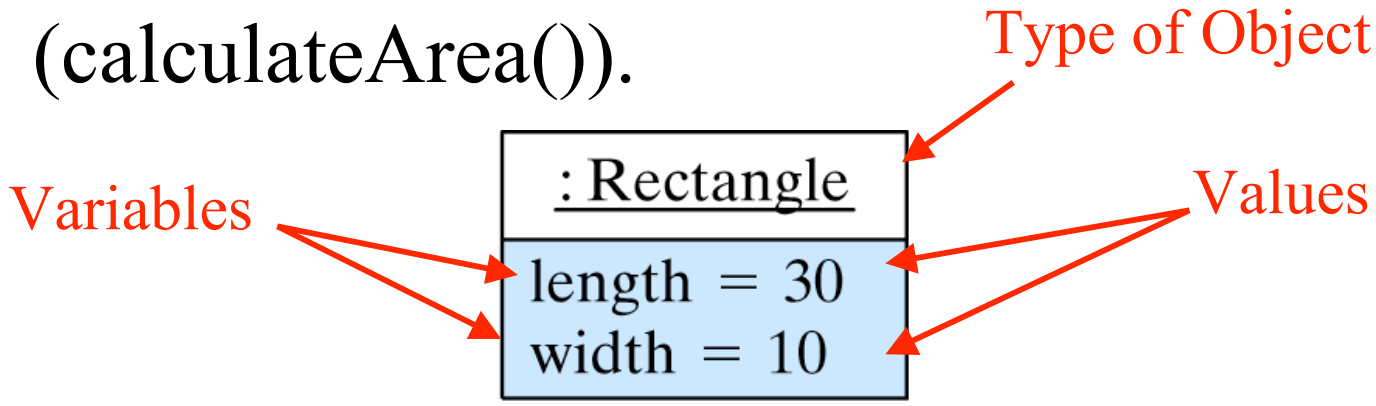
What is an Object?

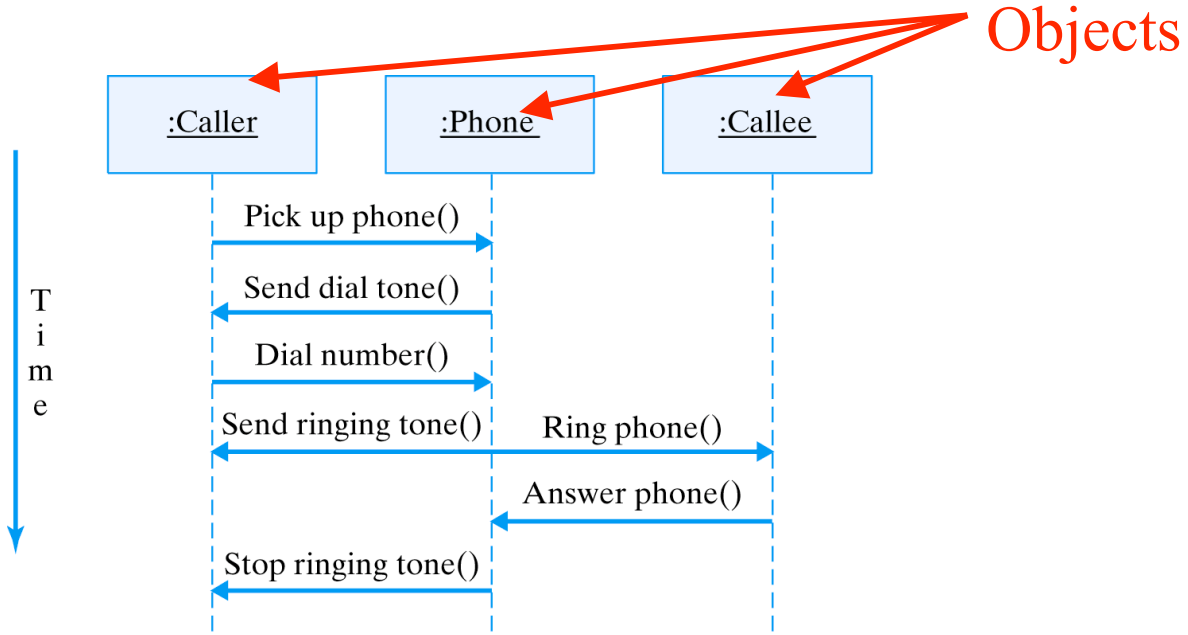
- In the real world an ***object*** (person) has ***attributes*** (name, hair color) and ***behaviors*** (eating, brushing teeth).
- In Java, an object (rectangle) has ***variables*** (length, width) and ***methods*** (calculateArea()).



