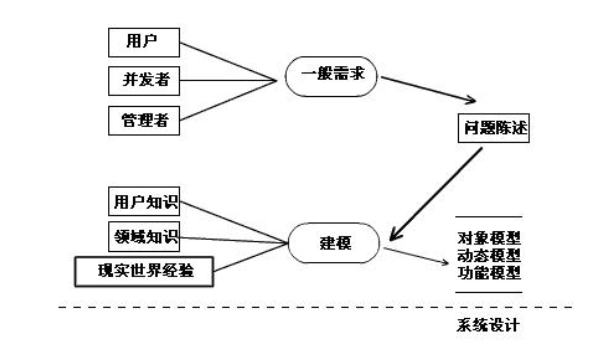
# 2.1面向对象的基本概念[3][5]

**软件工程学家Codd和Yourdon都认为：**

**面向对象 = 对象 + 类 + 继承 + 通信**

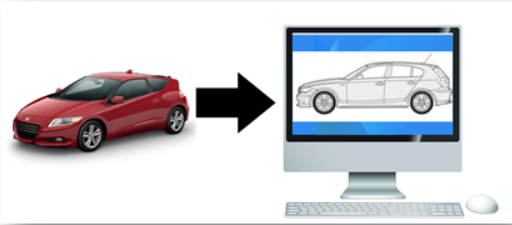
**在学习面向对象程序设计之前，一般都会学习面向过程的程序设计，例如，使用C面向过程的程序设计语言，面向过程的语言是按流程化的思想来组织的。在这些语言的设计思想中，通常将存放基本数据类型的变量作为程序处理对象、以变量的赋值作为程序的基本操作、以变量值的改变作为程序运行的状态。这种程序设计风格存在着**数据抽象简单**、**信息完全暴露**、**算法复杂**、**无法很好地描述客观世界**等缺点。在程序设计过程中，为了**实现有限度的代码重用**，公共代码被组织成为**过程或函数**。当需要代码重用时，调用已经组织好的过程或函数。在这种应用方式中，如果软件项目庞大，程序的调试和维护将变得异常困难。**



## 2.1.1 面向对象的方法[3][5]

尽可能模拟人类习惯的思维方式，使开发软件的方法与过程尽可能接近人类认识世界解决问题的方法与过程。

由现实的事物->计算机处理额模型



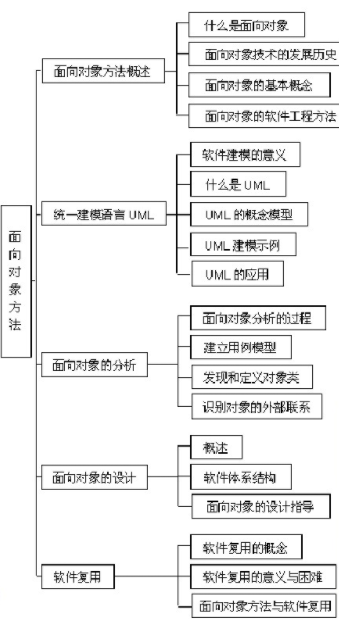
面向对象方法具有以下几个要点：

（1）面向对象的软件系统是由**对象**组成的，软件中的任何元素都是对象，复杂的软件对象由比较简单的对象组合而成。

（2）把所有对象都划分成各种对象类，每个对象类都定义了一组数据和一组方法，数据用于表示对象的**静态属性**，是对象的**状态信息**。

（3）按照子类与父类的关系，把若干个对象类组成一个**层次结构**的系统。

（4）对象彼此之间仅能通过传递消息进行联系。



## 2.1.2 对象[1][3][5]

对象（Object）是面向对象的**基本构造单元**。是系统中用来**描述客观事物的一个实体**，一个对象由**一组属性**和对属性进行操作的**一组方法**组成。



对象不仅能表示具体的实体，也能表示抽象的规则、计划或事件。主要有如下的对象类型：

1. 有形的实体：如汽车、书、计算机。

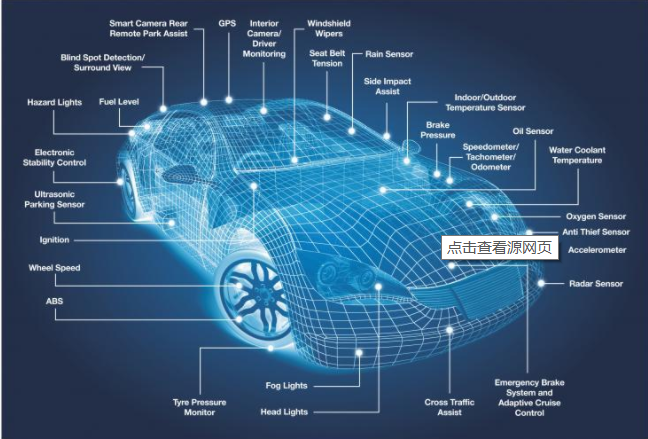
1. 作用：如医生、教师、员工、学生。

1. 事件：如提问、事故、中断、开会等。

（4）性能说明：如车厂对车辆的性能说明，往往要列出型号及各种性能指标等。



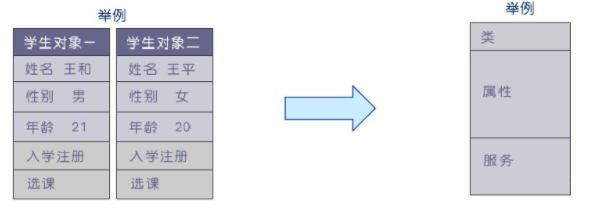
## 2.1.3 类[1][3][5]

