

Classes and Metaclasses in Smalltalk

Michael McThrow

San Jose State University
Computer Science Department
CS 152 – Programming Paradigms

November 18, 2020



Table of Contents

- 1 Classes in Smalltalk
- 2 Metaclasses in Smalltalk
- 3 Demo of Squeak
- 4 Question-and-Answer Session

Table of Contents

- 1 **Classes in Smalltalk**
- 2 Metaclasses in Smalltalk
- 3 Demo of Squeak
- 4 Question-and-Answer Session

Classes in Smalltalk

Believe it or not, Smalltalk does not have class files like Java. How classes and their associated methods are laid out in Smalltalk is an implementation detail.

We will cover how Squeak defines classes.

Defining Classes Using Squeak

```
Object subclass: #NameOfSubclass
  instanceVariableNames: ''
  classVariableNames: ''
  poolDictionaries: ''
  category: ''
```

- In the above code, the class `NameOfSubclass` inherits from the base class `Object`.
- `instanceVariableNames`, `classVariableNames`, and `poolDictionaries` are strings containing names, where each name is delimited by a space
- `category` is a string that describes the category the class belongs in.

Difference between Instance Variables and Class Variables

- An instance variable is associated with an object.
- A class variables is shared among all instances of a class.
 - For example, suppose the class `GUIPoint` had a class variable called `Window`, which refers to the window the GUI point is drawn on. All `GUIPoint` objects in the system have access to the same `Window` object.

Details about Inheritance in Smalltalk

- All objects inherit from a superclass.
- The base class in Smalltalk is `Object`. All objects ultimately derive from `Object`.
- Unlike C++ and Common Lisp, Smalltalk does not support multiple inheritance; i.e., a class may only inherit one superclass in Smalltalk.

Table of Contents

- 1 Classes in Smalltalk
- 2 Metaclasses in Smalltalk
- 3 Demo of Squeak
- 4 Question-and-Answer Session

- Everything is an object in Smalltalk.

- Everything is an object in Smalltalk.
- All objects are instances of classes.

- Everything is an object in Smalltalk.
- All objects are instances of classes.
- Because **everything** is an object, that means that classes are objects, too.

- Everything is an object in Smalltalk.
- All objects are instances of classes.
- Because **everything** is an object, that means that classes are objects, too.
- Because all objects are instances of classes, that means classes are themselves instances of classes.

**EVERYTHING
IS AN OBJECT**



**ALL OBJECTS
ARE INSTANCES
OF CLASSES**



**CLASSES
ARE OBJECTS**



**CLASSES ARE
THEMSELVES INSTANCES
OF METACLASSES**



imgflip.com

Metaclasses

- According to the textbook, “a class whose instances are themselves classes is called a **metaclass**” [Goldberg and Robson, p. 76].
- “Each class is an instance of its own metaclass.” [p. 77].
 - “Whenever a new class is created, a new metaclass is created for it automatically” [p. 77].
- Metaclasses are instances of the class `Metaclass`.
- Metaclasses don't have their own names; you access a metaclass by sending the `class` message to the class (e.g., `Rectangle class`).

Class Methods and Instance Methods

- Class methods are equivalent to Java's `static` methods. They are associated with the metaclass.
- Instance methods are associated with instances of the class. When defining them, they are defined for the class, not the metaclass.
- There are no dedicated constructors in Smalltalk. Rather, you could create one or more class methods that call the `new` method, which is inherited from the `Object` base class.

Additional Details about Metaclasses

- If a class is inherited from a superclass, then the class's metaclass is inherited from the superclass's metaclass.
- “Class variables are accessible to both the class and its metaclass” [Goldberg and Robson, p. 84].

Table of Contents

- 1 Classes in Smalltalk
- 2 Metaclasses in Smalltalk
- 3 Demo of Squeak**
- 4 Question-and-Answer Session

Table of Contents

- 1 Classes in Smalltalk
- 2 Metaclasses in Smalltalk
- 3 Demo of Squeak
- 4 Question-and-Answer Session**