A UML Object Diagram

What Is Object-Oriented Programming?

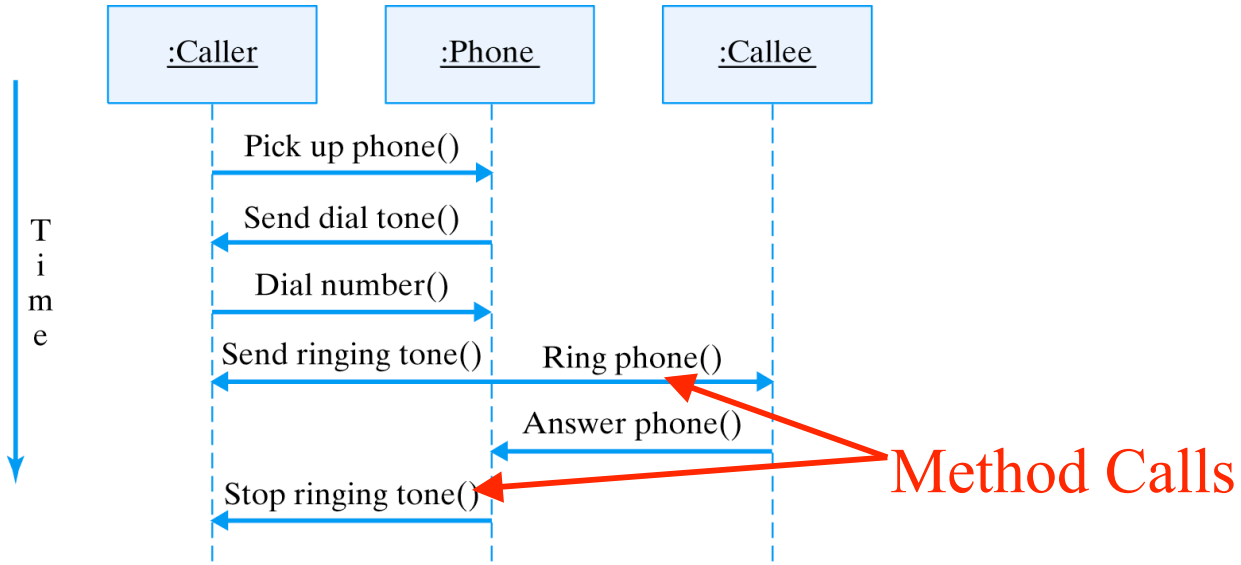
- Interacting Objects
  - An OOP is a set of interacting objects that communicate by sending messages to each other.



A UML Sequence Diagram

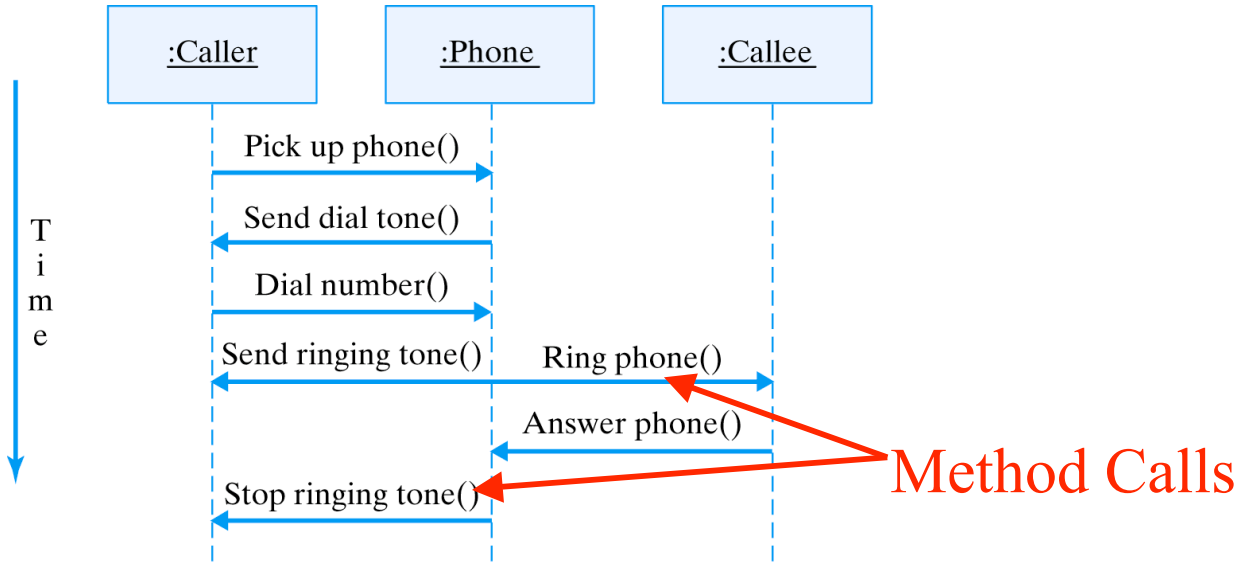
What is a Message?