一个类定义了一组大体上相似的对象。一个类所包含的方法和数据描述一组对象的共同行为和属性。例如，窗口，车轮，玻璃等都是类的例子。

我们一般人的分类如下图所示，就是找到目标的本质特征然后将其共性进行归类，得出一个抽象的概念。

但面向对象中的类要更加详细和具体。如下图：



“学生”是一个类名,其属性有姓名，性别，年龄等，可以定义“如雪注册”，“选课”等操作。

## 2.1.4 封装[3][5]

**只隐藏对象的属性和实现细节，仅对外提供公共访问方式**

封装（Encapsulation）就是把一个对象的方法和属性组合成一个**独立的单位**，并尽可能隐蔽对象的属性、方法和实现细节的过程。

对外访问时提供了4种访问控制级别：

（1）public：公有访问。

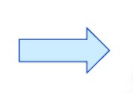
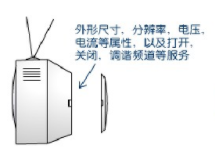
（2）protected：受保护的。

（3）private：私有访问。

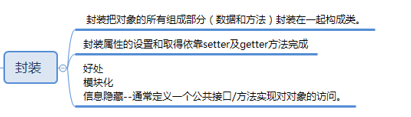
（4）default：默认的。

举例：

电视机包括外形尺寸，分辨率，电压电流等属性，具备打开关闭，调频，转换频道等服务，封装就意味着将其属性和服务结合成一个不可分的整体，对外有电视屏，插头，按钮等借口，用户可以通过接口使用电视，但不关心内部的实现细节。



但封装也存在副作用，如果过于强调严格的封装，将会使得对象的任何属性都不允许外部直接存取，因此需要增加许多意※义※不※明的只负责读写的服务，增添了编程工作的负担和运行的开销。因此对象往往拥有不同程度的可见性。



## 2.1.5 继承[1][5]

**提高代码复用性；继承是多态的前提**

继承（Inheritance）又称之为“泛化”，是一种一般类与特殊类的层次模型。继承性是指特殊类的对象具有其一般类的属性和方法，在其之上又增加了自己的特殊属性和方法。

