

Objects and Classes

Object Oriented Programming

A computer program is a **model** of the world.

We want to **structure** our model in a logical and comprehensible fashion.

One natural structure is to represent the world as a collection of interacting **objects** of different kinds.

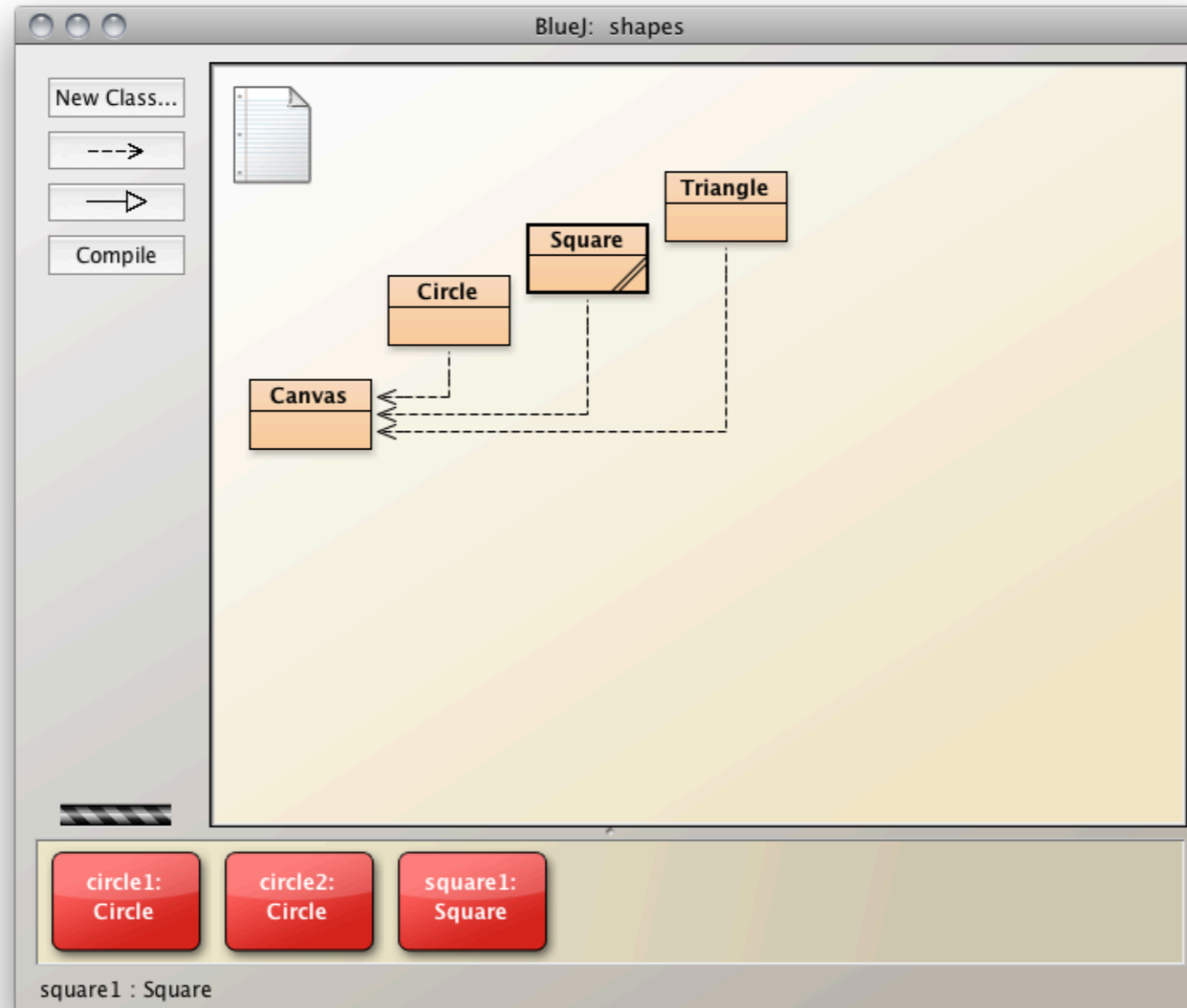
Objects and Classes

Objects represent 'things' from the real world, or from some problem domain (example: "the red car down there in the car park")

Classes represent all objects of a kind (example: "car").