- A ***message*** represents the passing of information from one object to another.
- In Java, passing a message is done by ***calling a method***.



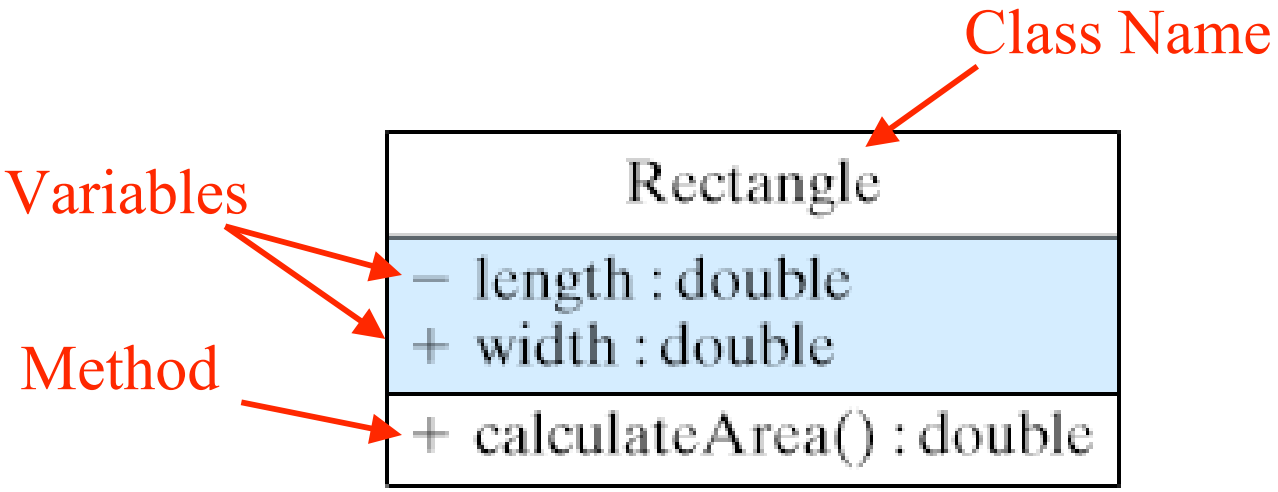
Passing Information

- One or more ***parameters*** may be sent when calling a method.
- A ***return value*** is used to pass information back at the end of the method.



What is a Java Class?

