

## 对象与类

# 对象与类

1. 类的构造器方法，用于创建和初始化一个对象，通过new 来调用。构造器没有返回值。
2. 在一个类中不要编写引用可变对象的访问器方法，这样会破坏封装性。一旦接受这个方法返回值的变量改变了该可变对象，类的实例域(成员变量)就随之改变。正确的方式是返回该可变对象的克隆(clone) (P113)
3. 方法可以访问所属类的所有对象的私有数据。也就是说权限修饰符限制的是类的访问权限而非对象。对一个类的不同对象可以通过方法互相访问私有数据。一个类中的辅助方法由于往往与当前的实现细节紧密相连，或者需要特别的协议以及一个特别的调用次序，因此不应该设置为公有，如果数据的表达方式发生了改变，不再需要某一个私有方法，那么可以直接去掉，但对公有方法则不行，因为可能有其他的外部类调用它。(P114)
4. 封装：将对象的实例域隐藏起来，对外暴露更改和访问实例域的方法，外部不能直接访问实例域，可以通过暴露出来方法访问。
5. 类文件名与源文件中的公有类（public）相同，在一个源文件中只能包含一个公有类，但可以有任意数目的非公有类。
6. 静态域和静态方法属于类而非对象，每个对象只是拿到静态域的一个拷贝。当一个方法不需要访问对象的状态或者只需要访问类的静态域时，使用静态方法。
7. 方法参数：在Java中参数传递均为按值传递。基本数据类型参数是按值传递，方法不能修改其值。方法可以改变一个对象参数的状态。对象参数看似是按引用传递，其实只是传递了对象参数的拷贝，方法不能让对象参数引用一个新的对象。
8. this关键字：a. 代表调用该方法的对象，通常作为隐式参数传入方法中 b. 在构造器中调用重载的另一个构造器，this(...)的形式
9. 初始化数据域的方法：a. 在构造器中初始化 b. 在声明中赋值 c. 初始化块中初始化

## 8. 工厂方法\*\*\*