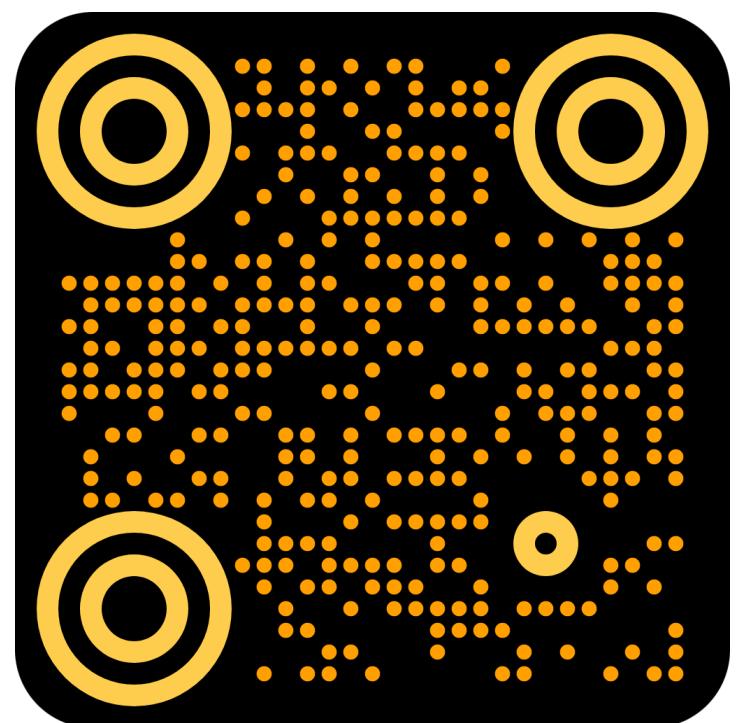


# Ease Virtual Machine Level Tooling With Language Level Ordinary Object Pointers

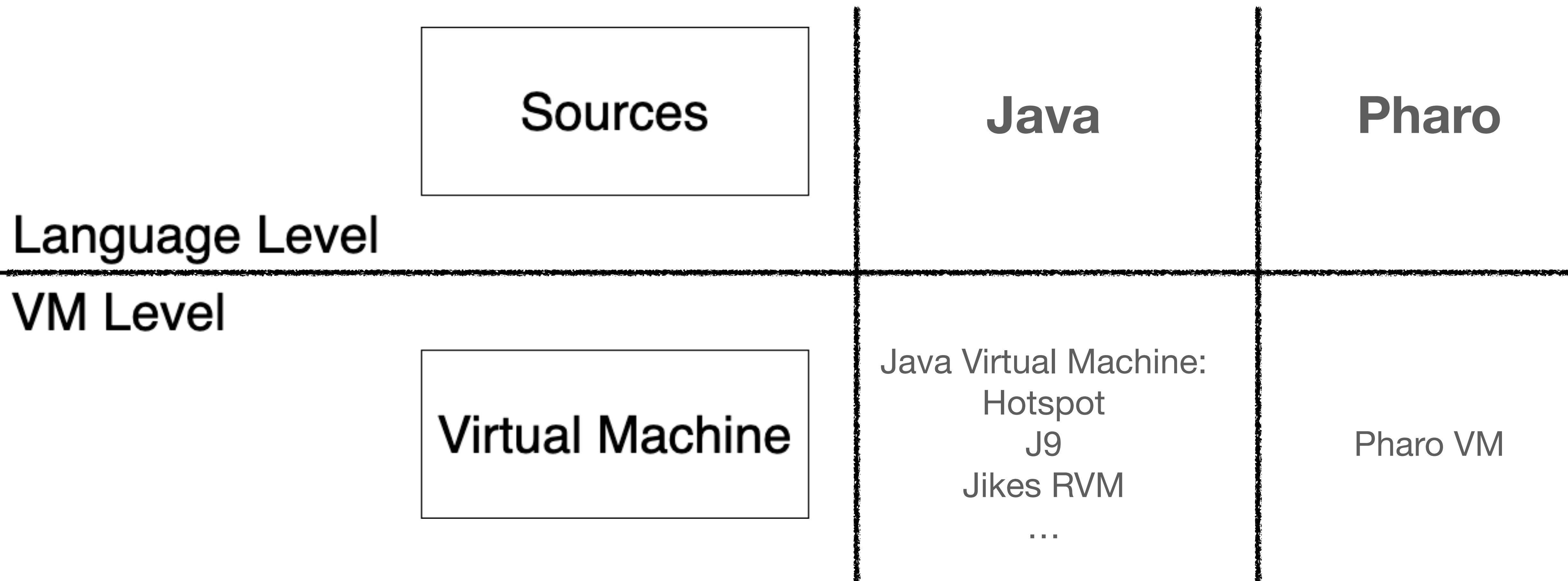


Pierre Misse-Chanabier  
Theo Rogliano



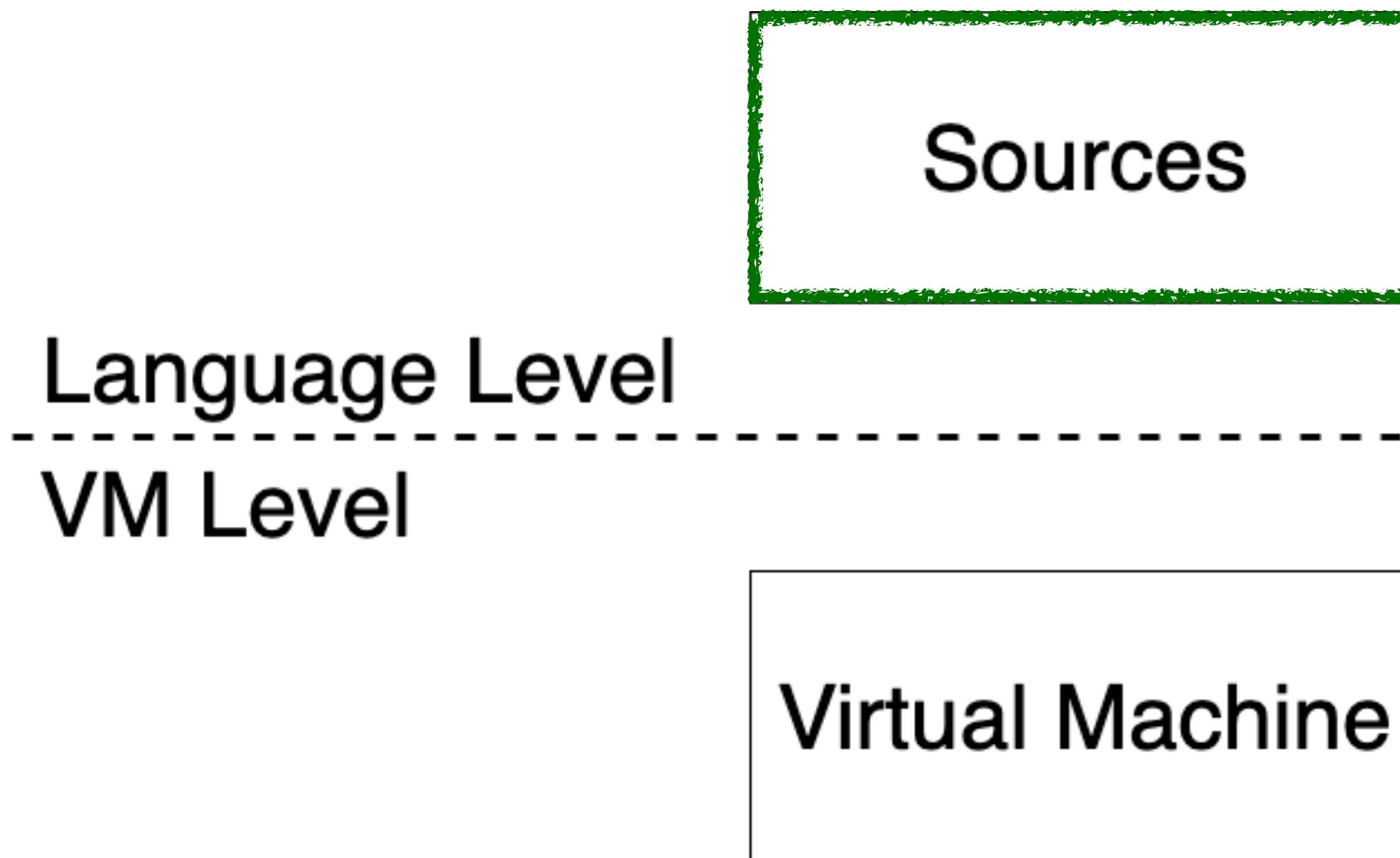
# Who Does Not Love Tools ?

## Tooling Levels



# Who Does Not Love Tools ?

## Tools at the Language Level



Debuggers, Profilers, Compilers, Code Browser,  
XUnit, Refactoring, Project Management

.....



