Effective Java

# Item1

考虑用静态工厂方法代替构造器

## 优势:

* 静态工厂方法与构造器不同的第一大优势在于:他们有名称。

当一个类需要多个带有相同签名的构造器时,就用静态工厂方法代替构造器,并且慎重的选择名称以便突出他们之间的区别.

* 不必在每次调用他们的时候都创建一个新对象.这使得不可变类可以使用预先构建好的实例,或者将构建好的实例缓存起来进行重复利用.
* 静态工厂模式可以返回原返回类型的任何子类型的对象.这项技术适用于基于接口的框架(interface-based framework)
* 静态工厂方法返回的对象所属的类,在编写包含该静态工厂方法的类时可以不必存在,这种灵活的静态工厂方法构成了服务提供者框架(service Provider framework)的基础
* 在创建参数化类型实例的时候,它们使代码变得更加简洁.

## 劣势

* 类如果不含有共有的或者受保护的构造器,就不能被子类化.
* 它们与其他的静态方法实际上没有任何区别.

## 惯用名称

* valueOf—不太严格地讲,该方法返回的实例与他的参数具有相同的值，这样的静态工厂方法实际上市类型转换
* of—valueOf的一种更为严格地替代
* getInstance—返回的实例是通过方法的参数来描述的,但是不能够说与参数具有相同样的值
* newInstance—像getInstance一样,但newInstance能够返回的每个实例都与其他实例不太

## 结构

服务提供者框架中有三个重要的组件:服务接口(service interface)，这是提供者实现的,提供者注册API(Provider Registration API ),这是系统用来注册实现,让客户端访问他们的,服务访问API(Service Access API),是客户端用来获取服务的实例的,服务访问API一般允许但是不要求客户端指定某种选择提供者的条件,如果没有则api会返回默认实现的一个实例,服务访问API是灵活的静态工厂,它构成了服务提供者框架的基础.

## 总结

* 简而言之,静态工厂方法和共有构造器都各有用处,我们需要理解它们各自的长处.静态工厂通常更加合适,因此切忌第一反应时提供共有的构造器.