory Profiler
Refresh views
Show visualizations for the top: 5
Summary
Memory
Heatmap
Allocations per Second Total
Allocations per Second per Allocator
Evaluator
Group by Methods
er

Index	Allocator	Total allocations	Total size in bytes
1	PharoDarkTheme	15,629	854 KB
2	GrafPort	4,096	224 KB
3	RubScrollBar	1,842	100 KB
4	GeneralScrollBarMorph	480	26 KB
5	TabLabelMorph	346	18 KB
6	AlignmentMorph	312	17 KB
7	ImageMorph	261	14 KB
8	ScrollBarMorph	180	9 KB
9	RubTextScrollPane	160	8 KB
10	PanelMorph	118	6 KB
11	AlphaImageMorph	80	4 KB
12	PharoDarkTheme class	80	4 KB
13	SpNotebookTabSelectorMorph	40	2 KB
14	RubLineNumberDisplayer	31	1 KB
15	SpRubScrolledTextMorph	20	1 KB
16	SpRubTextFieldMorph	10	560 B
17	SimpleBalloonMorph	1	56 B

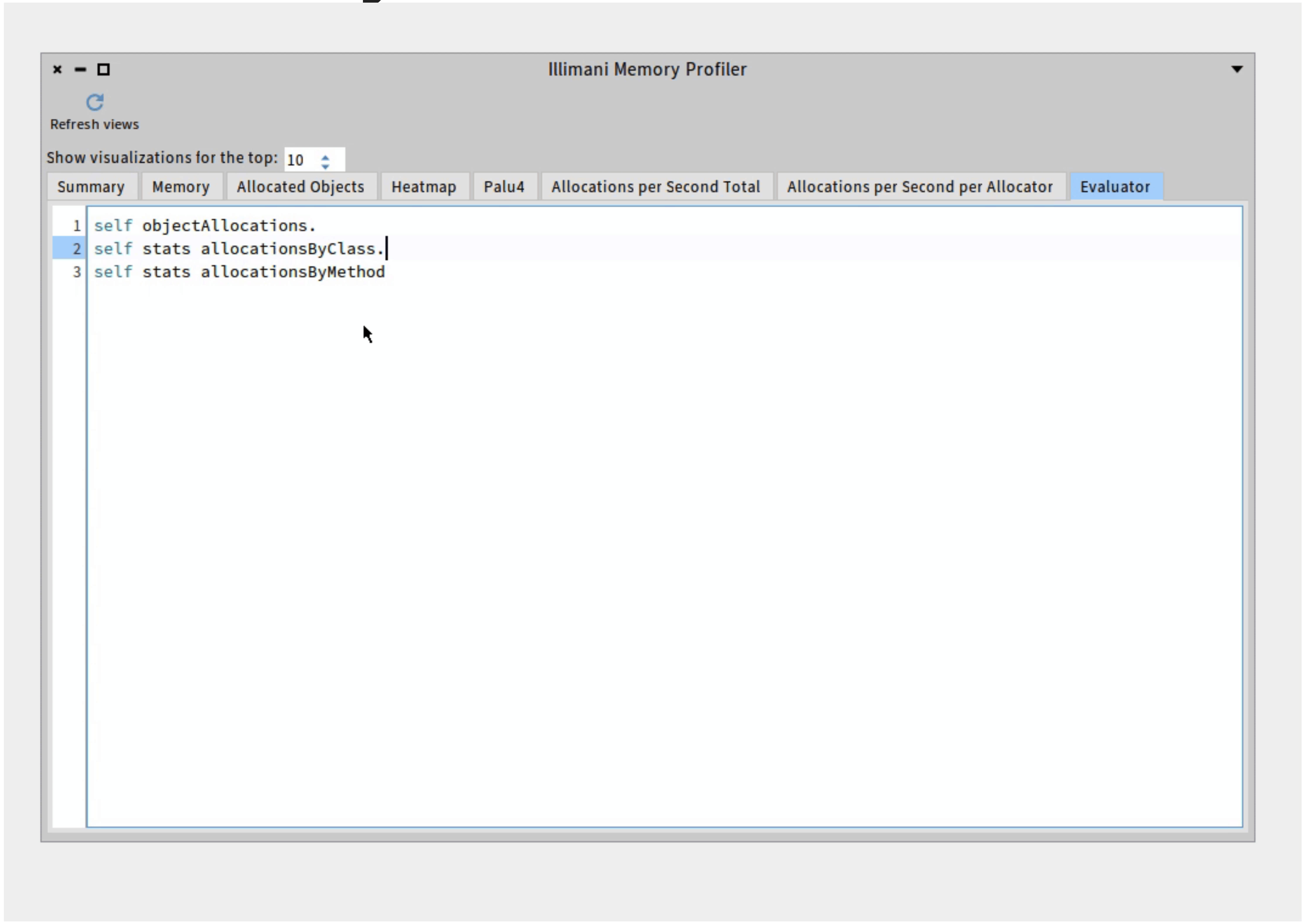
A Memory Profiler for Pharo



A Memory Profiler for Pharo



A Memory Profiler for Pharo


Illimani Memory Profiler
Refresh views
Show visualizations for the top: 10
Summary
Memory
Allocated Objects
Heatmap
Palu4
Allocations per Second Total
Allocations per Second per Allocator
Evaluator

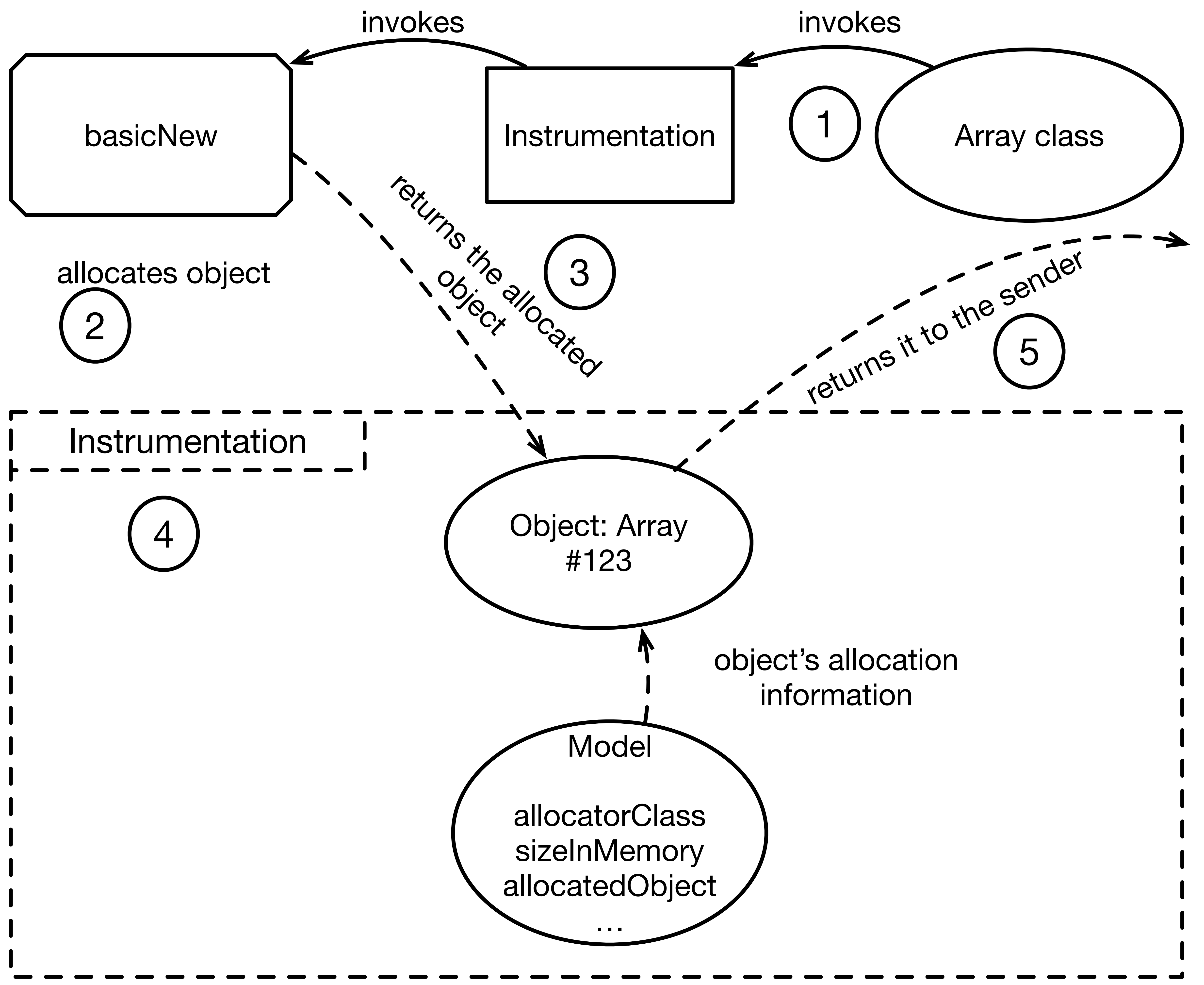
```
self objectAllocations.
self stats allocationsByClass.
self stats allocationsByMethod
```

