1

前言

对于代码质量的检查，一直是一个比较棘手的问题。虽然我们部门有关于java代码的各种规范，领导也多次强调对代码的评审。

但是在实际执行过程中，由于开发人员的水平参差不齐，在开发项目时还有各种其他因素，比如时间紧任务重，自己的代码不规范没有意识到，就疏于对代码质量的检查，然后草率的提交。

还有就是大家对代码检测的不重视，有种意识就是即使代码不规范，我们的程序也可以正常运行，为什么要多次一举呢。由于我们的工程和代码时间比较久远，想改却不知道如何下手，将代码规范转换成实际的编程规范时，执行起来却很困难，即使我们有心改这也将是一个长期而痛苦的过程，需要很大人力和时间。

现在，向大家推荐一个很优秀的代码质量检测工具—FindBugs，一个用静态分析的方式来寻找Java代码中Bug的工具，，它检查类或者jar文件，将字节码和一组缺陷模式进行对比以发现可能的问题。

有了静态分析工具，就可以在不实际运行程序的情况下对软件进行分析。不是通过分析类文件的形式或结构来确定程序的意图，而是通常使用Visitor模式来鉴别代码是否符合一些固定的规范。具体介绍大家可以百度一下进行普及。

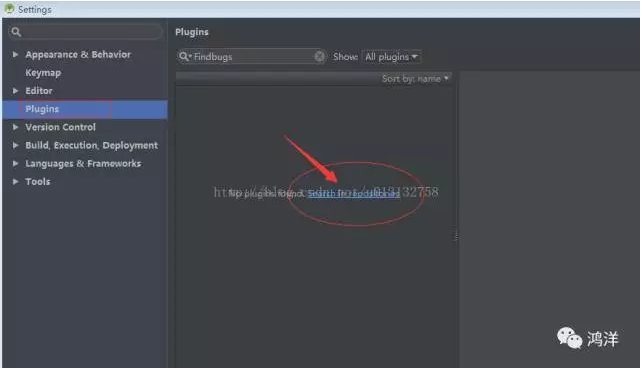
2

Findbugs安装教程

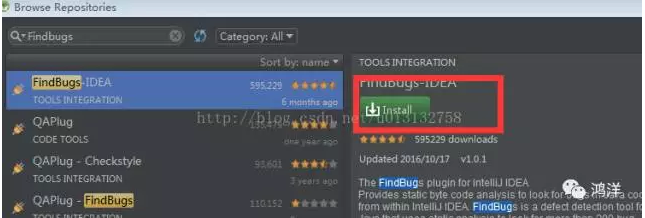
AndroidStudio的Findbugs插件的安装有在线安装和离线安装，这里介绍在线安装。

Findbugs的在线安装非常简单，打开AndroidStudio 点击File->Settings->Plugins ， 具体如下图所示：

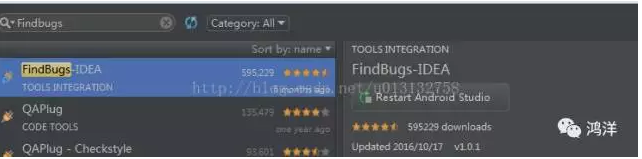
1.输入搜索Findbugs 。



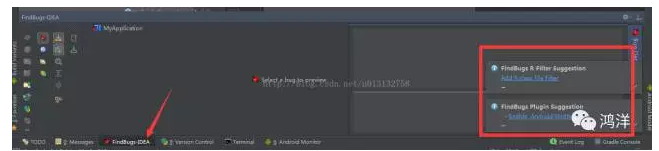
2.点击Search in respostories.



3.下载安装



4.然后重启Android Studio，会发现下面工具栏多出一个愤怒的小鸟的图标，这就是Findbugs。如下图所示，左下角弹出框提示是提示你加入你的对应的缺陷匹配规则，这个可以不用去管他。

