**1.1 抽象的进步**

如FORTRAN，BASIC和C 它们的抽象原理依然要求我们着重考虑计算机的结构，而非考虑问题本身的结构

OOP允许我们根据问题来描述问题，而不是根据方案

1. 所有东西都是对象。可将对象想象成一种新型变量
2. 程序是一大堆对象的组合；通过消息传递，各对象知道自己该做些什么
3. 3) 每个对象都有自己的存储空间
4. (4) 每个对象都有一种类型 根据语法，每个对象都是某个“类”的一个“实例”
5. (5) 同一类所有对象都能接收相同的消息

**1.2 对象的接口**

类型”决定了接口，而“类”是那个接口的一种特殊实现方式。

“接口”（Interface）规定了可对一个特定的对象发出哪些请求。

**1.5 继承：重新使用接口**

在Java语言中，继承是通过extends关键字实现的 使用继承时

有两种做法可将新得的衍生类与原来的基础类区分开。第一种做法十分简单：为衍生类添加新函数（功能）第二个办法是改变基础类一个现有函数的行为

**1.6 多形对象的互换使用**

我们要进行的一项重要处理就是将衍生类的对象当作基础类的一个对象对待

1.6.2 抽象的基础类和接口

设计程序时，我们经常都希望基础类只为自己的衍生类提供一个接口。也就是说，我们不想其他任何人实际创建基础类的一个对象，只对上溯造型成它，以便使用它们的接口。为达到这个目的，需要把那个类变成“抽象”的——使用abstract关键字。