

OOP – Inheritance

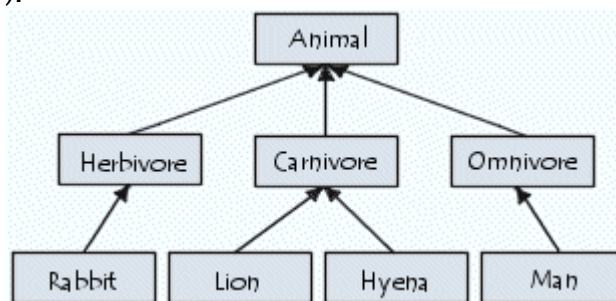
The concept of inheritance

Inheritance is specific to object-oriented programming, where a new class is created from an existing class. *Inheritance* (often referred to as *subclasses*) comes from the fact that the subclass (the newly created class) contains the attributes and methods of the parent class. The main advantage of inheritance is the ability to define new attributes and new methods for the subclass which are then applied to the inherited attributes and methods.

This can be used to create a highly specialized hierarchical class structure. The biggest advantage is that there is no need to start from scratch when wanting to specialize an existing class. As a result, class libraries can be purchased, providing a base that can then be specialized at will (the company selling these classes tends to protect member data using encapsulation).

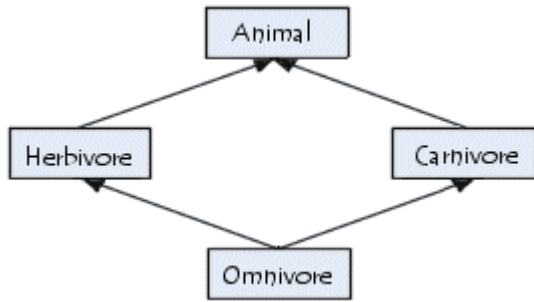
Class hierarchy

The parent-child relationship between classes can be represented in a hierarchical view often called *class tree view*. The class tree view starts with a general class called superclass (sometimes referred to as *base class*, *parent class*, *ancestor class*, *mother class* or *father class*), there are many genealogical metaphors). Derived classes (*child class* or *subclass*) become more specialized further down the tree. Therefore we usually refer to the relationship that links one child class to a parent class by the phrase "is a(n)" x of y).



Multiple inheritance

Some object oriented languages, such as C++ allow multiple inheritance, meaning that one class can inherit attributes from two superclasses. This method can be used to group attributes and methods from several classes into one single class.



Source: <http://en.kioskea.net/contents/422-oop-inheritance>