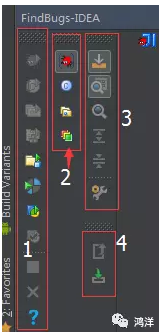


3

Findbugs的简单使用教程

Findbugs使用之前首先介绍一下左边工具栏里面的这些按钮都是干嘛的。

首先我们可以看Findbugs的按钮主要分4块，如下所示：



这4块是按照功能来区分的主要是：

1、代码检测功能区

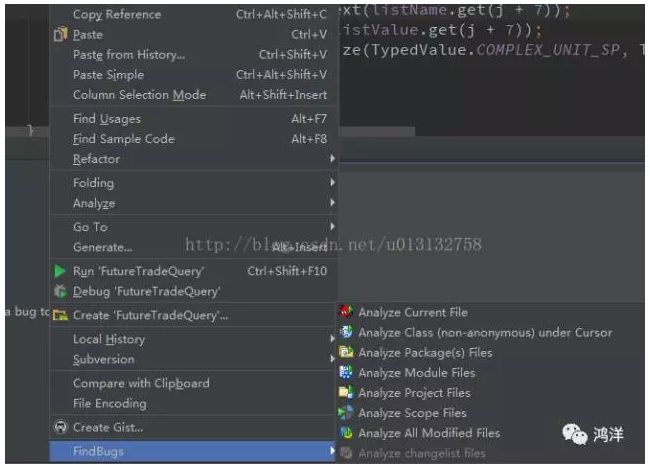
2、bug分组显示功能区

3、用户体验功能区

4、导入/出错误集合

（1）代码检测功能区

该区的按钮主要用来检测代码中的bug，我们检测或停止检测bug就是点击这个区域的按钮。这些按钮我们点击打开一个文件右击弹出菜单最下面有个Findbugs，光标移动到下面就会出现这个区域的图标。

