## 整洁代码

|  |  |
| --- | --- |
| **清单项目** | **分类** |
| 使用可以表达实际意图(Intention-Revealing)的名称 | 有意义的名称 |
| 每一个概念只用一个词 | 有意义的名称 |
| 使用方案/问题领域名称 | 有意义的名称 |
| 类应该是比较小的! | 类 |
| 函数应该是比较小的! | 函数 |
| 只做一件事 | 函数 |
| DRY(Don’t Repeat Yourself)原则，(拒绝重复) | 函数 |
| 用代码来解释自己的做法(译者注：即代码注释) | 注释 |
| 确定应用了代码格式化 | 格式 |
| 使用异常而不是返回码 | 异常 |
| 不要返回Null | 异常 |

## 安全

| **清单项目** | **分类** |
| --- | --- |
| 如果不用于继承，使类为final | 基础 |
| 避免重复代码 | 基础 |
| 权限限制：程序应该运行在保证功能正常的最小权限模式下。 | 基础 |
| 最小化类和成员的可访问性 | 基础 |
| 注释出安全相关的信息 | 基础 |
| 系统的输入必须检查是否有效和在允许范围内 | 拒绝服务(Denial of Service) |
| 避免对于一些不寻常行为的过分日志 | 拒绝服务(Denial of Service) |
| 在任何情况下都释放资源(流，连接等等) | 拒绝服务(Denial of Service) |
| 从异常中清除敏感信息(暴露文件路径，系统内部相关，配置) | 私密信息(Confidential Information) |
| 不要把高度敏感的信息写到日志 | 私密信息(Confidential Information) |
| 考虑把高度敏感的信息在使用后从内存中清除 | 私密信息(Confidential Information) |
| 限制包，类，接口，方法和域的可访问性 | 可访问性的扩展(Accessibility Extensibility) |
| 限制类和方法的可扩展性(通过使它为final) | 可访问性的扩展(Accessibility Extensibility) |
| 检验输入(有效的数据，大小，范围，边界情况等等) | 输入检验(Input Validation) |
| 把从不可信对象得到的输出作为输入来检验 | 输入检验(Input Validation) |
| 为native方法定义包装类(而不是定义native方法为pulibc) | 输入检验(Input Validation) |
| 把从不可信对象得到的输出作为输入来对待 | 可变性 |
| 使public static域为final(避免调用方(caller)修改它的值) | 可变性 |
| 避免暴露敏感类的构造函数 | 对象构造 |
| 避免安全敏感类的序列化 | 序列化反序列化(Serialization Deserialization) |
| 通过序列化来保护敏感数据 | 序列化反序列化(Serialization Deserialization) |
| 小心地缓存潜在的特权操作结果 | 序列化反序列化(Serialization Deserialization) |
| 只有在需要的时候才使用JNI | 访问限制 |

## 性能

| **清单项目** | **分类** |
| --- | --- |
| 避免过分的同步 | 并发 |
| 保持同步区域比较小 | 并发 |
| 知道string连接的性能情况 | 综合编程 |
| 避免创建不需要的对象 | 创建和销毁对象 |

## 综合

| **清单项目** | **分类** |
| --- | --- |
| 对可以恢复的情况使用已受检异常(checked exceptions)，对于程序错误使用运行时异常(runtime exceptions) | 异常 |
| 更多地使用标准异常 | 异常 |
| 不要忽略异常 | 异常 |
| 检查参数的有效性 | 方法 |
| 返回空数组或集合，而不是null | 方法 |
| 最小化类和成员的可访问性 | 类和接口 |
| 在pulibc类中，使用访问器方法(accessor methods)(译者注：访问器方法即我们平常用的get/set方法)而不是public域 | 类和接口 |
| 最小化本地变量的范围 | 综合编程 |
| 通过接口引用对象 | 综合编程 |
| 遵循广泛接受的命名规则 | 综合编程 |
| 避免使用finalizer | 创建和销毁对象 |
| 当你重写equals时总是重写hashCode | 综合编程 |
| 总是重写toString | 综合编程 |
| 使用枚举来代替int常量 | 枚举和注解(Annotations) |
| 使用标记接口(marker interface)(译者注：标记接口是一种没有任何行为的接口，实现它只是为了让实现类属于某种类型，如JDK中的Serializable,Cloneable等)来定义类型 | 枚举和注解(Annotations) |
| 对共享可变的数据使用同步访问 | 并发 |
| 使用executors而不是task和thread | 并发 |
| 注释中描述线程安全情况 | 并发 |
| 存在有效的JUnit/JBehave测试用例 | 测试 |