Precise Memory Profiler

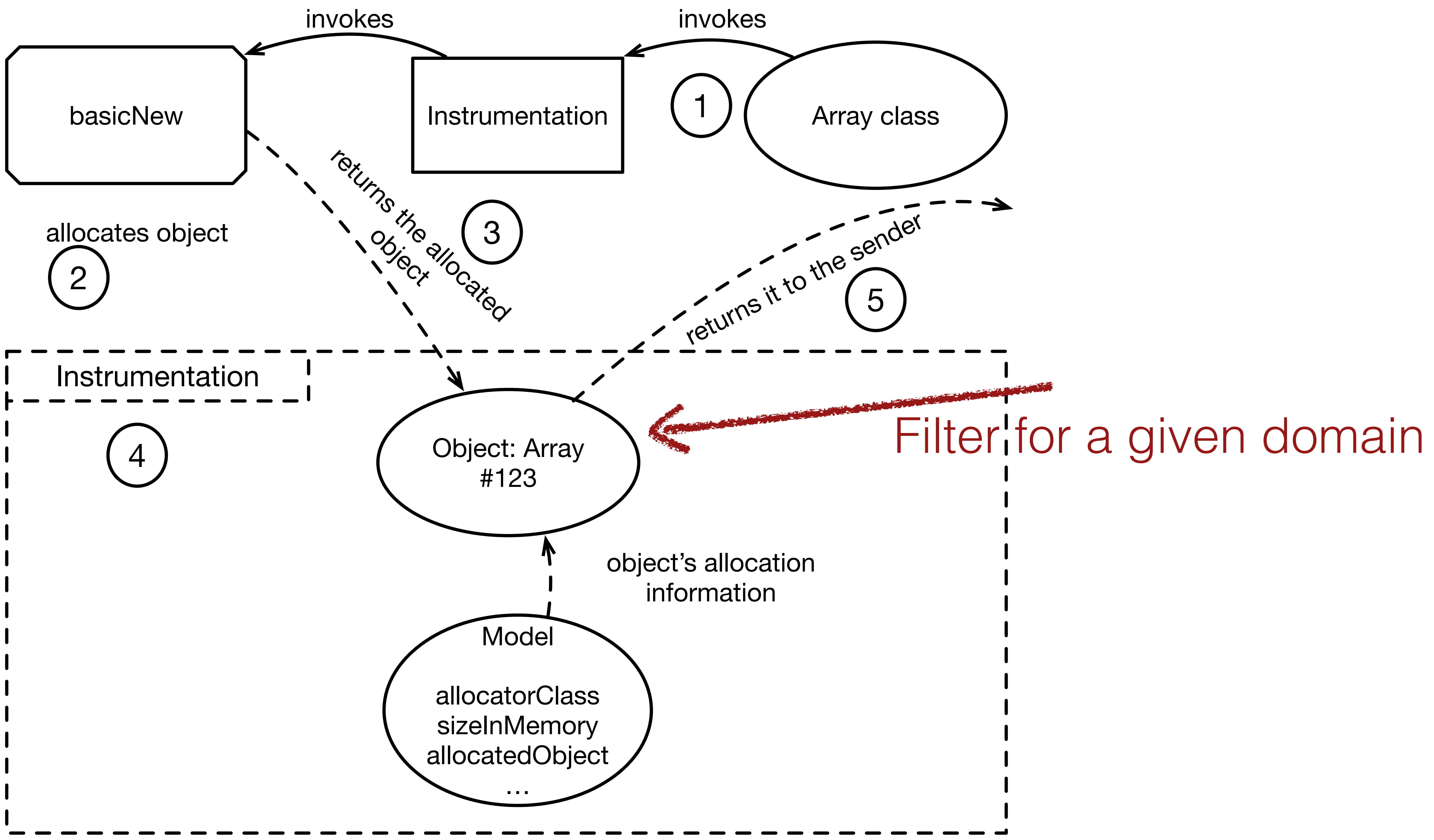
We instrumented the 4 methods that allocate objects In Pharo:

