# 第一章对象入门

1.2 对象接口

如果不显式的给出关键字 private，public，protected。那么就是隐式friendly，类似于internal

在Java 语言中，继承是通过extends 关键字实现的

**等价与类似关系**

一，我们完全能够将衍生类的一个对象换成基础类的一个对象！可将其想象成一种“纯替换”。就是衍生类没有定义新的函数。等价关系。

二，我们必须为衍生类型加入新的接口元素。所以不仅扩展了接口，也创建了一种新类型。这种新类型仍可替换成基础类型，但这种替换并不是完美的，因为不可在基础类里访问新函数。我们将其称作“类似”关系

**抽象类抽象方法**

抽象方法也许只能在一个抽象类里创建。继承了一个类后，那个方法就必须实现，否则继承的类也会变成“抽象”类

**关于接口**

interface（接口）关键字将抽象类的概念更延伸了一步，它完全禁止了所有的函数定义。“接口”是一种相当有效和常用的工具。另外如果自己愿意，亦可将多个接口都合并到一起（不能从多个普通class 或abstract class 中继承）。

**关于Import**

﻿Java编译器我们希望的类是什么。쀀 import的作用是指示编译器导入一个“包쀀 ”——或者说一个“类库”（在其他语言里，可将“库”想象成一系列函数、数据以及类的集合。但请记住， Java的所有代码都必须写入一个类中）。

**关于Static**

Static方法不能调用实例方法

**开关语句**

完全可以在default 语句的末尾放置一个break，尽管它并没有任何实际的用处。

**文件中的类：**

一个java文件中必须要有一个public类和文件的名字一样，而且还可以包含其它的类。但对于.java 文件中的每个类，它们都有一个.class 扩展名。因此，我们最终从少量的.java 文件里有可能获得数量众多的.class 文件。

**关于classpath**

不要错误地以为Java 无论如何都会将当前目录作为搜索的起点看待。如果不将一个“.”作为CLASSPATH 的一部分使用，Java 就不会考虑当前目录。

## 第六章 类的再生

主要是类的合成和继承

**合成**

我们只需在新类里简单地置入对象句柄即可

关于tostring 方法，若编译器本来希望一个String，但却获得某个这样的对象，就会调用这个方法。

Final 方法

设计程序时，若希望一个方法的行为在继承期间保持不变，而且不可被覆盖或改写，就可

以采取这种做法。

第七章 polymorphism 多态

构造函数实际上是静态的，只是static是隐式的。

### 第十章 Java IO系统

从InputStream（输入流）衍生的所有类都拥有名为read()的基本方法，用于读取单个字节或者

字节数组。类似地，从OutputStream 衍生的所有类都拥有基本方法write()，用于写入单个字节或者字节数组。