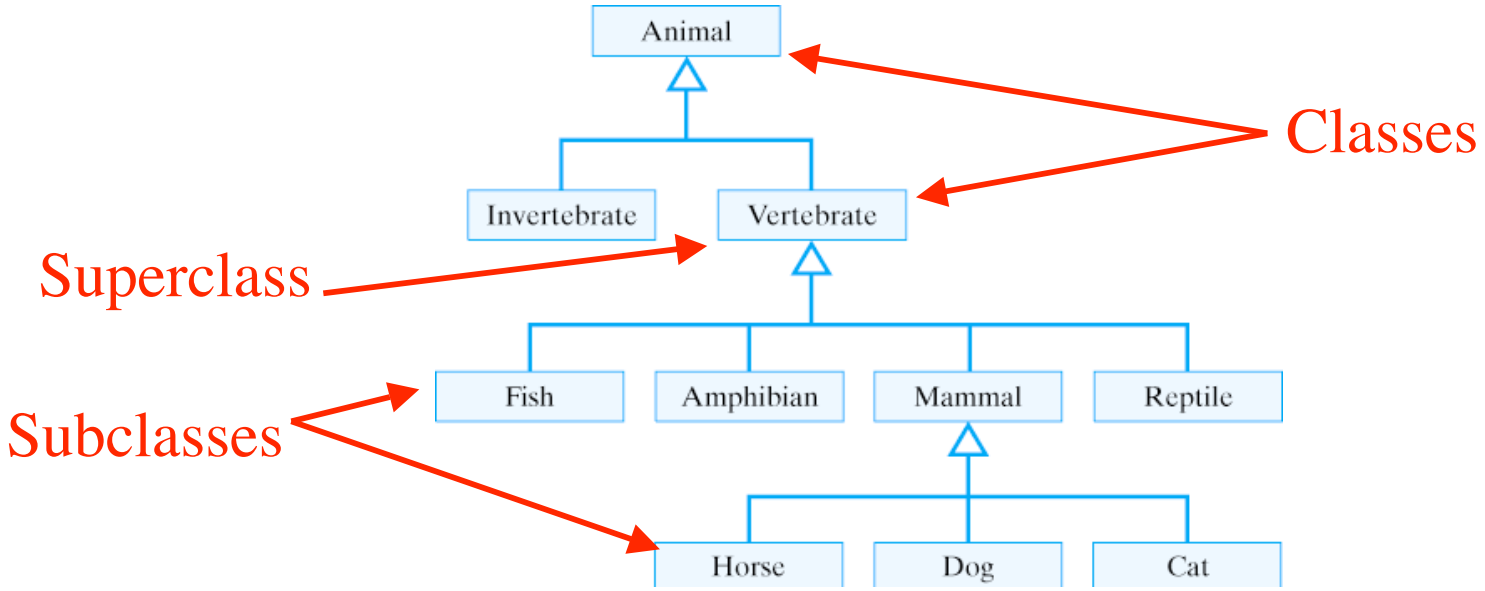
- A ***class*** (e.g., Rectangle) is a ***blueprint*** or ***template*** of all objects of a certain type.
- An object is an ***instance*** of a class.



A UML Class Diagram

What is Class Inheritance?

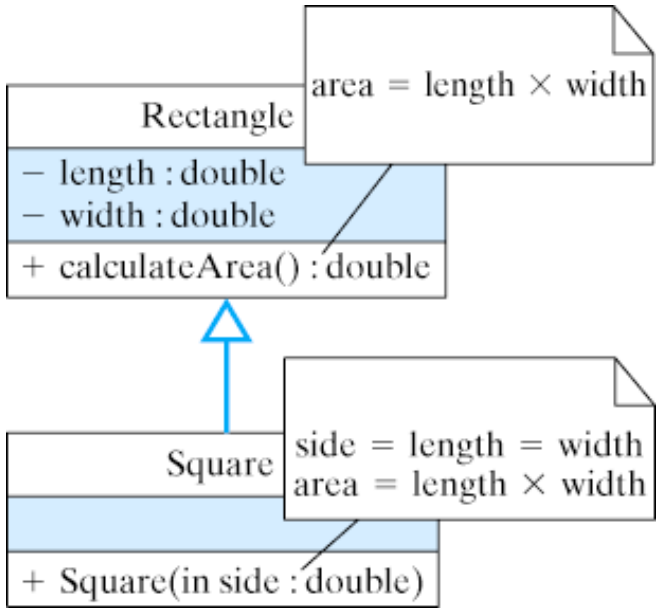
- Inheritance: Objects belonging to a ***subclass*** inherit certain characteristics and behaviors from objects belonging to a ***superclass*.**