All objects in a class share a common **structure** but have different **details**.

Creating Objects

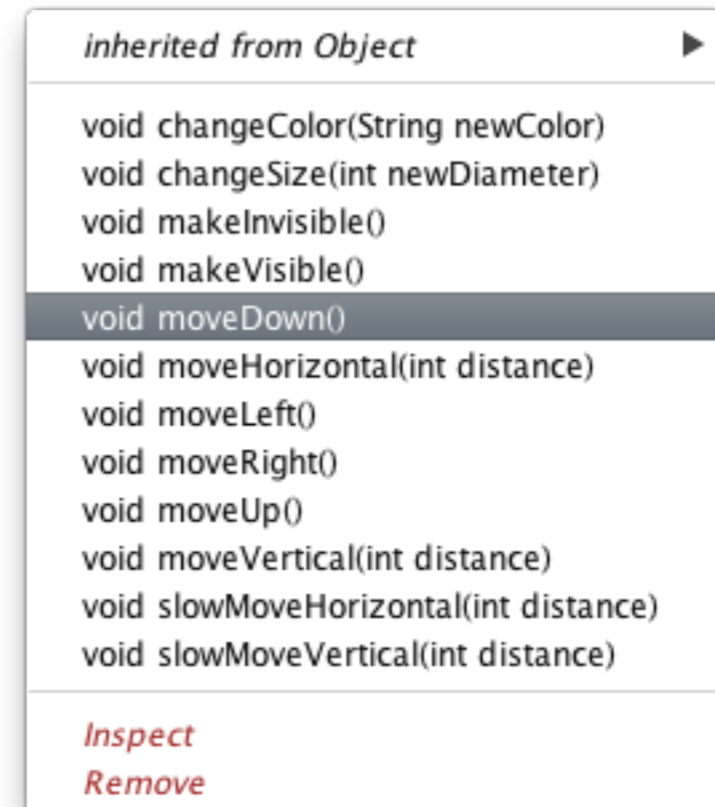


Methods

Objects have **methods**.

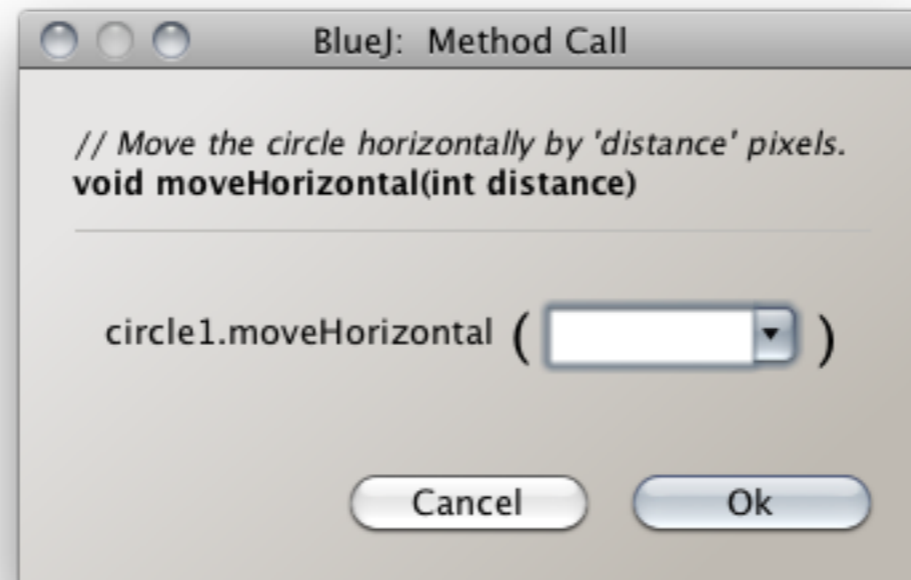
Methods provide ways to **do things** with the object.

We **call** (or invoke) a method to execute it.