如下图所示：  


根据上图的图标右部功能描述我们很容易明白这些按钮是干嘛的，下面我们一个个来说一下。

从上到下：

1、分析当前文件

2、分析游标下的类（就是光标所在的类）

3、分析包内所有文件

4、分析当前模块的所有文件

5、分析项目中的所有文件

这个按钮就是分析检测当前project中所有文件，找出bugs。

6、分析范围内的所有文件

这个按钮就是分析检测当前文件一定范围中所有文件，我试了一下点击后分析的还是当前项目中的所有文件，可能是因为项目结构的问题吧。

7、分析所有模块内的所有文件

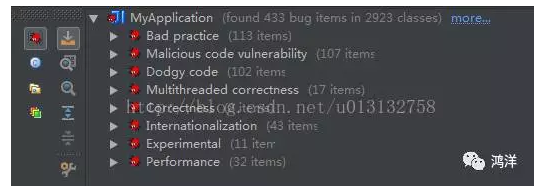
这个按钮就是检测所有模块中的代码缺陷，主要和第一个按钮功能还是差不多。

8、分析改变的所有文件

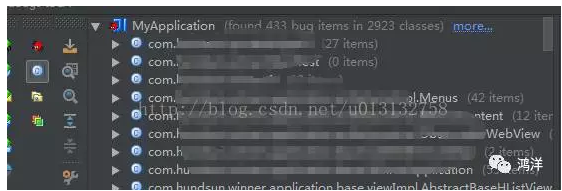
这个按钮一般显示都是暗色的点击不了，当有svn等版本控制时改变后才能点击检测。

（2）bug分组功能区

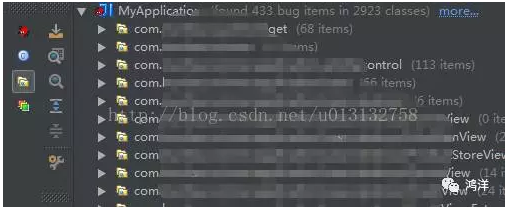
1、根据bug的分类来分组显示bug，选中后bug分组如下所示：



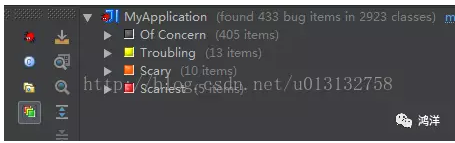
2、根据类来分组显示bug，就是一个类一个类显示bug。



3、根据包来分组显示bug

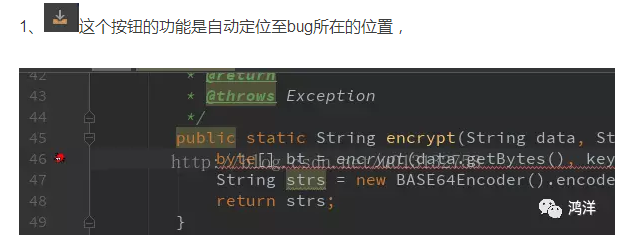


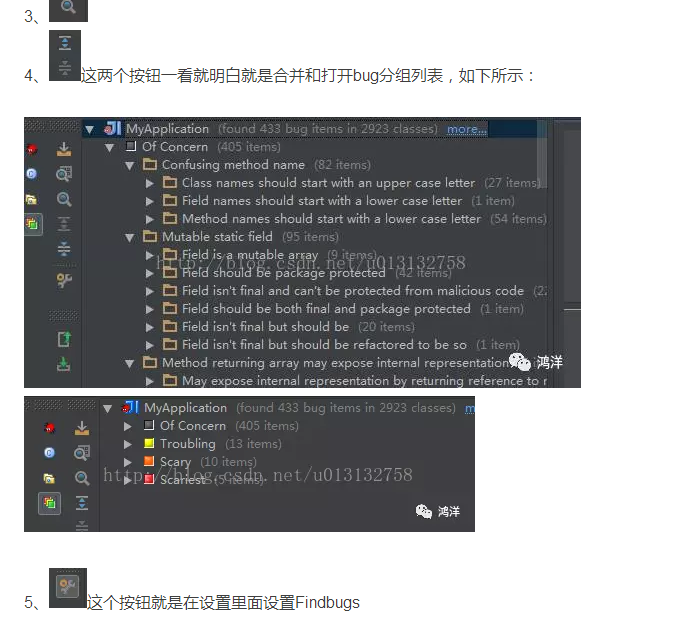
4、根据bug的等级分组显示bug，由上至下bug的等级越来越低，也就是最上面的是最致命的。



