第七章 方法

2018年7月17日

10:28

**第38条：检查参数的有效性**

非公有方法通常使用断言（assertion）来检查它们的参数。

每当编写方法或者构造器的时候，应该考虑它的参数有哪些限制。

应该把限制写到文档中，并且在这个方法的开头处，通过显式的检查来实施这些限制。

**第39条：必要时进行保护性拷贝**

假设类的客户端尽其所能来破坏这个这个类的约束，因此你必须保护性地设计程序。

为了保护实例的内部信息，对于构造器的每个可变参数进行保护性拷贝是必要的。

保护性拷贝是在检查参数的有效性之前进行的，并且有效性检查是针对拷贝之后的对象，而不是针对原始对象

对于参数类型可以被不可信任方子类化的参数，请不要使用clone方法进行保护性拷贝。

在内部对象被返回给客户端之前，也需要对它们进行滚保护性拷贝。

**第40条：谨慎设计方法签名**

* 谨慎地选择方法的名称。
* 不要过于追求提供便利的方法。
* 避免过长的参数列表。
* 队医参数类型，要优先使用接口而不是类。
* 对于boolean参数，要优先使用两个元素的枚举类型。

**第41条：慎用重载**

方法重载是在编译时做出的决定的。

重载方法的选择是静态的，而对于被覆盖的方法的选择是动态的，被覆盖的方法是在运行时决定的。

安全而保守的策略是，永远不要导出两个具有相同参数数目的重载方法。

方法使用可变参数时，保守的策略时根本不要重载它。

**第42条：慎用可变参数**

在定义参数数目不定的方法时，可变参数方法是一种很方便的方式，但它们不应该被过度滥用。

不必改造具有final数组参数的每个方法；只当确实是在数量不定的值上执行调用时才使用可变参数。

可变参数类型必须作为参数列表的最后一项，而不能放在定长参数的前面。

可变参数是兼容数组类参数的，但是数组类参数却无法兼容可变参数。

**第43条：返回零长度的数组或者集合，而不是null**

返回类型为数组或集合的方法没理由返回null，而不是返回一个零长度的数组或者集合。

**第44条：为所有导出的API元素编写文档注释。**

方法的文档注释应该间接地描述出它和客户端之间的约定。

Javadoc{@code}标签比HEML<code>或者<tt>标签更好，因为它避免了转义HTML元字符。

当为泛型或者方法编写文档时，确保要在文档中说明所有的数据类型参数。

当为枚举类型编写文档时，要确保在文档中说明常量。

为注解类型编写文档时，要确保在文档中说明所有成员，以及类型本身。