

SMA:
Software Modeling and Analysis

Practical Session

Week 04

Assignment 03

Discussion

A03 - Exercise 01

Metamodels (2.5 pts)

i) What is a metamodel?

A model of a model; a prescriptive view on an existing model. It determines the syntax and semantics of models that conform to it.

ii) How are metamodels used in Pharo?

Every object is an instance of a class. Every class inherits from `Object`. Every class is an instance of its (unique) metaclass, which inherits from `Class`. Every metaclass is an instance of `Metaclass`, which is itself a class.

iii) What are responsibilities of a metaclass in Pharo?

Instance creation, creating initialized instances of the metaclass's sole instance, initialization of classvariables, method compilation, ...

A03 - Exercise 01

Metamodels (2.5 pts)

- iv) Where is `ProtoObject` located in Pharo's class hierarchy?
`ProtoObject` is the root class for all other classes including `Object`.
`ProtoObject` is the superclass of `Object`.

- v) What is the purpose of the class `ProtoObject`?
The class `ProtoObject` only contains the core behavior needed to make the system work. The idea of `ProtoObject` is to have a lean class that separates the concerns.

A03 - Exercise 02

Sub and super classes (3 pts)

(you have to provide your code snippet and the result)

i) How many superclasses does `Collection` have?

2: `Collection allSuperclasses size.`

ii) How many direct subclasses does `Collection` have?

32: `Collection subclasses size.`

iii) How many indirect subclasses does `Collection` have?

129: `Collection allSubclasses size -
Collection subclasses size.`

A03 - Exercise 03

Class identity (3 pts)

a) Who new amIClassy.

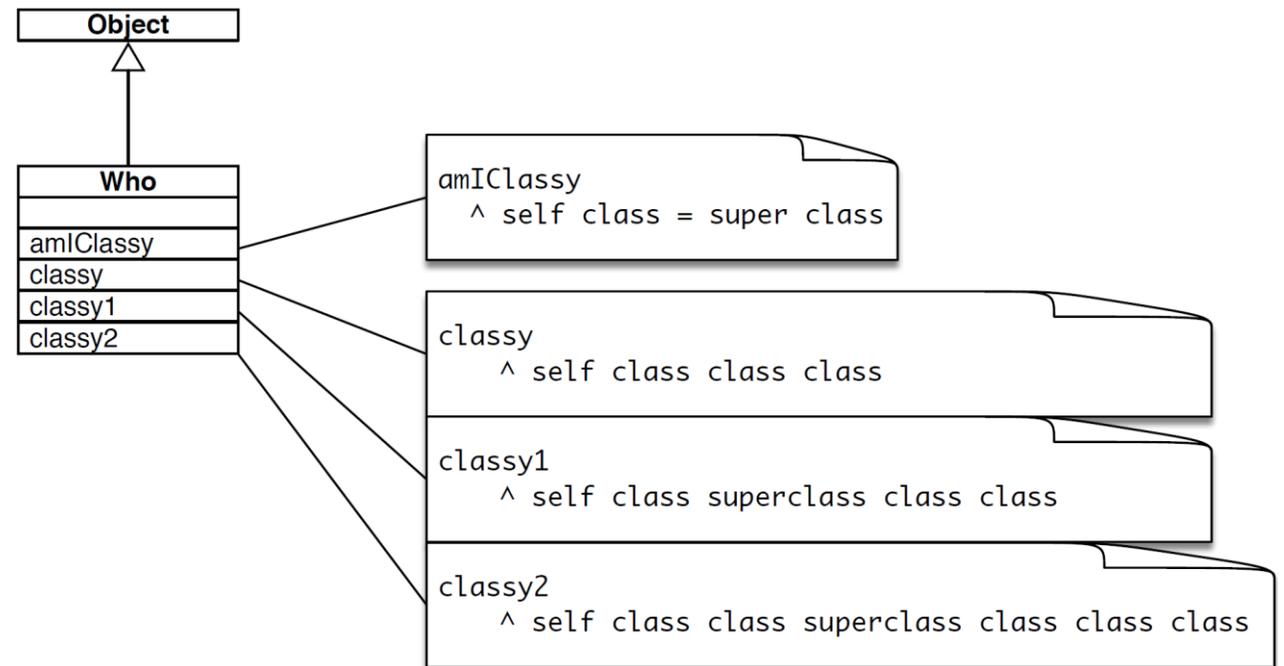
True: Super is used in the context of the class of the method implementation.

b) Who new classy = Who new classy1.

True: Both elements represent the same object.

c) Who new classy1 = Who new classy2.

True: Both elements represent the same object.



A03 - Exercise 04

Object instantiation (1.5 pts)

- i) Where is `new` defined?
It is first defined in the class `Behavior`.
- ii) Explain Pharo's message implementation resolution strategy for the `new` message.
*When the `new` is sent to a class it is being resolved throughout its metaclass chain.
The search ultimately ends in `Behavior`.*
- iii) List the concrete code in GT finally executed by the message `new`.
`self basicNew initialize` in the class `Behaviour`.

Assignment 04

Preview

A04 - Exercise 01 | Hierarchy traversal

Write a *method*.

Find the *longest inheritance chain* among all Smalltalk classes in the Pharo programming environment.

NB:

To access all classes of Smalltalk, you can use

```
SystemNavigation default allClasses
```

A04 - Exercise 02 | Method overrides

Write a ***method***.

Find all ***abstract method overrides*** in the Pharo system.

A04 - Exercise 03 | Query methods

Write a *method*.

Find all *query method implementing classes*.

NB:

Query methods test a property of an object. Such methods are prefixed with "is", "was" or "will".

A04 - Exercise 04 | Root methods

Write a ***method***.

i) Find all ***root methods*** in GT.

NB:

A "root method" is a method whose selector has been implemented in a class, such that the super classes of that class do not understand it.

ii) (BONUS) Find all ***duck-typed methods*** in GT.

Duck-typed methods have the same selector but are not related by inheritance. That is, after finding all root methods, find those with the same selector.

A04 - Exercise 05 | Dynamic coding

Dynamic extension of code.

Step 1:

Redefine `Call>>doesNotUnderstand: aMessage.`

-> Add dynamically an instance variable to the class `Call`.

Step 2:

Add dynamically the provided method to the class `Call`.

Step 3:

Resend the initial message to `self`.