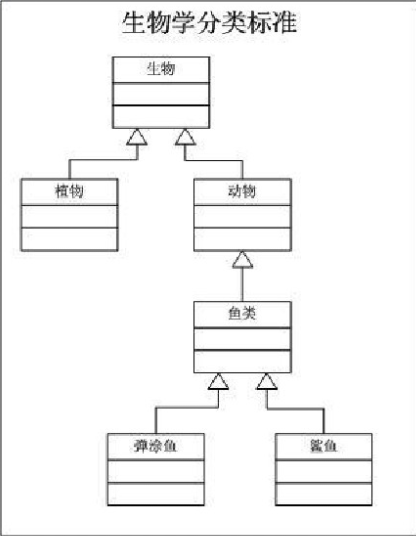
在继承中，需要明确这样两个概念，子类和父类。

子类：指的是通过继承创建的新类称为“子类” 或者“派生类”。

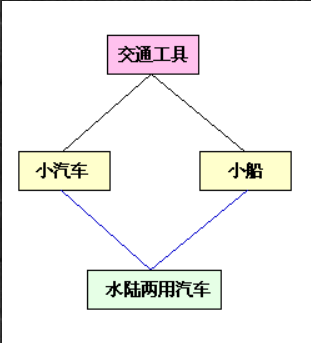
父类：指的是被继承的类称为“基类”、“ 父类” 或“ 超类”。

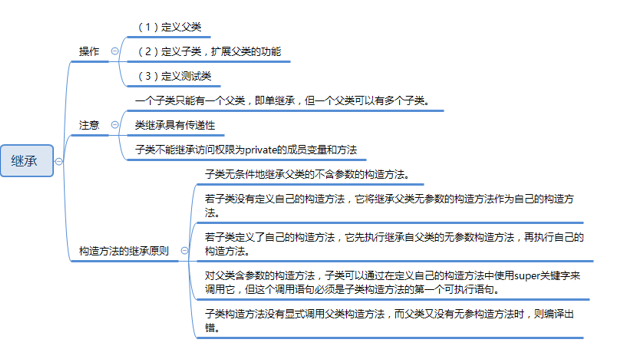
继承性分为单重继承和多重继承两类。

单重继承：指的是一个子类只有一个父类；



多重继承：指的是一个子类可以有多个父类。



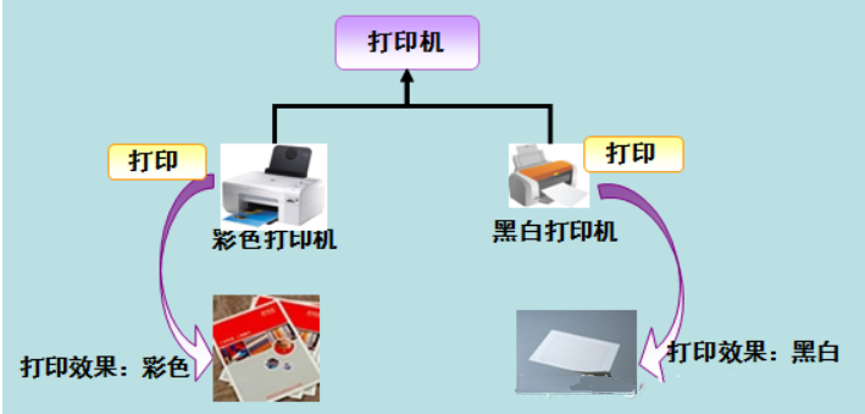


## 2.1.6 多态[1][5]

**是父类或接口定义的引用变量可以指向子类或具体实现类的实例对象**

多态性（Polymorphism）是指类中**同一函数名对应多个功能相似的不同函数**，可以使用相同的调用方式来调用这些具有不同功能的同名函数，这些同名的函数可以是**参数的个数或是类型不同**，但是函数名相同，当进行调用的时候，根据所传的数据选定相应的函数，从而去执行不同的功能。

多态性的机智不但为软件的结构提供了灵活性，减少了信息的冗余，而且显著的提高了软件的复用性和可拓展性。



多态的优点

1. 消除类型之间的耦合关系

2. 可替换性

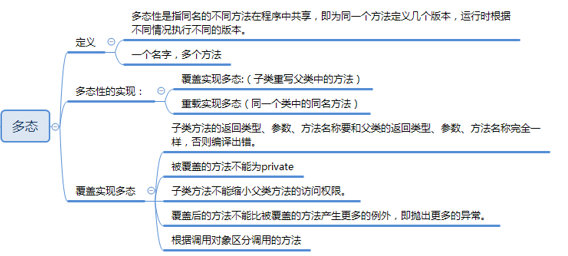
3. 可扩充性

4. 接口性

5. 灵活性

6. 简化性

与多态性实现有关的语言功能包括：重载，动态绑定，类属等等



## 2.1.7 消息[3][5]

消息是实现对象之间进行通信的一种机制，对于一个对象可以接收不同形式的多个消息，并产生不同的结果；相同形式的消息可以发送给不同的对象，并产生不同的结果；在发送消息的时候可以不考虑具体的接收者，对象可以对消息做出响应，也可以拒绝消息，也就是说不是必须要对消息做出响应。

通常，一个消息由以下几部分组成：

（1）提供服务的对象名。

（2）服务的标识，即方法名。

（3）输入信息，即实际参数。

（4）响应结果，即返回值或操作结果。

举例：

还是看电视的例子

用户通过按钮，遥控发出转换频道的信息，电视机变换对电视台的接收信号频率，并将结果显示给用户。

用户发出的信息：

接受者—电视机

要求的服务—转换频道

输入信息—转换后的频道序号

应答信息—转换后的频道节目

## 3.1 参考资料

[1]<https://wenku.baidu.com/view/bb428d2d4631b90d6c85ec3a87c24028915f85c5.html>

[2] https://zhuanlan.zhihu.com/p/36574600

[3]https://wenku.baidu.com/view/078f4a3b88eb172ded630b1c59eef8c75ebf95f3.html

[4]《UML用户指南》 人民邮电出版社 Grady Booch, James Rumbaugh, Ivar Jacobson著 邵维忠 麻志毅 马浩海 刘辉 译 2013年1月第1版

[5]《UML2基础、建模与设计教程》 清华大学出版社 杨弘平等 2015年10月第1版

问题：

1. 面向对象技术的三个特征

答案：

封装，隐藏内部实现  
继承，复用现有代码  
多态，改写对象行为

1. 面向对象系统中功能复用的两种最常用技术是什么？

答：**类继承和对象组合**，类继承允许你根据其他类的实现来定义一个类的实现。父类的内部细节对子类可见。类继承是在编译时刻静态定义的，且可直接使用，类继承可以较方便地改变被复用的实现。对象组合是类继承之外的另一种复用选择。新的更复杂的功能可以通过组装或组合对象来获得。对象组合要求被组合的对象具有良好定义的接口。

3. 面向对象技术的四个特征：

答：抽象，分离变与不变

封装，隐藏内部实现

继承，复用现有代码

多态，改写对象行为

4. 什么是多态

答：多态是指父对象中的同一个行为能在其多个子对象中有不同的表现。也就是说子对象可以使用重写父对象中的行为，使其拥有不同于父对象和其它子对象的表现，这就是overriding（重写）