- `Behavior >> #basicNew`
- `Behavior >> #basicNew:`
- `Array class >> #new:`
- `Number >> @`

Precise Memory Profiler



Precise Memory Profiler



Identifying Object Allocation Sites

- Target Application: MorphicUI
- Morphic has 659 classes and 11126 methods (rough metric)
- Filtered domain: We captured only the Color Allocations

Identifying Object Allocation Sites

We opened 30 Pharo core tools and we let each of the instances of the tools render for 100 Morphic rendering cycles.

The tools are: Iceberg, Playground, and the Pharo Inspector. We opened 10 of each making 30 in total.

Identifying Object Allocation Sites

Demo

Identifying Object Allocation Sites

Top 5 allocator classe summary

Allocator Class	Allocated Colors	%
PharoDarkTheme	15,629	66 %
GrafPort	4,096	17 %
RubScrollBar	1,842	8 %
GeneralScrollBarMorph	480	2 %
TabLabelMorph	346	1 %
Rest of the classes	1293	2 %

Identifying Object Allocation Sites

Allocation site

Top 5 allocator classe summary

Allocator Class	Allocated Colors	%
PharoDarkTheme	15,629	66 %
GrafPort	4,096	17 %
RubScrollBar	1,842	8 %
GeneralScrollBarMorph	480	2 %
TabLabelMorph	346	1 %
Rest of the classes	1293	2 %

Color Palette Solution

What about caching the Color creation?

Color Palette Solution

Demo

Color Palette Solution

Baseline vs Color Palette implementation

Allocator Class	Baseline	Color Palette	Difference
PharoDarkTheme	15,629	0	-
RubScrollBar	4,096	4,096	1x
Total allocations	23,686	7,974	3x

Other Allocation Sites

We profiled again the same execution setup, opening 30 Pharo tools, but this time not filtering the allocated objects but capturing them all.

Other Allocation Sites

Top 5 allocator classe summary

Allocator Class	Allocated Colors	%
Rectangle	699,625	45 %
Margin	300,663	19 %
ByteString	111,662	7 %
OrderedCollection	78,474	5 %
WriteStream	60,448	4 %
Rest of the classes	314,480	20 %

Discussion

- Cost of the instrumentation
- Stressing the GC

Conclusions

- ILLIMANI is a memory profiler that can precisely capture object allocations. It provides a rich object model that allows a user to query and group the object allocations at constant time.
- We validated our tool profiling the opening of 30 Pharo core tools. We were able to find object allocation sites. UITheme was making 99,9% of redundant allocations.
- We found other allocation sites when profiling all the allocations produced.

Merci

About Pharo
github.com/pharo-project/pharo
pharo.org

open source
initiative
Approved License®

Give our profiler a try!
https://github.com/jordanmontt/illimani-memory-profiler

Contact me
https://github.com/jordanmontt
sebastian.jordan@inria.fr