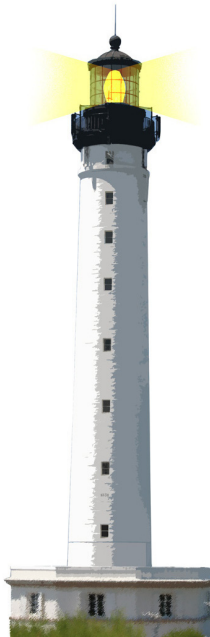
Learning Object-Oriented Programming and Design with TDD

Inheritance Basics

Stéphane Ducasse
http://stephane.ducasse.free.fr



Pharo
http://www.pharo.org

W5S01

Goal

- What is inheritance?
- When to use it?

Inheritance In a Nutshell

- Remember: A class defines the behavior of all its instances
- With inheritance: A class can extend/reuse/refine the state/behavior of another class
- When a class inherits from a another one (superclass):
  - its instances can execute methods of the superclass
  - its instances have the state of the superclass in addition to the ones of the superclass

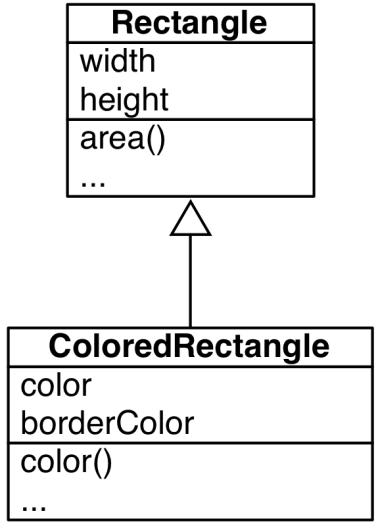
Pharo, Java and others

Pharo treats inheritance the same way as Java/C#:

- Single inheritance between classes
- Static inheritnance instance variables (at compile-time)
- Dynamic for methods (at runtime)

The Basics

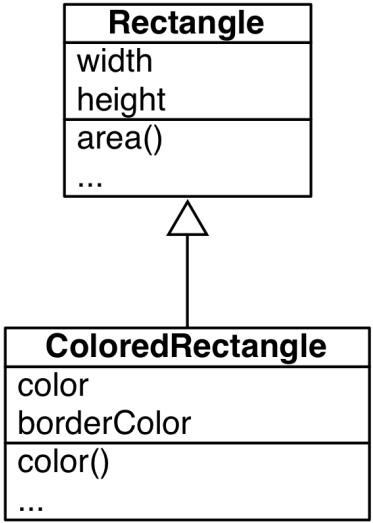
- Often we want small adaptations
- We want to extend existing behavior and state
- Solution: **class inheritance**
- A class extends the definition of its superclass with new state and/or behavior



Possible Actions while Inheriting

A subclass

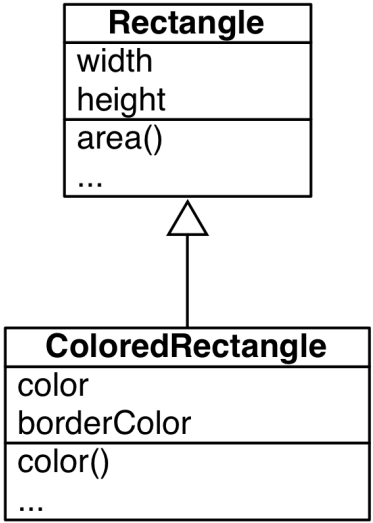
- can add specific state and behavior:
  - color, borderColor, ...
- can use superclass behavior and state
- can specialize and redefine superclass behavior
  - area (i.e., can take into border)



Adding State and Behavior

A subclass can add specific state and behavior:

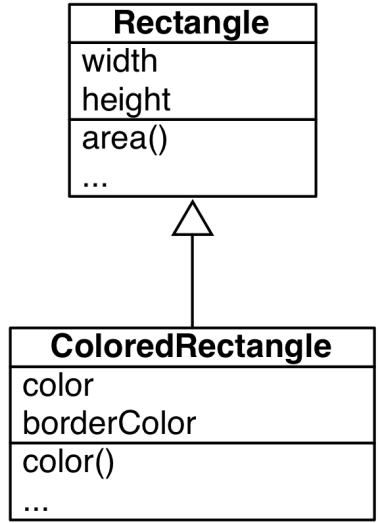
- color, borderColor, ...