# Who Does Not Love Tools ?

## Tools at the VM Level

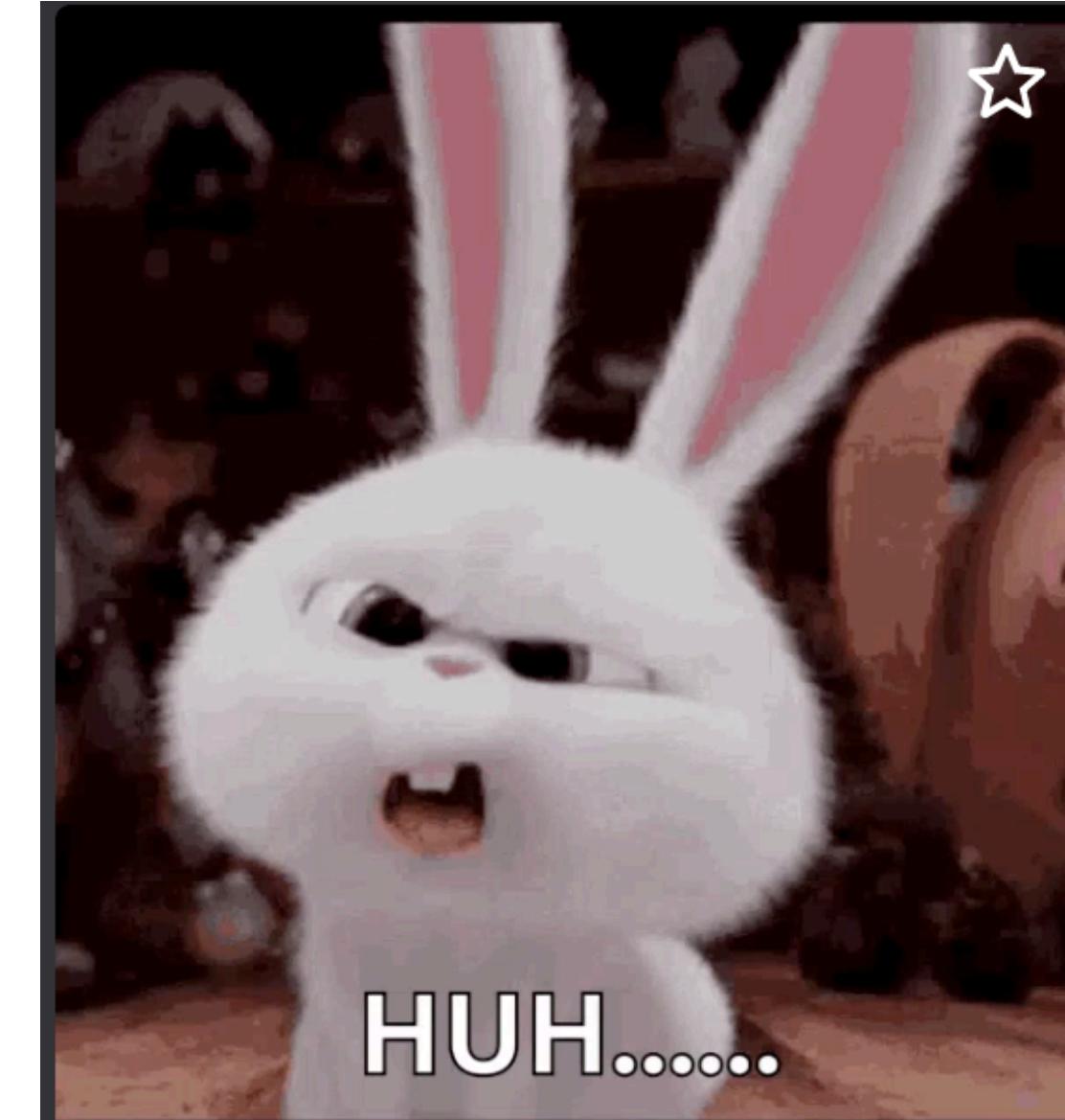
Sources

Language Level

-----  
VM Level

Virtual Machine

Debuggers, Profilers, Memory visualisation, Bootstrap  
Difficult to write, requires expertise



# Who Does Not Love Tools ?

## Why Should we Care About VM Level Tools ?

```
Form >> #scaledByDisplayScaleFactor
    self halt.
    ^ self scaledToSize: self extent * self currentWorld displayScaleFactor
```

# Who Does Not Love Tools ?

## Don't Close the Environment !



**Form >> #scaledByDisplayScaleFactor**  
**self halt.**  
^ **self scaledToSize: self extent \* self currentWorld displayScaleFactor**



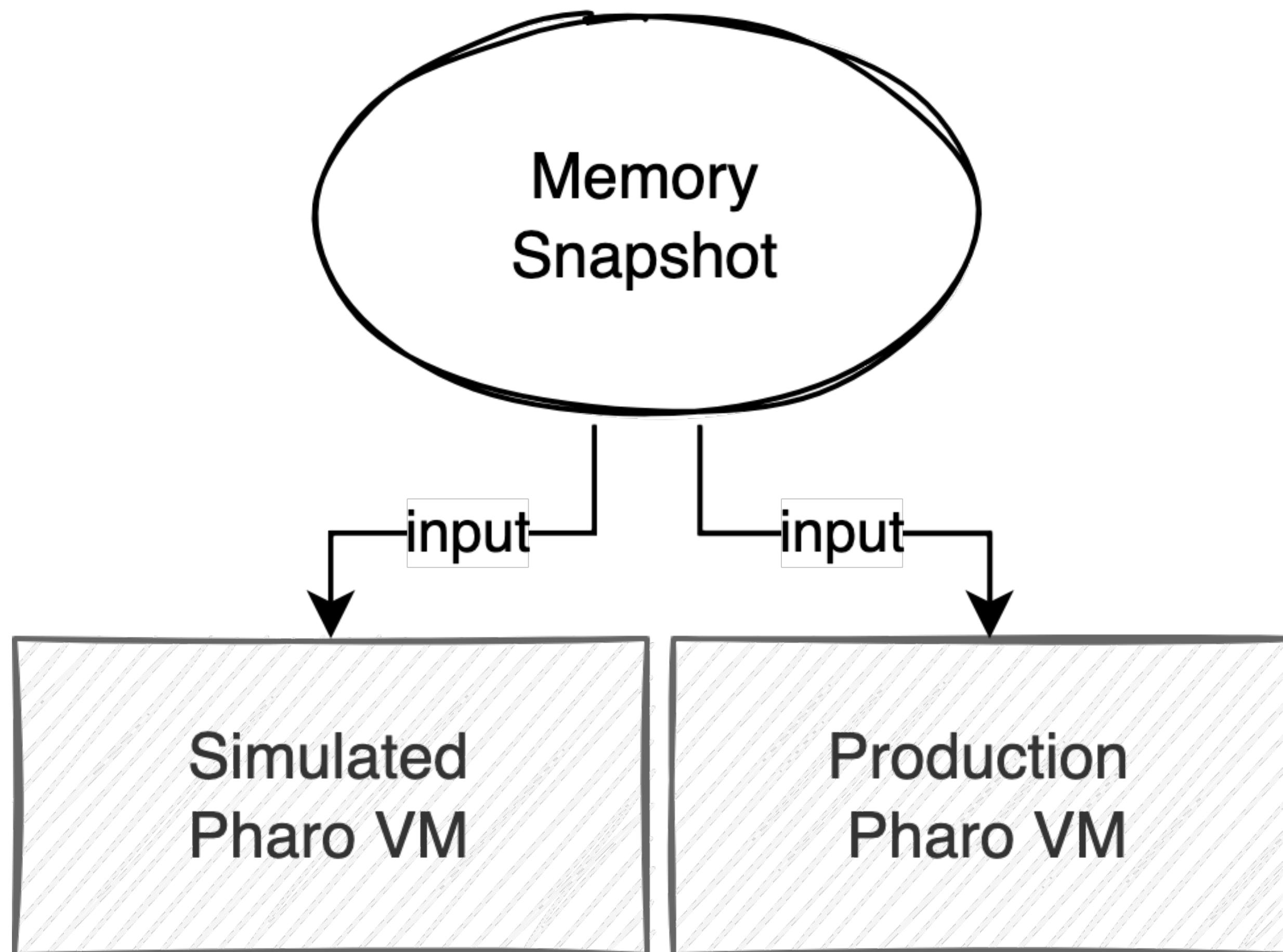
# Who Does Not Love Tools ?

We need the VM support !

```
Halt
SmallInteger(Object)>>haltOnce
Form>>scaledByDisplayScaleFactor
ThemeIcons>>iconNamed:
MorphicRootRenderer(Object)>>iconNamed:
MorphicRootRenderer(OSWorldRenderer)>>setAttributesDefault
MorphicRootRenderer class(OSWorldRenderer class)>>forWorld:
[ :arg5 | tmp2 := arg5 forWorld: arg1 ] in AbstractWorldRenderer
FullBlockClosure(BlockClosure)>>cull:
[ :arg4 | (arg1 value: arg4) ifTrue: [ ^ arg2 cull: arg4 ] ] in
arg2 cull...etc...
OrderedCollection>>do:
OrderedCollection(Collection)>>detect;ifFound;ifNone:
OrderedCollection(Collection)>>detect;ifFound:
AbstractWorldRenderer class>>detectCorrectOneForWorld:
```

# Let's Code VM Level Tools !

Looking for the Class Form in Memory ...