A UML Class Diagram

Extending a Class

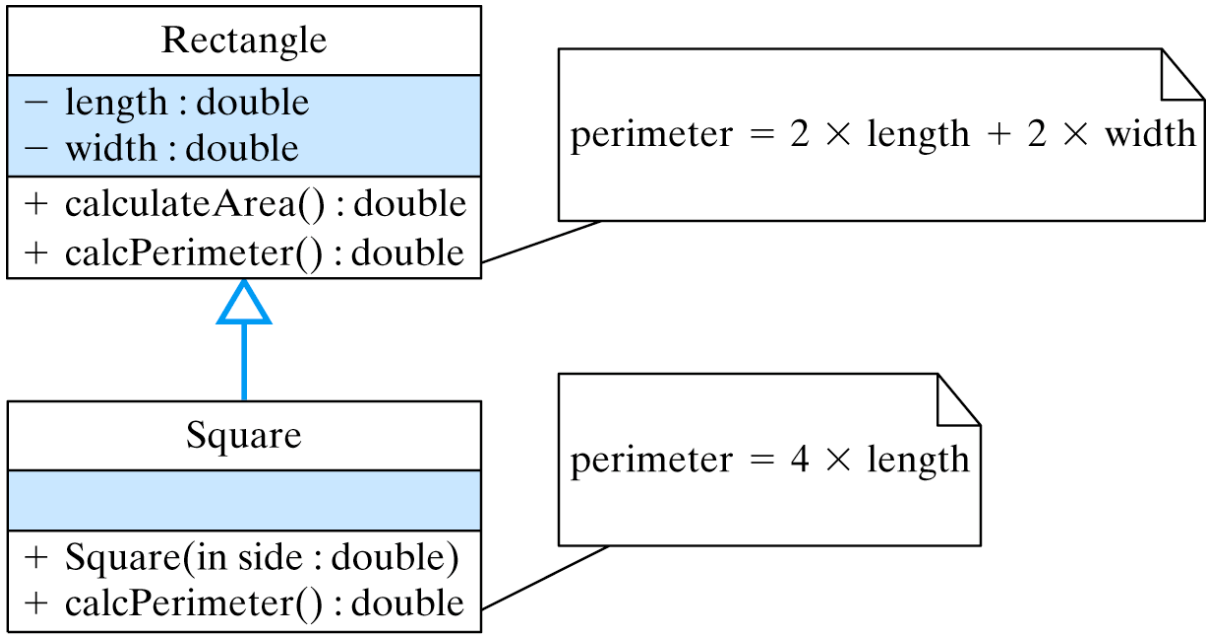
- Code Reuse: Inheritance allows us to define one class in terms of another.



A **Square** is as **Rectangle** whose sides are equal. The **Square** class inherits the **calculateArea()** method.

Overriding a Method

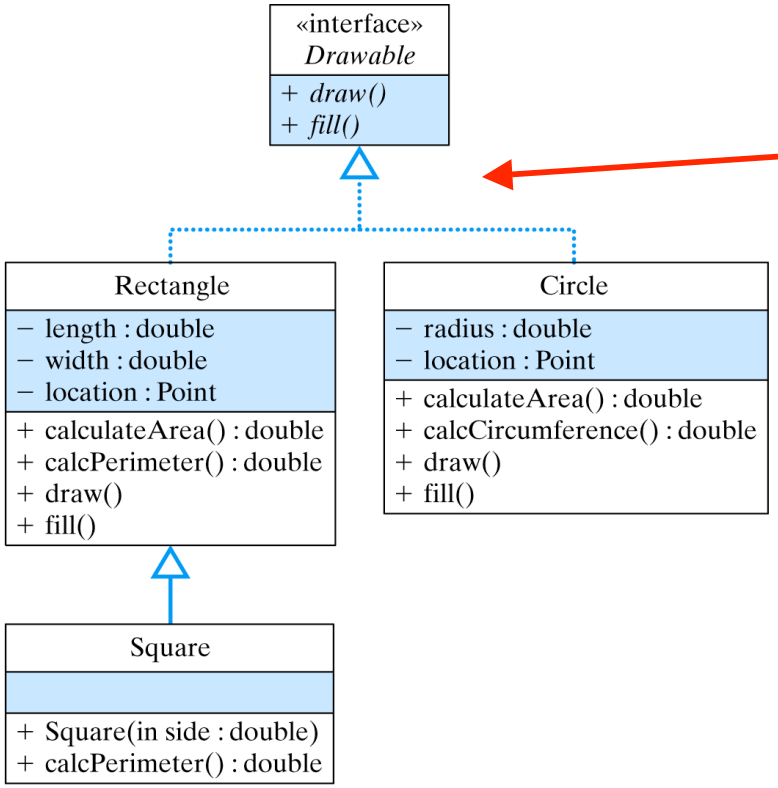
- Code Reuse: A method can be ***overridden*** by defining it in the subclass.



A **Square** class can be given a more efficient **calcPerimeter()** method.

What is an Polymorphism?

- A ***polymorphic method*** has different behavior for different objects.



The draw() and fill() methods may be polymorphic.