Method Parameters

Some methods have **parameters** which affect their operation.



Data types

Parameters have **types** which describe what **kind of data** is needed.

The `int` type represents **integers** (whole numbers).

The `String` type represents **text**.

We'll encounter other types later.

Multiple instances

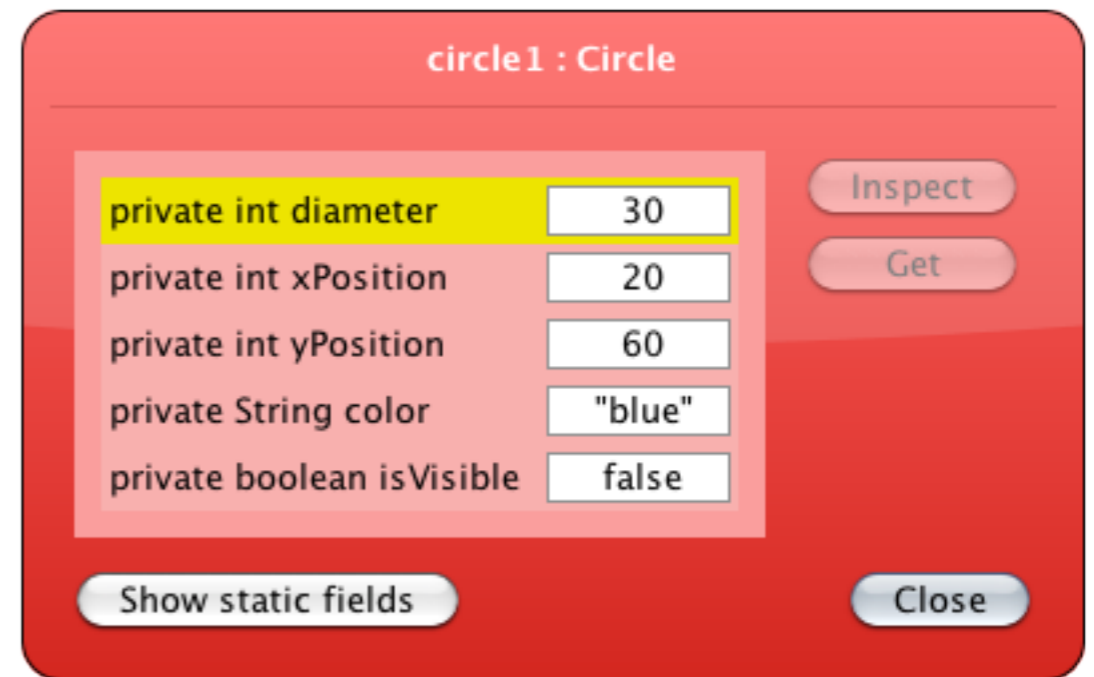
We can create multiple objects from the same class.

We say that an object is an **instance** of the class.

State

Each object has a **state** defined by a set of **fields**.

All objects in a class have the **same fields**, but they may have **different state**.