# Let's Code VM Level Tools !

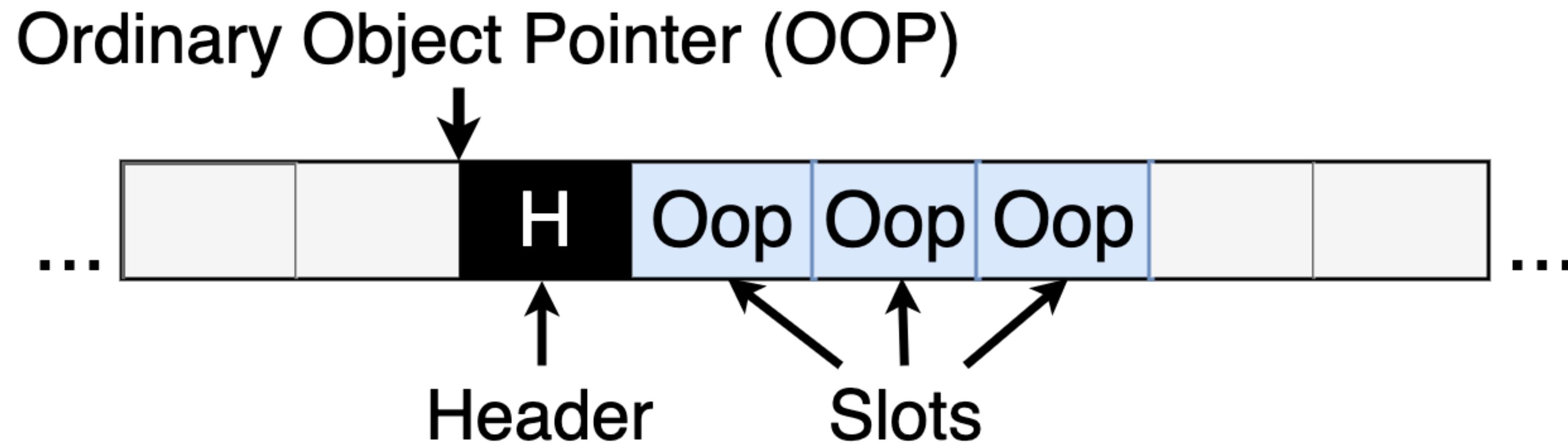
## ... With the Help of the VM Simulator

```
\200^U^@^@^@^C LanguageEnvironment^@^@^@^@^@^M^L\200^W^@^@^@^C ShortIntegerAr  
ray^@^@^@^@^@^@^@^M^L\200^W\356.^BDiskStore^@^@^@^@^@^@^M^L\200^U\207^@'  
^BMemoryStore^@^@^@^@^@^@^M^L\200^W^@^@^@^BClipboard^@^@^@^@^@^@^M^L\200^Vf  
\234^K^CMCMethodDefinition^@^@^@^@^@^@^M^L\200^R^@^@^@^A Locale^@^@^M^L\200  
Pw\310&^AASTCache^M^L\200^S^@^@^@^BOSEnvironment^@^@^@^@^M^L\200^S\323\2711^C  
InternetConfiguration^@^@^@^M^L\200^P^0k^D^AZnServer^M^L\200^VWB^S^CMCGitHu  
bRepository^@^@^@^@^@^@^M^L\200^U\301] (^DMCGitBasedNetworkRepository^@^@^@^  
@^@^M^L\200^VQa^M^BZnLogEvent^@^@^@^@^@^M^L\200^S^@^@^@^BDisplayScreen^@^  
@^@^M^L\200^R^@^@^@^ACursor^@^@^M^L\200^T^@^@^@^AForm^@^@^@^M^L\200^V^@^@  
^@^BStrikeFont^@^@^@^@^@^M^L\200^S^@^@^@^BFreeTypeCache^@^@^@^M^L\200^U^@  
^@^@^BLogicalFont^@^@^@^@^@^M^L\200^P^@^@^@^BFreeTypeSettings^M^L\200^V^@^@  
^@^BWorldMorph^@^@^@^@^@^M^L\200^V^@^@^@^BCPUWatcher^@^@^@^@^@^@^M^L\200^  
P^@^@^@^BPharoCommonTools^M^L\200^V^@^@^@^BGTPlayBook^@^@^@^@^@^@^M^L\200^W  
^@^@^@^DSystemSettingsPersistence^@^@^@^@^@^@^M^L^@^Q^@^@^@^A Default^@a^L  
^@^A^@^@^@^Gh\224\250^0^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@  
\300\242\371^N^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^H\243\37  
1^N^@^@^@^@^M^L^@^S^@^@^@^B priorityLists^@^@^@^@^S^L^@^A^@^@^@^B\200^0\345^N^  
@^@^@^@^@^220t\346^N^@^@^@^@^@^M^L^@^U^@^@^@^C registeredClassName^@^@^@^@^@^S^L  
^@^A^@^@^@^B\200^0\345^N^@^@^@^@^@^8u\346^N^@^@^@^@^@^S^L^@^A^@^@^@^B\200^0\345^  
N^@^@^@^@^@^350u\346^N^@^@^@^@^@^S^L^@^A^@^@^@^B\200^0\345^N^@^@^@^@^@^@^220v\346^N  
^@^@^@^@^@^S^L^@^A^@^@^@^B\200^0\345^N^@^@^@^@^@^350w\346^N^@^@^@^@^M^L^@^T;\22  
6"^\B pushInstVar:^@^@^@^@^M^L^@^S\313\372^G^B storeInstVar:^@^@^@^S^L^@^A^@^@  
^@^B\200^0\345^N^@^@^@^@^@^220x\346^N^@^@^@^@^M^L^@^V^@^@^@^C immediateSubclas  
S:^@^@^@^@^@^@^S^L^@^A^@^@^@^B\200^0\345^N^@^@^@^@^@^8y\346^N^@^@^@^@^@^S^L^@^A^  
^@^@^@^B\200^0\345^N^@^@^@^@^@^210z\346^N^@^@^@^@^@^M^L^@^S^@^@^@^B signalContext  
^@^@^@^@^M^L^@^R^@^@^@^B handlerContext^@^@^@^S^L^@^A^@^@^@^B\200^0\345^N^@^@^@^@^  
@0{\346^N^@^@^@^@^S^L^@^A^@^@^B\200^0\345^N^@^@^@^@^330{\346^N^@^@^@^@^M^
```

Is this it ?

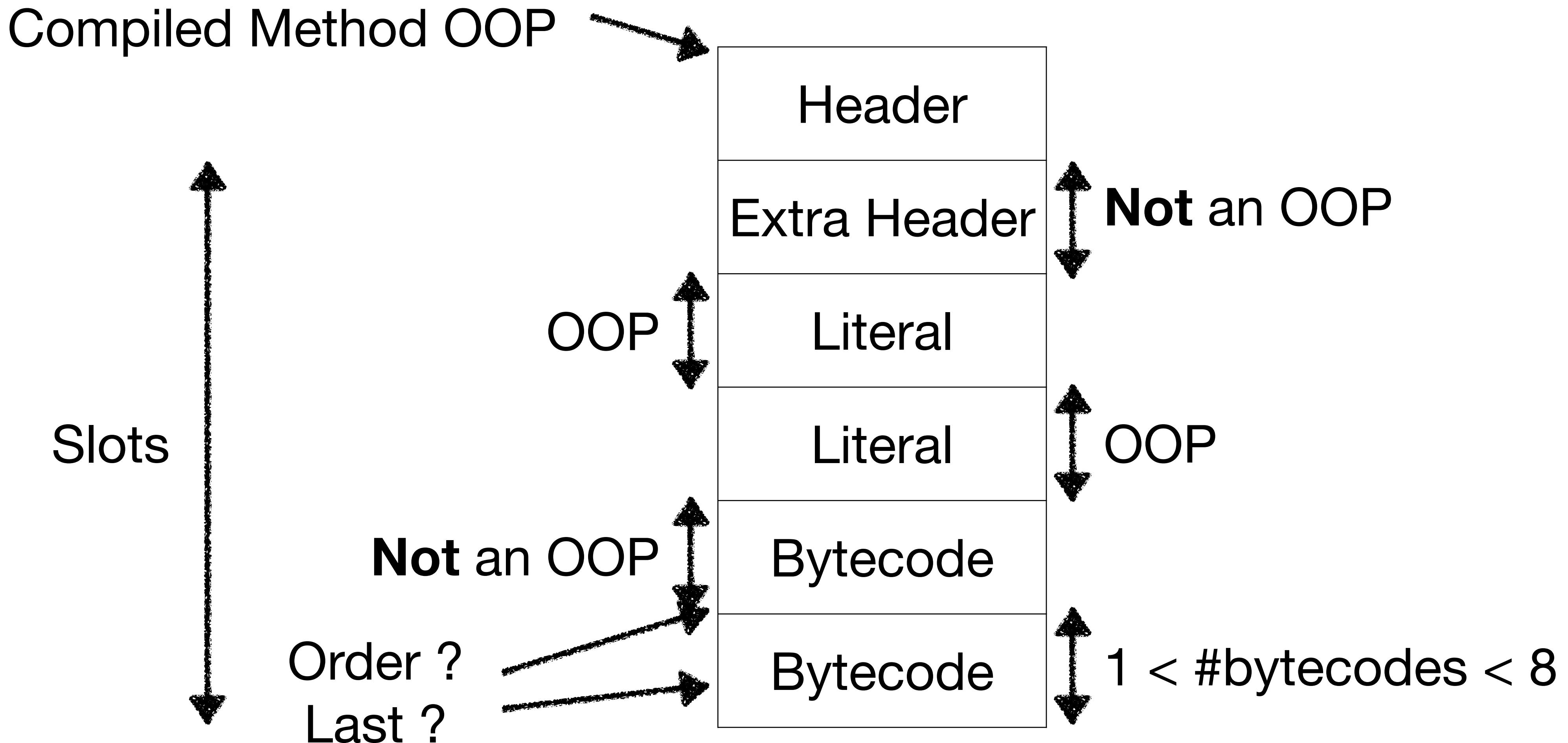
# Let's Code VM Level Tools !