（3）用户体验帮助区

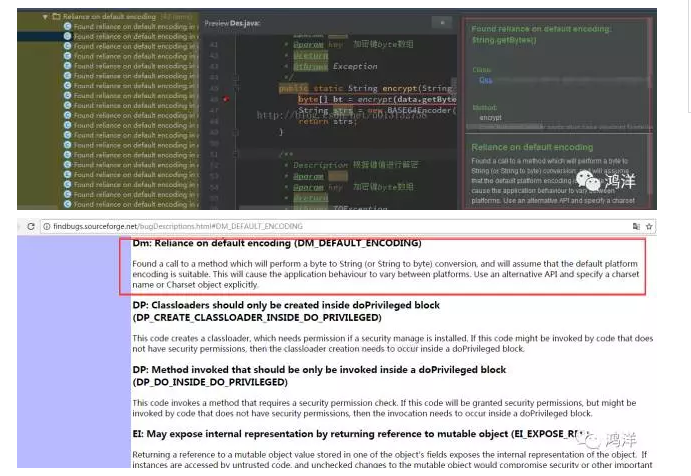
该区主要是用来帮助体验的





（5）bug修改

 检测到bug是什么bug呢？如何修改呢，我们选中一个bug会在最右边显示并描述，如下图所示，右下角绿色字体可以点击链接至Findbugs官网。



问题描述都已经很清楚了，具体如何修改大家实在不会可以百度，或者请教大佬。

推荐一篇不错的Findbugs错误规则整理的博客：

http://blog.csdn.net/jdsjlzx/article/details/21472253/

错误代码解释

rule.findbugs.IMSE\_DONT\_CATCH\_IMSE.name=不良实践 - 捕获可疑IllegalMonitorStateException  
rule.findbugs.BX\_BOXING\_IMMEDIATELY\_UNBOXED.name=性能 - 基本类型包装之后立刻解包  
rule.findbugs.IJU\_SETUP\_NO\_SUPER.name=使用错误 - TestCase定义的setUp没有调用super.setUp()  
rule.findbugs.TQ\_ALWAYS\_VALUE\_USED\_WHERE\_NEVER\_REQUIRED.name=使用错误 - 某个值使用了注解限制类型，但是这个限制永远不会发生  
rule.findbugs.TLW\_TWO\_LOCK\_WAIT.name=多线程错误 - 等待两个被持有的锁  
rule.findbugs.RV\_01\_TO\_INT.name=使用错误 - 0至1的随机数被当做整数0  
rule.findbugs.NP\_PARAMETER\_MUST\_BE\_NONNULL\_BUT\_MARKED\_AS\_NULLABLE.name=高危 - 参数必须非null但是标记为可为null  
rule.findbugs.RV\_ABSOLUTE\_VALUE\_OF\_RANDOM\_INT.name=使用错误 - 尝试计算32位随机整数的绝对值  
rule.findbugs.EC\_INCOMPATIBLE\_ARRAY\_COMPARE.name=使用错误 - 使用equals()比较不兼容的数组  
rule.findbugs.UL\_UNRELEASED\_LOCK\_EXCEPTION\_PATH.name=多线程错误 - 方法没有在所有异常路径释放锁  
rule.findbugs.SE\_NONSTATIC\_SERIALVERSIONID.name=不良实践 - serialVersionUID不是static的  
rule.findbugs.UCF\_USELESS\_CONTROL\_FLOW.name=高危 - 无用控制流  
rule.findbugs.BC\_IMPOSSIBLE\_CAST.name=使用错误 - 不可能的转换  
rule.findbugs.XSS\_REQUEST\_PARAMETER\_TO\_SEND\_ERROR.name=安全风险 - servlet的反射导致跨站脚本漏洞  
rule.findbugs.DM\_NEW\_FOR\_GETCLASS.name=性能 - 仅为了获得一个方法就创建了一个对象  
rule.findbugs.OBL\_UNSATISFIED\_OBLIGATION.name=试验 - 方法可能在清理流或资源时失败  
rule.findbugs.UW\_UNCOND\_WAIT.name=多线程错误 - 无条件等待  
rule.findbugs.DLS\_DEAD\_LOCAL\_STORE\_OF\_NULL.name=高危 - 把null设置给不会用到的局部变量  
rule.findbugs.NM\_CLASS\_NAMING\_CONVENTION.name=类名应该以大写字母开头  
rule.findbugs.RC\_REF\_COMPARISON\_BAD\_PRACTICE\_BOOLEAN.name=使用错误 - 怀疑对两个布尔值的引用进行比较  
rule.findbugs.MWN\_MISMATCHED\_NOTIFY.name=多线程错误- 不匹配的notify()  
rule.findbugs.NM\_VERY\_CONFUSING.name=错误 - 非常容易迷惑的方法名  
rule.findbugs.FI\_NULLIFY\_SUPER.name=不良实践 - 空Finalizer禁用了超类的finalizer  
rule.findbugs.MTIA\_SUSPECT\_STRUTS\_INSTANCE\_FIELD.name=高危 - 继承了struts Action的类使用了实例变量  
rule.findbugs.DM\_STRING\_CTOR.name=性能 - 方法调用了效率很低的new String(String)构造方法  
rule.findbugs.STCAL\_INVOKE\_ON\_STATIC\_DATE\_FORMAT\_INSTANCE.name=多线程错误 - 调用静态DateFormat  
rule.findbugs.NP\_NULL\_PARAM\_DEREF\_NONVIRTUAL.name=使用错误 - 非虚拟方法调用向非空参数传入了null  
rule.findbugs.FI\_EMPTY.name=不良实践 - 应该删除空的finalizer  
rule.findbugs.CD\_CIRCULAR\_DEPENDENCY.name=试验 - 类间存在循环引用  
rule.findbugs.EC\_UNRELATED\_TYPES.name=使用错误 - 使用equals()比较不同类型  
rule.findbugs.EI\_EXPOSE\_STATIC\_REP2.name=恶意代码漏洞 - 把可变对象保存到静态字段中可能会暴露内部静态状态  
