Compare Object Orientated Programming in Go with Smalltalk

DANIEL VOSS

Smalltalk

- Created during 1970s
- Pure Object Oriantated
- Dynamically typed
- Usually everything (IDE, language …) implemented in VM like Squeak, Pharo …

Object Orientation Concept

- Everything is an object.
- Every object is an instance of a class.
- Every class has a superclass.
- Everything happens by sending messages.

Messages

No “function calls“ only messages with arguments between objects

Everything happens by sending messages (If-Else, Loops, Class creation …).

- unary message:
  5 sqrt
- binary messages:
  *3 * 4*
- keyword messages:
  10 between: 5 and: 15

Class and Inheritage

```
Object subclass: #Person
        instanceVariableNames: 'surname age'
        classVariableNames: 'count'
        package: 'Peoples'
Person>> age
        ^ age
Person>> age: anObject
        age:= anObject
```

```
Person subclass: #Teacher
        instanceVariableNames: 'subject'
        classVariableNames: ''
        package: 'Peoples'
Teacher>>age
        ^(age - 1)
```

Metaclass

- classes are object, instance of metaclass
- message *subclass:* creates hidden metaclass and a unique instance of it
- new message to that unique instance creates object as expected
- class side view (behaviour of instance of metaclass) can be changed:

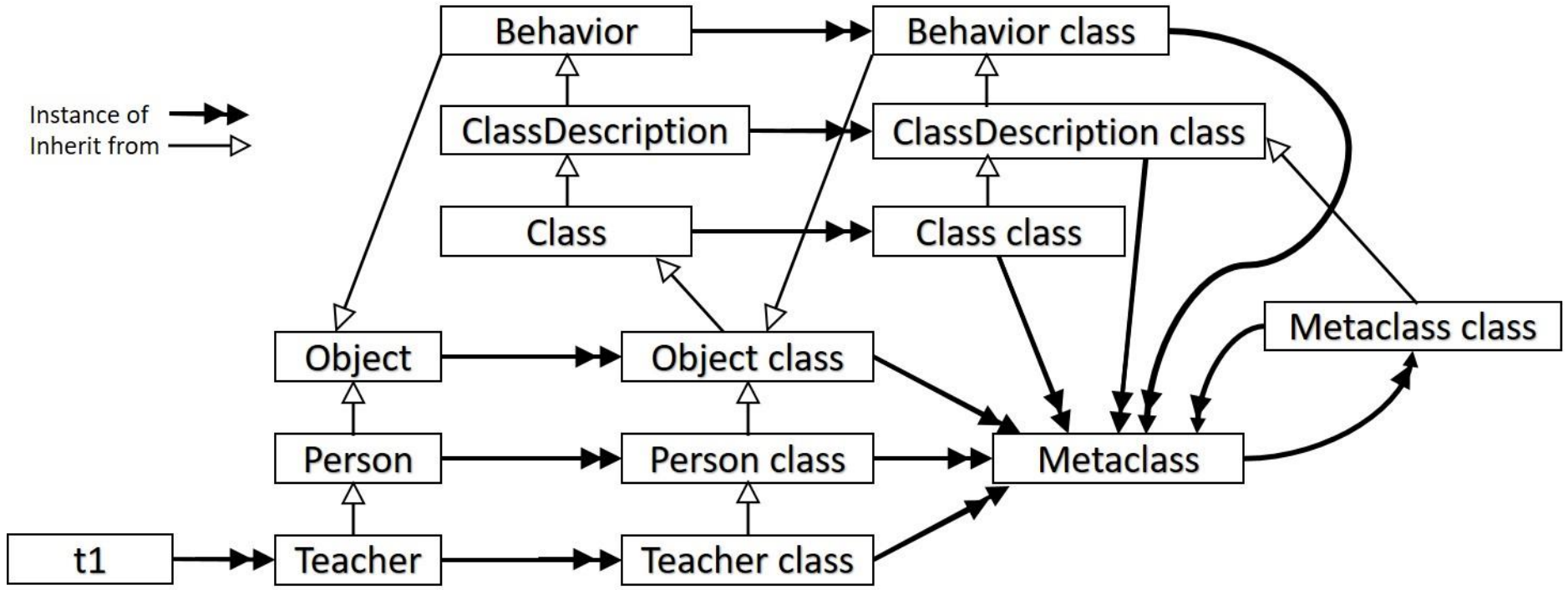
```
Person class>>initialize
        count := 0.
Person class>>new
count := count +1.
        ^super new.
```

```
Person class>>new: aAge
        | person |
        person := self new.
        person age: aAge.
        ^person
```

Metaclass instance

Everything is an object -> Metaclass is an instance of Metaclass class

```
t1 := Teacher new.
```



Literature

- https://pharo.org/
- https://squeak.org/
- Pharo by Example. Andrew P. Black, Stephane Ducasse, Oscar Nierstrasz and Damien Pollet. 2011 http://pharo.gforge.inria.fr/PBE1/PBE1.html
- syntax-across-languages 2008 http://rigaux.org/language-study/syntax-across-languagesper-language/Smalltalk.html