Use Superclass Behavior

A subclass can use superclass behavior and state

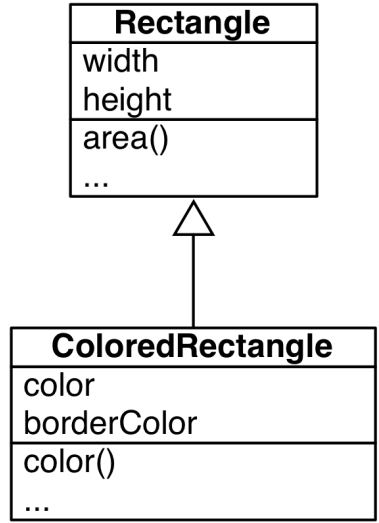
- saving a ColoredRectangle will use methods of the superclass to get information



Specialising Superclass Behavior

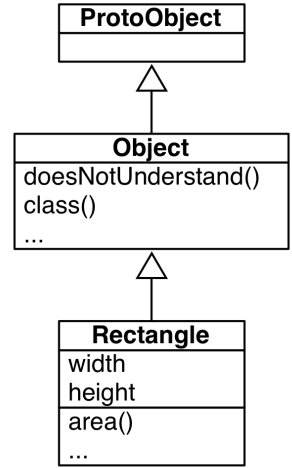
A subclass can specialize and redefine superclass behavior

- area (i.e., can take into border)
- the method ColoredRectangle >> area specialise the method Rectangle>>area



Root of Inheritance Hierarchy

- Object is the root of most classes
- ProtoObject (Object's superclass) is for special purposes (like raising a many as errors as possible)...
  - ...but we will ignore it as it is not important



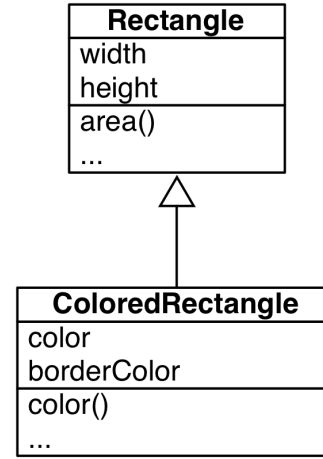
The Basics

Inheritance is

- Static for state (i.e., during class creation)
- Dynamic for behavior (i.e., during execution)

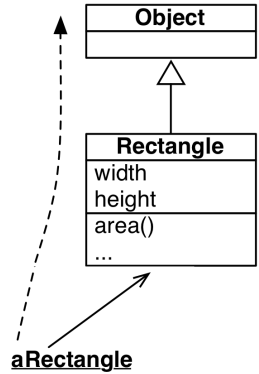
Inheritance of Instance Variables

- Happens during class definition compilation
- computed from
  - the class own instance variables
  - the ones of its superclasses
- ColoredRectangle instances will be characterized by width, height, color, borderColor instance variables



Inheritance of Behavior

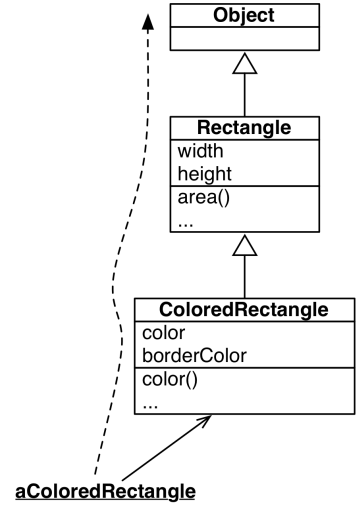
- Happens at runtime
- Depends on the receiver of the message
- The method is searched
  - starting from the receiver's class
  - then when not found going to the superclass



Inheritance of Behavior

- Happens at runtime
- Depends on the receiver of the message
- The method is searched
  - starting from the receiver's class
  - then when not found going to the superclass

(we will explain more in the next lectures)



Resources

- Pharo mooc - Videos W4S01: `http://mooc.pharo.org`
- Pharo by Example: `http://books.pharo.org`

What You Should Know

- Inheritance allows a class to refine state and behavior
- A class has 1 and only 1 superclass
- A class eventually inherits from Object
- Inheritance of state is static (happens at class compile time)
- Inheritance of behavior is dynamic (happens at message execution)

A course by Stéphane Ducasse

http://stephane.ducasse.free.fr

Reusing some parts of the Pharo Mooc by

Damien Cassou, Stéphane Ducasse, Luc Fabresse

http://mooc.pharo.org