## Ordinary Object Pointer (OOP)



# Let's Code VM Level Tools !

## Method Layouts



# Let's Code VM Level Tools !

## Finding a Class at the VM Level

```
findClassName: aClassName
| classNameIndex classNameOop className |
memory classTableEntriesDo: [ :aClassOop |
    classNameIndex := memory classIndexForOop: aClassOop.
    classNameOop := memory fetchPointer: classNameIndex ofObject: aClassOop.
    className := memory convertStringOopToStringObject: classNameOop.
    className = aClassName ifTrue: [ ^ aClassOop ]].
^ memory nilOOP
```

```
memory findClassName: #Form >>> 406749864
```

# Let's Code VM Level Tools !

## Knowledge Gaps

**findClassName:** aClassName

| classNameIndex classNameOop className |

memory classTableEntriesDo: [ :aClassOop |

    classNameIndex := memory classNameIndexForOop: aClassOop.

    classNameOop := memory fetchPointer: classNameIndex ofObject: aClassOop.

    className := memory convertStringOopToStringObject: classNameOop.

    className = aClassName ifTrue: [ ^ aClassOop ].

^ memory nilOOP

**VM level hidden OOP**

**OOP**

**Low level style**

**OOP based API**

# Let's Code VM Level Tools !

## Knowledge Gaps recaps

### Issues

- Ordinary Object Pointers (OOP)
- API manipulating OOPs
- VM level information

# Polyphemus

## Introducing LLOOPs

### Language level OOPs

#### Issues

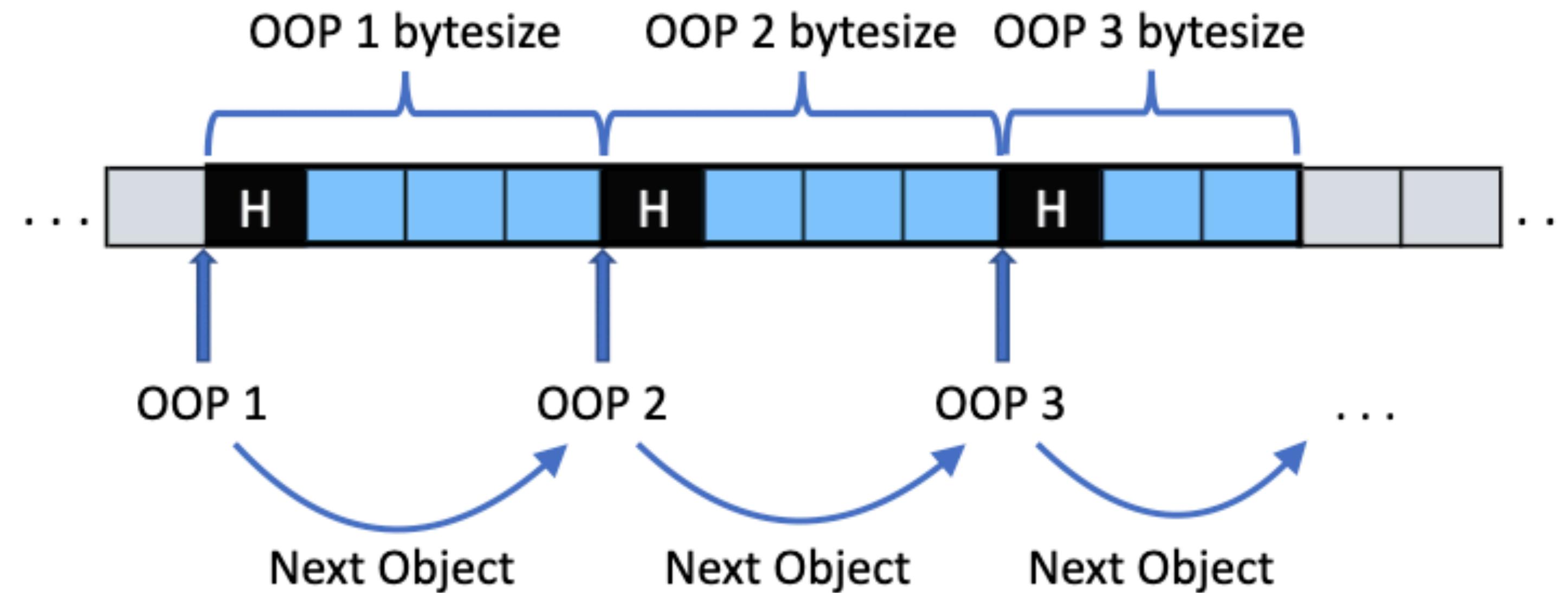
- Ordinary Object Pointers (OOP)
- API manipulating OOPs
- VM level information

#### Solutions

- Language Level Entities
- Identifying and Typing OOP
- VM and Language level information

# Polyphemus

## Iterating the Memory



# **Polyphemus**

## **Identifying the Each OOP**

OOP	OOP	OOP	OOP
-----	-----	-----	-----

**Is it a Class ?**

**Is it a Metaclass ?**

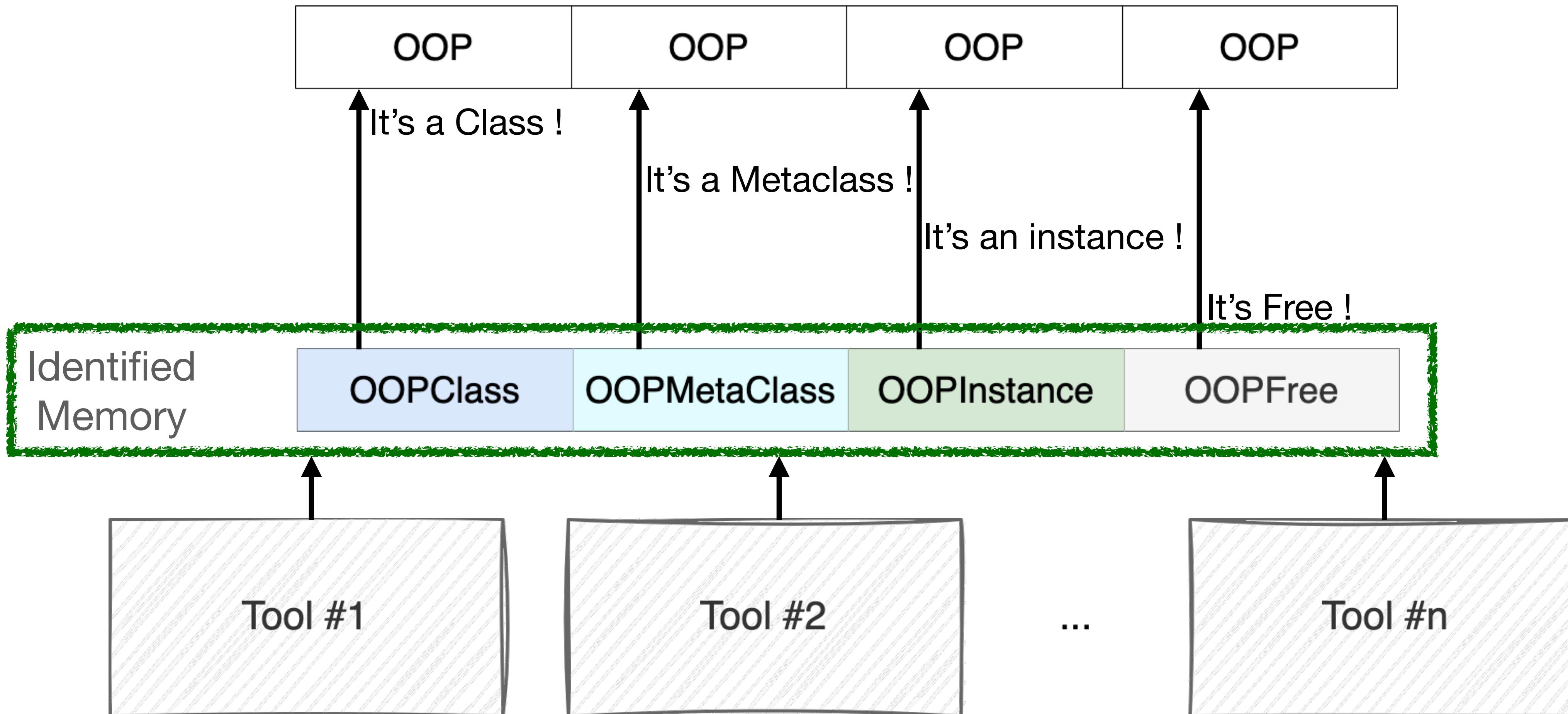
**Is it a Free Object ?**

**Is it a [...] ?**

**If no particularity, it's an instance !**

# Polyphemus

## LLOOP based tools



# Polyphe<sup>m</sup>us Tools

## Writing Tools

```
identifiedMemory allClassesOop.
```

```
identifiedMemory reifiedMetaclass.
```

```
identifiedMemory allClassesOop select: [ :o | o isClassSide ].
```

```
identifiedMemory objects
```

```
    select: [ :o | o isCompiledMethodOop ]
```

```
    thenCollect:[ :aCompiledMethod |
```

```
        [ aCompiledMethod decompile ] on: Error do: [ nil ].
```