rule.findbugs.DMI\_INVOKING\_TOSTRING\_ON\_ANONYMOUS\_ARRAY.name=错误 - 对数组执行toString  
rule.findbugs.SIC\_INNER\_SHOULD\_BE\_STATIC\_ANON.name=性能 - 可以重构成一个静态内部类  
rule.findbugs.STI\_INTERRUPTED\_ON\_UNKNOWNTHREAD.name=错误 - 在thread实例上调用了静态Thread.interrupted()方法  
rule.findbugs.CN\_IDIOM\_NO\_SUPER\_CALL.name=不良实践 - clone方法没有调用super.clone()  
rule.findbugs.VA\_FORMAT\_STRING\_BAD\_ARGUMENT.name=错误用法 - 格式化字符串占位符与传入的参数不匹配  
rule.findbugs.EQ\_DOESNT\_OVERRIDE\_EQUALS.name=高危 - 类没有覆盖父类的equals方法  
rule.findbugs.BC\_IMPOSSIBLE\_DOWNCAST\_OF\_TOARRAY.name=错误用法 - 集合转换为数组元素时发生的类型转换错误  
rule.findbugs.SE\_NO\_SUITABLE\_CONSTRUCTOR\_FOR\_EXTERNALIZATION.name=不良实践 - 类是可扩展的，但是没有提供无参数的构造方法  
rule.findbugs.TQ\_EXPLICIT\_UNKNOWN\_SOURCE\_VALUE\_REACHES\_ALWAYS\_SINK.name=错误用法 - 数值需要类型标示，但是却标记为未知  
rule.findbugs.SIC\_INNER\_SHOULD\_BE\_STATIC\_NEEDS\_THIS.name=性能 - 可以筹够成一个静态内部类  
rule.findbugs.EQ\_CHECK\_FOR\_OPERAND\_NOT\_COMPATIBLE\_WITH\_THIS.name=不良实践 - equals检测不兼容的参数操作  
rule.findbugs.RV\_RETURN\_VALUE\_OF\_PUTIFABSENT\_IGNORED.name=错误用法 - 忽略了putIfAbsent的返回值，传递给putIfAbsent的值被重用  
rule.findbugs.STCAL\_INVOKE\_ON\_STATIC\_CALENDAR\_INSTANCE.name=多线程错误 - 调用静态Calendar  
rule.findbugs.MS\_CANNOT\_BE\_FINAL.name=恶意代码漏洞 - 字段不是final的，不能防止恶意代码的攻击  
rule.findbugs.IS\_INCONSISTENT\_SYNC.name=多线程错误 - 不一致的同步  
rule.findbugs.SE\_NO\_SERIALVERSIONID.name=不良实践 - 类是可序列化的，但是没有定义serialVersionUID  
rule.findbugs.EI\_EXPOSE\_REP2.name=恶意代码漏洞 - 可能暴露内部实现，通过与可变对象引用协作  
rule.findbugs.NM\_METHOD\_CONSTRUCTOR\_CONFUSION.name=错误用法 - 明显的方法/构造方法混淆  
rule.findbugs.ICAST\_INTEGER\_MULTIPLY\_CAST\_TO\_LONG.name=高危 - 整形乘法的结果转换为long型  
rule.findbugs.QF\_QUESTIONABLE\_FOR\_LOOP.name=高危 - for循环中存在复杂，微妙或者错误的自增  
rule.findbugs.DLS\_DEAD\_STORE\_OF\_CLASS\_LITERAL.name=错误用法 - 类中保存了无用字符  
rule.findbugs.NM\_FUTURE\_KEYWORD\_USED\_AS\_MEMBER\_IDENTIFIER.name=不良实践 - 使用了未来java版本中成为关键字的标识  
rule.findbugs.BC\_VACUOUS\_INSTANCEOF.name=高危 - instanceof会一直返回true  
rule.findbugs.INT\_VACUOUS\_BIT\_OPERATION.name=高危 - 在整形上进行位操作时有一些位上出现空洞  
rule.findbugs.NP\_NULL\_INSTANCEOF.name=错误用法 - 一个已知的null值被检测它是否是一个类型的实例  
rule.findbugs.SIC\_THREADLOCAL\_DEADLY\_EMBRACE.name=错误用法 - 非静态内部类和ThreadLocal的致命结合  
rule.findbugs.EQ\_UNUSUAL.name=高危 - 罕见的equals方法  
rule.findbugs.IJU\_NO\_TESTS.name=错误用法 - TestCase没有任何测试  
rule.findbugs.EQ\_OVERRIDING\_EQUALS\_NOT\_SYMMETRIC.name=错误用法 - equals方法覆盖了父类的equals可能功能不符  
rule.findbugs.XFB\_XML\_FACTORY\_BYPASS.name=高危 - 方法直接调用了xml接口的一个具体实现  
rule.findbugs.SWL\_SLEEP\_WITH\_LOCK\_HELD.name=多线程错误 - 方法在获得锁时调用了Thread.sleep()  
