Effective Java

# 考虑用静态工厂方法代替构造器

类可以通过静态工厂方法来提供他的实例，而不是通过构造器。提供静态工厂方法而不是共有的构造器。

四大优势：

第一大优势：它们有名称；

第二大优势：不必再每次调用它们的时候都创建一个新对象；

第三大优势：它们可以返回原返回类型的任何子类型的对象；

第四大优势：在创建参数化类型实例的时候，它们使代码变得更加简洁。

两个缺点：

第一个缺点：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；

第二个缺点：它们与其它的静态方法实际上没有任何区别；

静态工厂方法惯用名称：

valueOf ---- 该方法返回的实例与它的参数具有相同的值，实际上是类型转换方法；

of ---- valueOf的一种更为简洁的替代，在EnumSet中使用并流行起来；

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同样的值。对于Singleton来说，该方法没有参数，并返回唯一的实例；

newInstance ---- 像getInstance一样，但newInstance能够确保返回的每个实例都与所有其它实例不同；

getInstance ---- 像getInstance一样，但是在工厂方法处于不同的类中的时候使用。Type标示工厂方法所返回的对象类型；

newType ---- 像newInstance一样，但是在工厂方法处于不同的类中的时候使用。Type标示工厂方法所返回的对象类型。

# 遇到多个构造器参数时考虑用构建器