# Polyphemus Tools

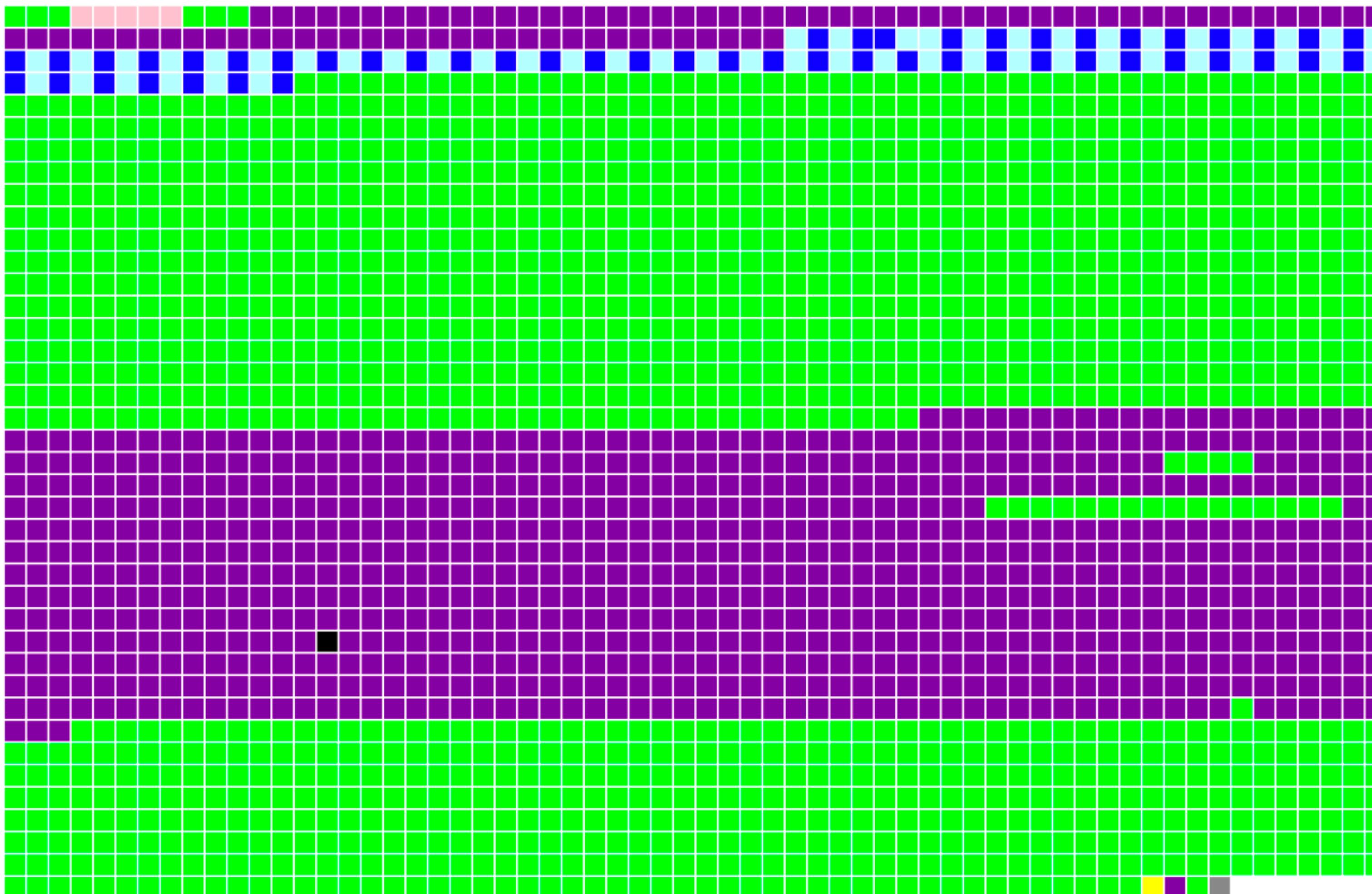
## Inspectors

# Compiled Method

address	4676295184
printString	Form >> #scaledByDisplayScaleFactor
header	1010000000000000000000000000000011000000000000000110000011101
class	CompiledMethod
oopClassTag	3101
format	Compiled method (24)
hash	0
pinned	false
space	Old Space
immutable	false
selector	scaledByDisplayScaleFactor
methodClass	Form
numLiterals	7
numBytecodes	16
bytecode	an Array [16 items] (76 128 216 76 76 129 76 130 131 104 148 92 118 1 0 253)
literal 1	halt
literal 2	extent
literal 3	currentWorld
literal 4	displayScaleFactor
literal 5	scaledToSize:
literal 6	Instance of AdditionalMethodState
literal 7	Instance of GlobalVariable

# Polyphemus Tools

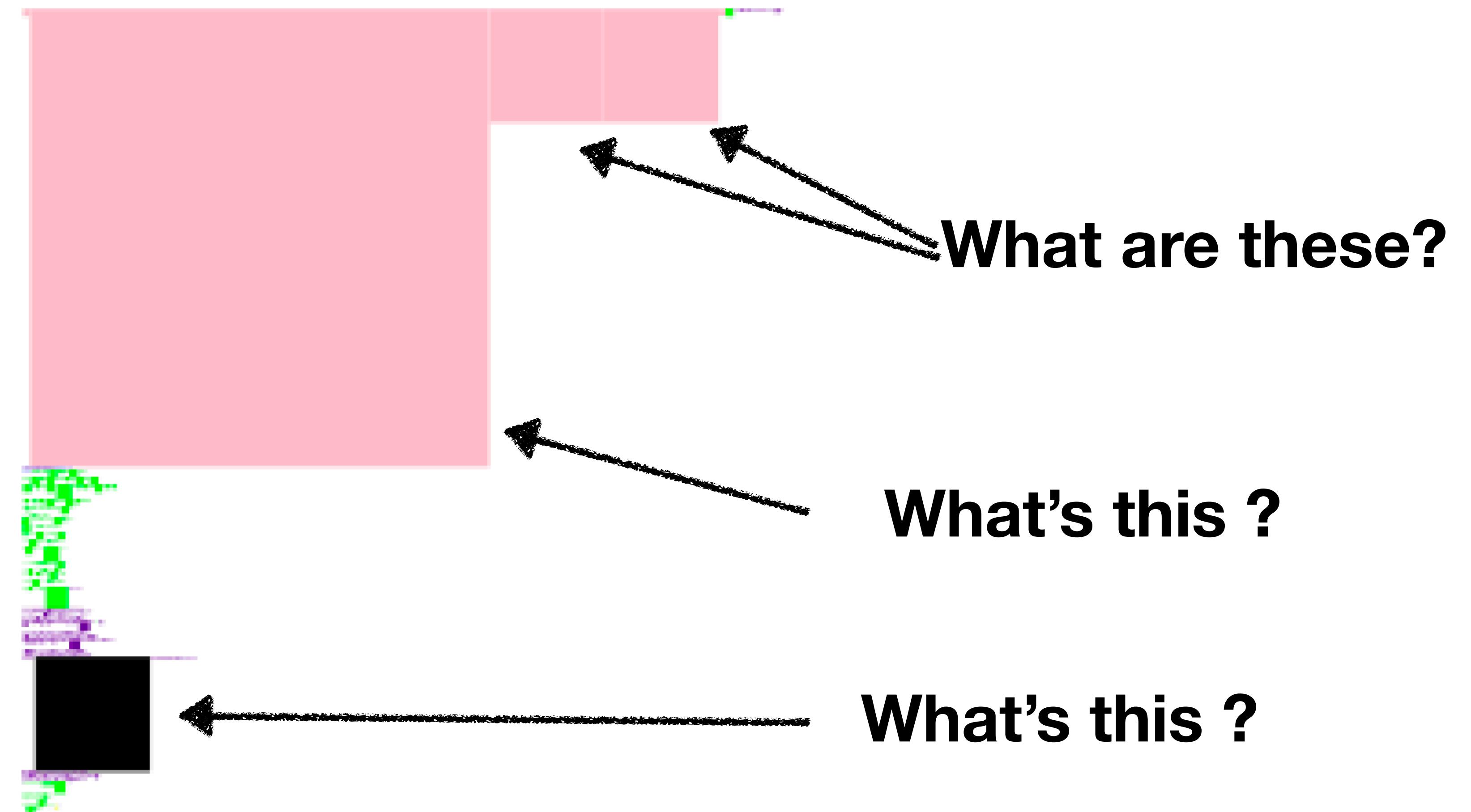
## Memory visualisation



- 1 pinned object
- 895 compiled method
- 51 class
- 5 special object
- 1 context
- 1 free chunk
- 1468 regular object
- 51 metaclass