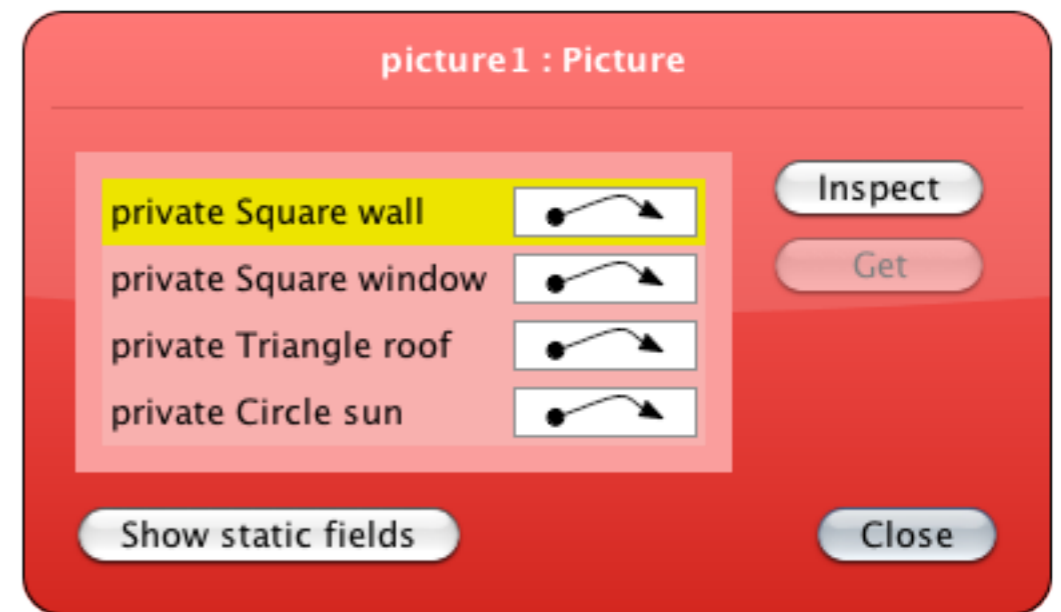
Object interaction

Objects can **refer** to other objects.

A method can:

construct other objects.

call other methods on different objects.