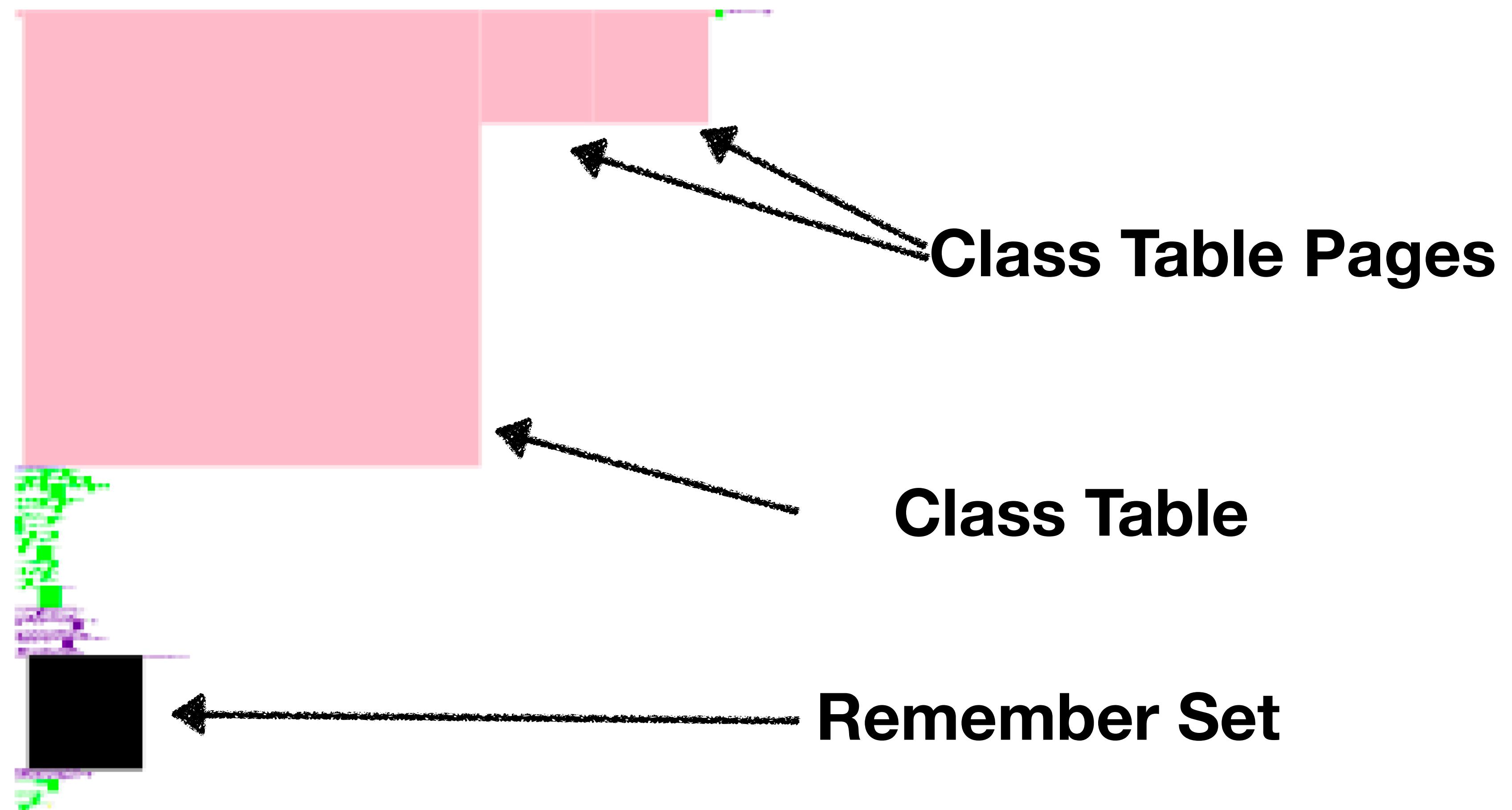
# Polyphemus Tools

## Evaluating Memory Size



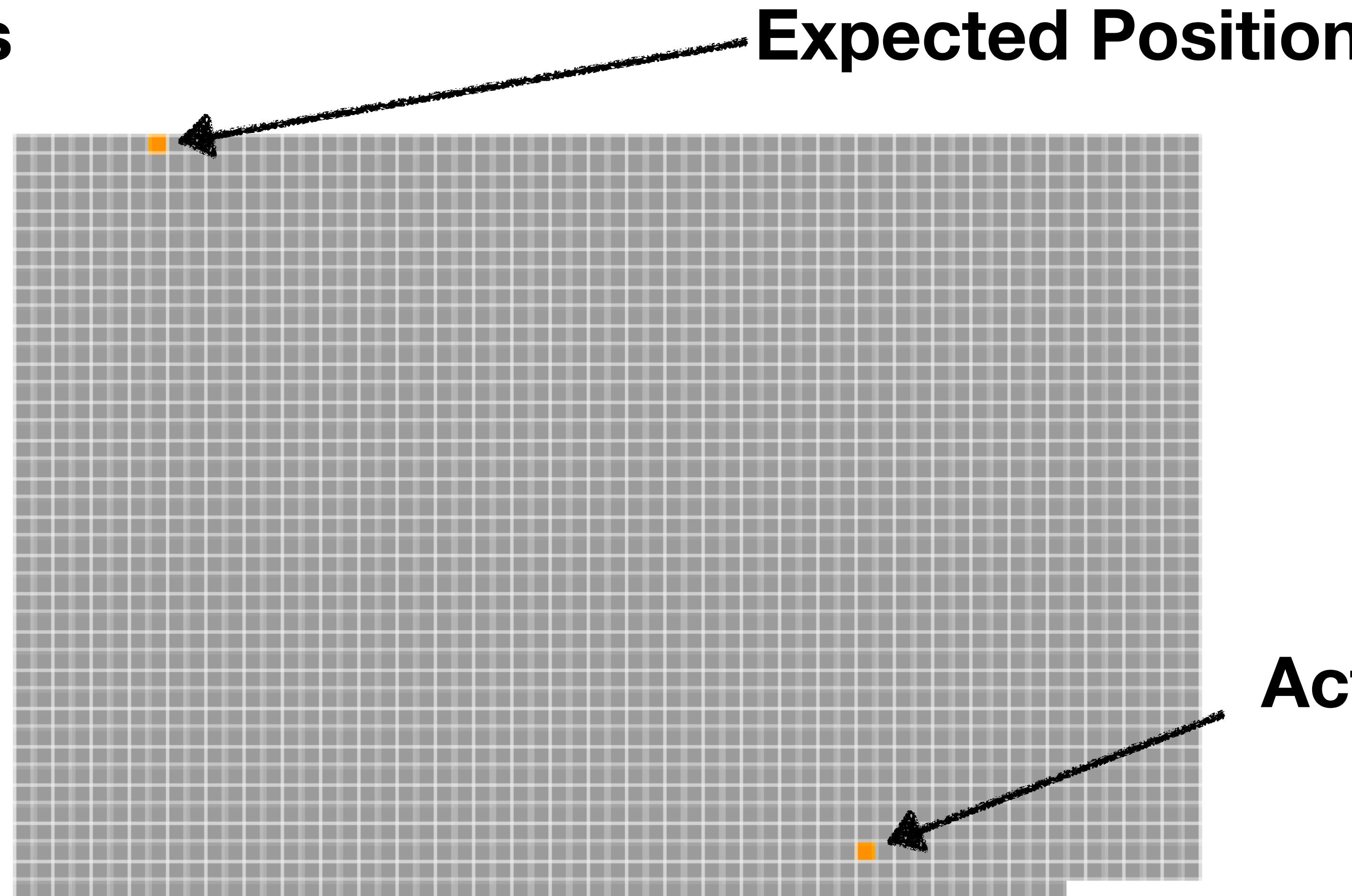
# Polyphemus Tools

## Evaluating Memory Size

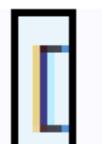


# Polyphemus Tools

## Named Objects



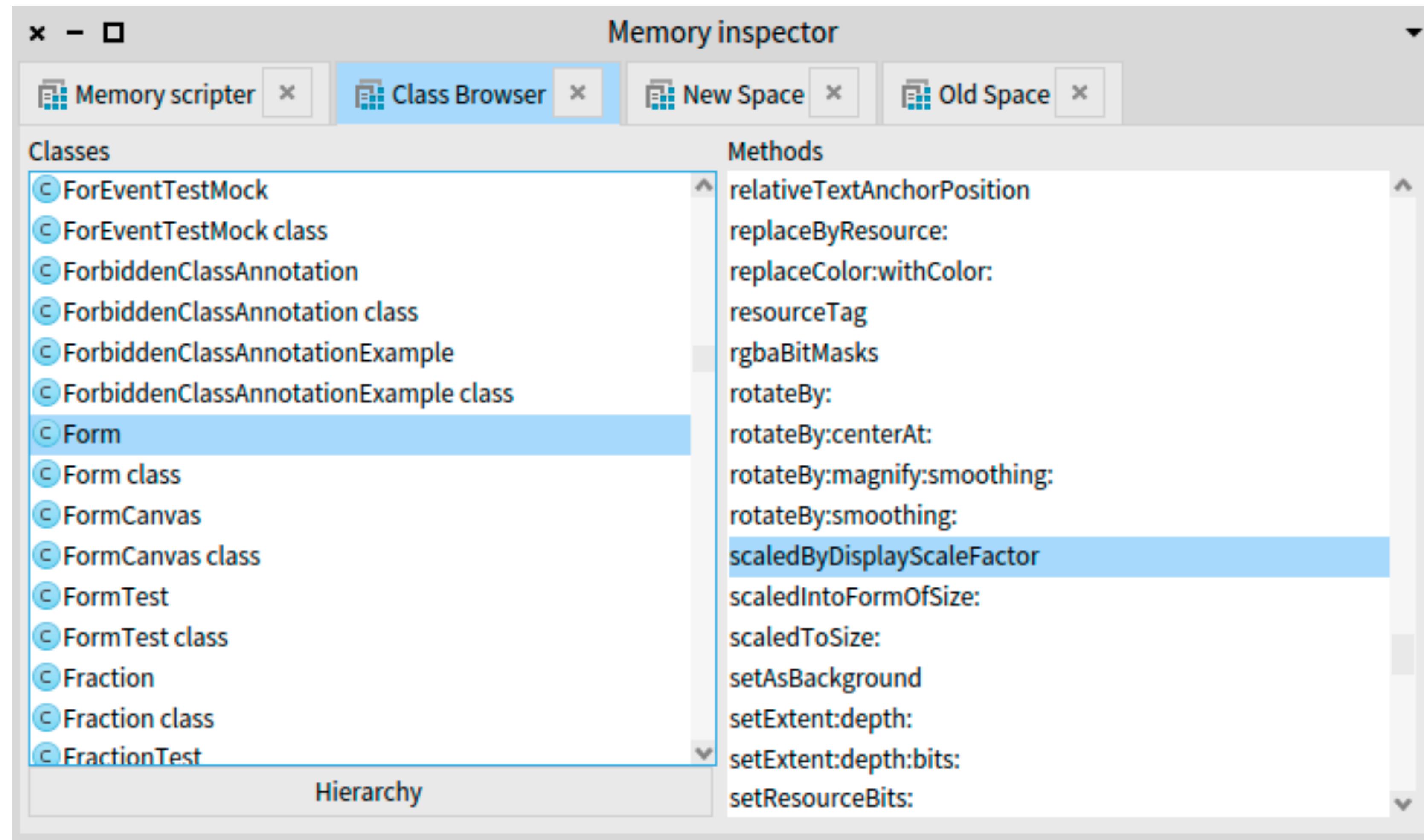
Selecting



:anOop | anOop numSlots = 60 ]

# Polyphemus Tools

## Memory Visualisation #2



# Real World Bug Fix #1

## A Meta-Error

**Form >> #scaledByDisplayScaleFactor**

**self halt.**

**^ self scaledToSize: self extent \* self currentWorld displayScaleFactor.**



# Real World Bug Fix #1

## Investigating

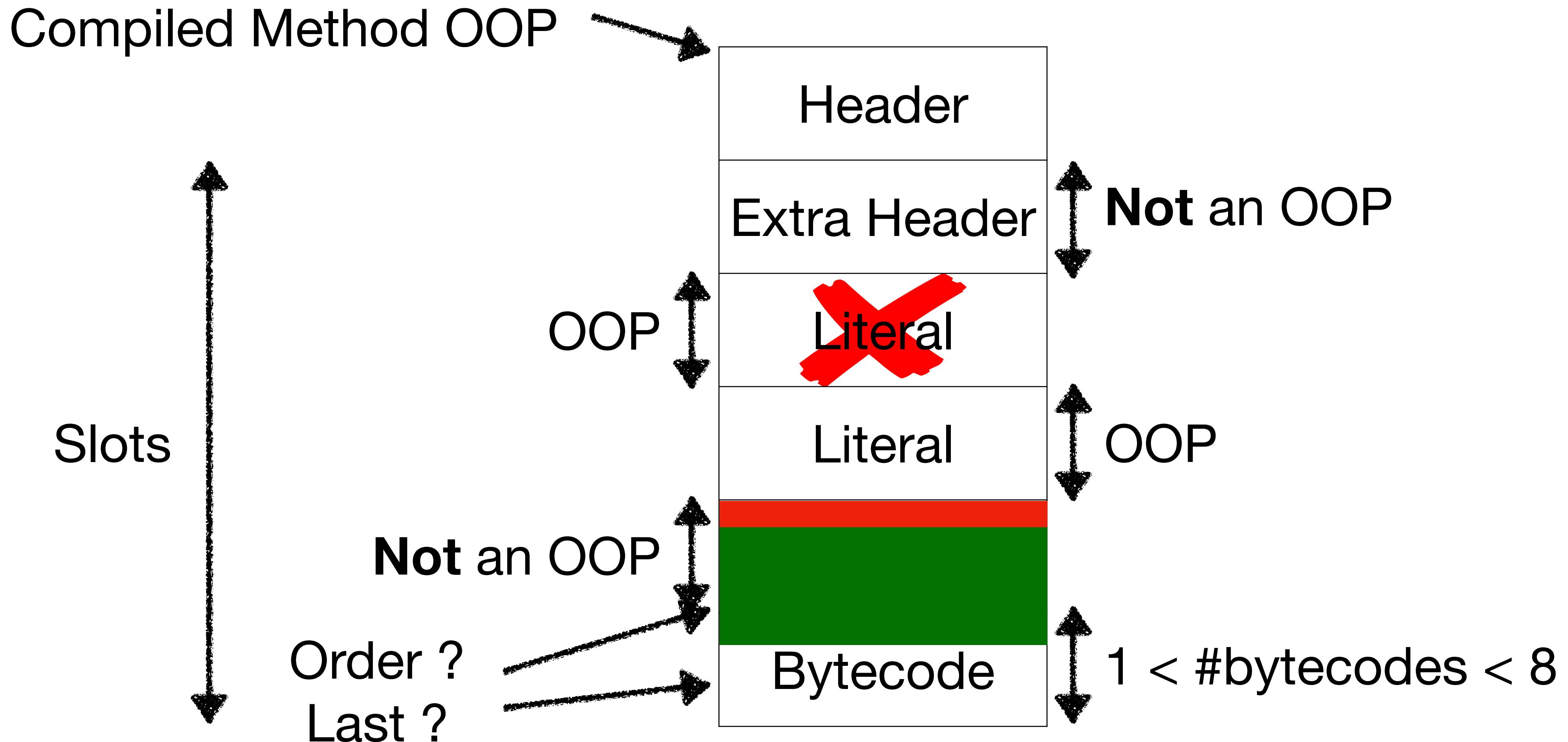
→ bytecode an Array [16 items] (76 128 216 76 76 129 76 130 131 104 148 92 118 1 0 253)  
bytecode an Array [13 items] (76 76 128 76 129 130 104 147 92 50 42 158 252)

literal 1	halt
literal 2	extent
literal 3	currentWorld
literal 4	displayScaleFactor
literal 5	scaledToSize:
literal 6	scaledByDisplayScaleFactor
literal 7	Instance of GlobalVariable

literal 1	extent
literal 2	currentWorld
literal 3	displayScaleFactor
literal 4	scaledToSize:
literal 5	scaledByDisplayScaleFactor
literal 6	Instance of GlobalVariable