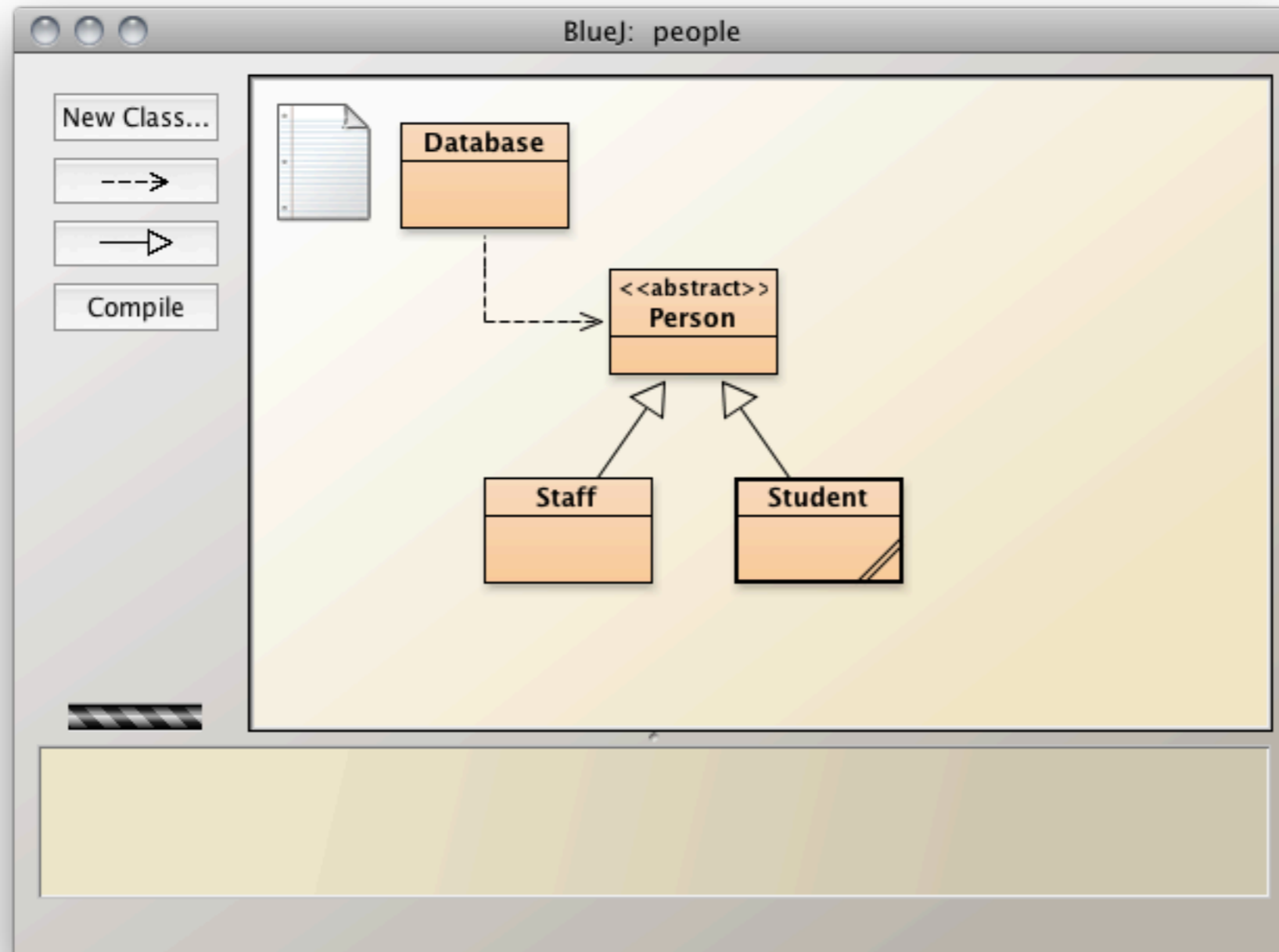
Source code

Each class has **source code** that defines how it works.

Changing the source code changes the behaviour of the class.

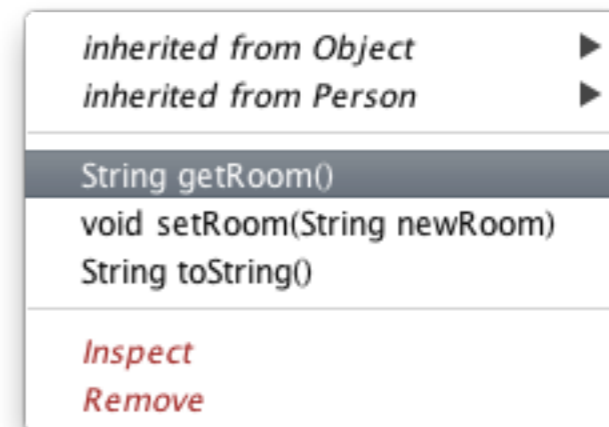
Source code needs to be **compiled** in order for changes to take effect.

Another example



Return values

Some methods provide a **return value** to get some information back from the object.

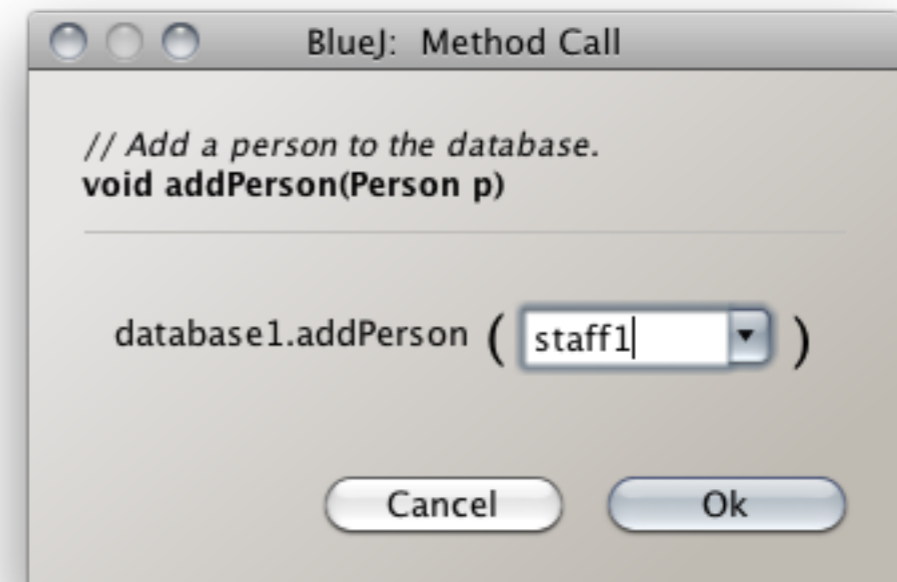


The return value also has a **type** to let you know what kind of data it provides.

The type `void` indicates there is no return value.

Objects as data

Objects can be used as data for methods in the same way as an `int` or `String`.



The object's **data type** is its **class**.

Objects can be used both as parameters and as return values.