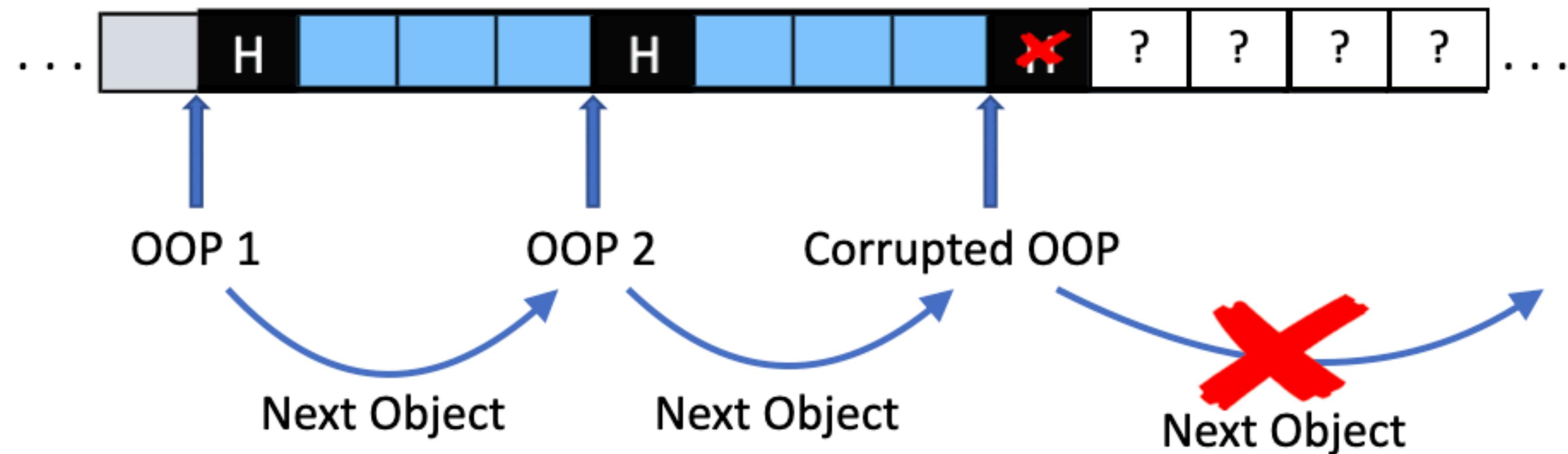
# Real World Bug Fix #1

## Method Patching Analysis



# Real World Bug Fix #2

## Iterating the Corrupted Memory



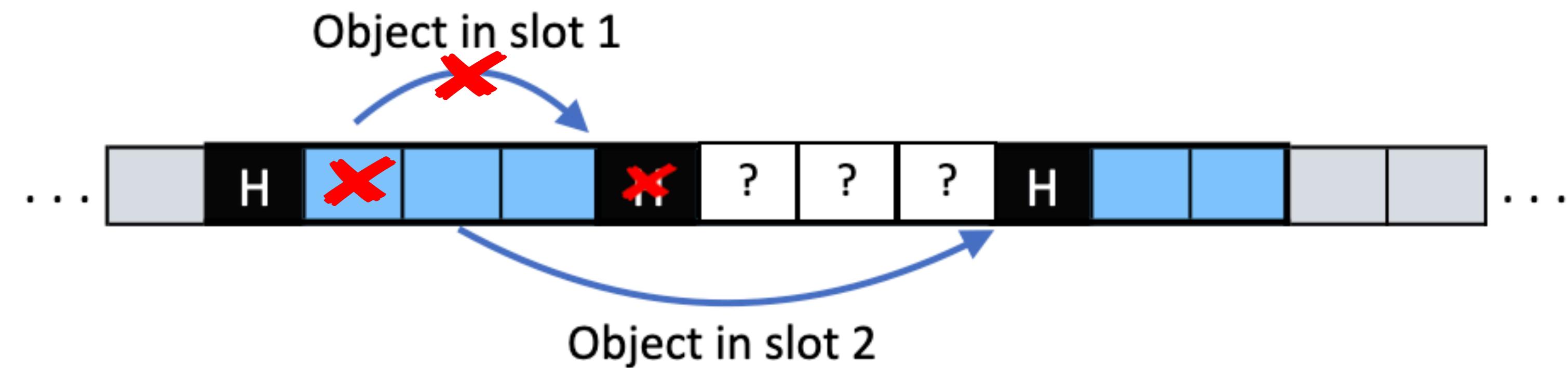
# Real World Bug Fix #2

We don't know what's after

Oop Oop Oop Oop ?

# Real World Bug Fix #2

## Recovering Objects



# Real World Bug Fix #2

## Tracking The Corruption

Oop	Oop	Oop	Oop	Oop	Oop	C	Oop	Oop	Oop	Oop
Oop	Oop	Oop	F	Oop						
Oop	Oop	C	Oop	Oop	Oop	F	Oop	Oop	Oop	Oop
Oop	F	Oop	F	Oop	C	Oop	Oop	Oop	Oop	Oop
Oop	Oop	F	Oop	Oop	Oop	Oop	F	Oop	Oop	Oop

# Real World Bug Fix #2

## Cleansing The Corruption

Oop	Oop	Oop	Oop	Oop	Oop	F	Oop	Oop	Oop	Oop	Oop
Oop	Oop	Oop		F	Oop						
Oop	Oop		F	Oop	Oop	Oop		F	Oop	Oop	Oop
Oop		F	Oop	F	Oop		F	Oop	Oop	Oop	Oop
Oop	Oop		F	Oop	Oop	Oop	Oop	Oop	F	Oop	Oop

# Real World Bug Fix #2

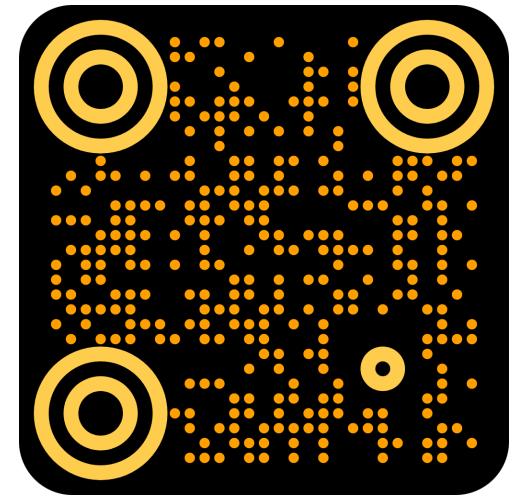
## Corruption Cleansing Analysis

- Objects' slots iteration
- Reference patching
- Focus on learning rather than how to do the previous items

# Conclusion

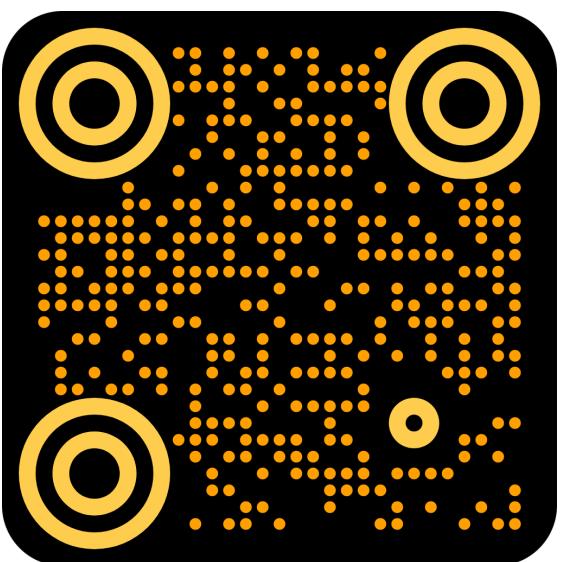


# Github

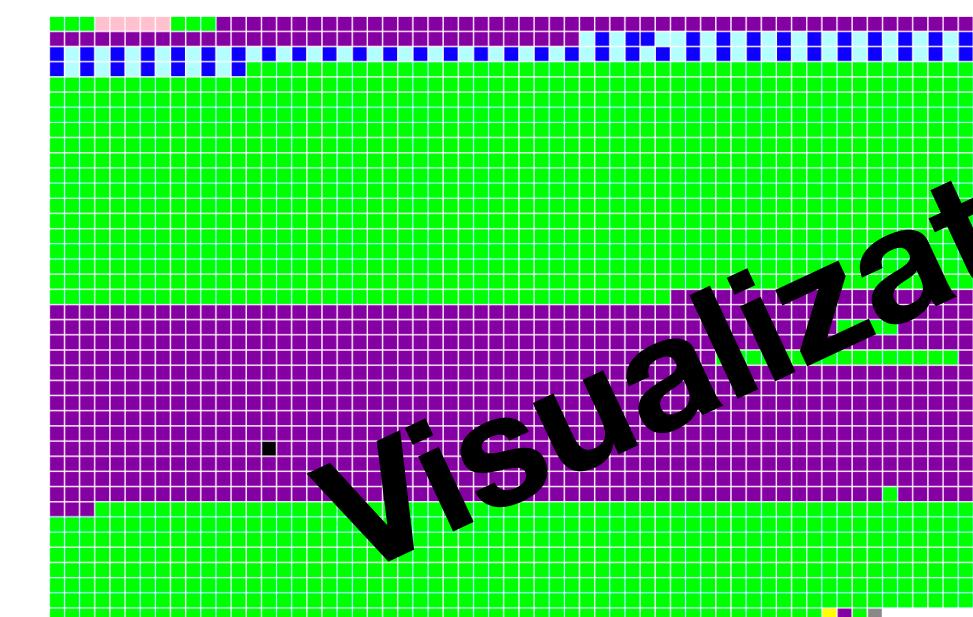


- Tooling at the VM level is difficult
  - LLOOPs bridge knowledge gaps
    - Language Level entities
    - Identifying and typing OOPs
    - Mix Language and VM level informations

# Paper



Pierre Misce-Chanabier  
pierre misse25@msn.com  
github.com/hogoww  
Discord tag: hogo#8547



+ Key	+ Value
address	406749864
printString	Form
header	101100000000000000001110011001000000001000000000000000000110110001
class	Form class
oopClassTag	1841
format	Non Indexable (1)
hash	1842
pinned	false
space	Old Space
immutable	false
numSlots	11
superclass	DisplayMedium
methodDict	Instance of MethodDictionary
format	65541
layout	Instance of GridLayout
organization	Instance of ClassOrganization
subclasses	Instance of Array
name	Form
classPool	Instance of Dictionary
sharedPool	nilObject
environment	Instance of SystemDictionary
category	Graphics-Display Objects-Forms