rule.findbugs.CN\_IDIOM.name=不良实践 - 类实现了Cloneable ，但是没有定义或使用clone方法  
rule.findbugs.WA\_AWAIT\_NOT\_IN\_LOOP.name=多线程错误 - 未在循环中使用的Condition.await()  
rule.findbugs.DM\_FP\_NUMBER\_CTOR.name=性能 - 方法调用了低效的浮点书构造方法；应该使用静态的valueOf代替  
rule.findbugs.SF\_SWITCH\_NO\_DEFAULT.name=Switch语句中没有包含default  
rule.findbugs.NP\_NULL\_ON\_SOME\_PATH\_FROM\_RETURN\_VALUE.name=高危 - 调用返回返回值可能出现null值  
rule.findbugs.NP\_CLONE\_COULD\_RETURN\_NULL.name=不良实践 - Clone方法可能返回null  
rule.findbugs.MS\_OOI\_PKGPROTECT.name=恶意代码漏洞 - 属性应该从接口中移除并将访问权限设置为包保护  
rule.findbugs.DM\_BOXED\_PRIMITIVE\_TOSTRING.name=性能 - 方法使用了装箱的基本类型只为了调用toString  
rule.findbugs.EQ\_ABSTRACT\_SELF.name=不良实践 - 抽象类定义了协变的equals方法  
rule.findbugs.DM\_STRING\_TOSTRING.name=性能 - 方法调用了String的toString()方法  
rule.findbugs.SE\_METHOD\_MUST\_BE\_PRIVATE.name=错误用法 - 方法必须是private的为了让序列化正常工作  
rule.findbugs.DL\_SYNCHRONIZATION\_ON\_BOOLEAN.name=多线程错误 - 在Boolean上使用同步可能导致死锁  
rule.findbugs.UWF\_UNWRITTEN\_FIELD.name=错误用法 - 未赋值属性  
rule.findbugs.IS2\_INCONSISTENT\_SYNC.name=多线程错误 - 不一致的同步  
rule.findbugs.IM\_AVERAGE\_COMPUTATION\_COULD\_OVERFLOW.name=高危 - 计算平均值可能溢出  
rule.findbugs.BIT\_SIGNED\_CHECK\_HIGH\_BIT.name=错误用法 - 检查位运算的符号  
rule.findbugs.FL\_MATH\_USING\_FLOAT\_PRECISION.name=错误用法 - 方法进行数学运算时使用了浮点数的精度  
rule.findbugs.WS\_WRITEOBJECT\_SYNC.name=多线程错误 - 类的writeObject()方法是同步的，但是没有做其他事情  
rule.findbugs.RV\_RETURN\_VALUE\_IGNORED.name=错误用法 - 方法忽略了返回值  
rule.findbugs.SQL\_NONCONSTANT\_STRING\_PASSED\_TO\_EXECUTE.name=安全风险 - 非常量的字符串传递给方法执行SQL语句  
rule.findbugs.JCIP\_FIELD\_ISNT\_FINAL\_IN\_IMMUTABLE\_CLASS.name=不良实践 - 不可变的类的属性应该是final  
rule.findbugs.AM\_CREATES\_EMPTY\_ZIP\_FILE\_ENTRY.name=不良实践 - 创建了一个空的zip文件的入口  
rule.findbugs.DM\_NEXTINT\_VIA\_NEXTDOUBLE.name=性能 - 使用Random的nextInt方法来获得一个随机整数，而不是nextDouble  
rule.findbugs.UI\_INHERITANCE\_UNSAFE\_GETRESOURCE.name=不良实践 - 如果类被扩展，GetResource的使用可能就是不安全的  
rule.findbugs.SIO\_SUPERFLUOUS\_INSTANCEOF.name=错误用法 - 不必要的类型检测使用instanceof操作符  
rule.findbugs.EQ\_OTHER\_NO\_OBJECT.name=错误用法 - equals()方法定义，但是没有覆盖equals(Object)  
rule.findbugs.USM\_USELESS\_ABSTRACT\_METHOD.name=试验 - 抽象方法已经在实现的接口中定义了  
rule.findbugs.MTIA\_SUSPECT\_SERVLET\_INSTANCE\_FIELD.name=高危 - 扩展Servlet的类使用了实例变量  
rule.findbugs.DM\_USELESS\_THREAD.name=多线程错误 - 使用默认的空run方法创建了一个线程  
rule.findbugs.ML\_SYNC\_ON\_UPDATED\_FIELD.name=多线程错误 - 方法在一个修改了的属性上进行了同步  
rule.findbugs.CO\_SELF\_NO\_OBJECT.name=不良实践 - 协变的compareTo()定义  
rule.findbugs.BC\_UNCONFIRMED\_CAST.name=高危 - 未检查/未证实的类型转换  
rule.findbugs.FI\_FINALIZER\_NULLS\_FIELDS.name=不良实践 - Finalizer空属性  
rule.findbugs.BIT\_AND.name=错误用法 - 不兼容的位掩码(BIT\_AND)  
rule.findbugs.FE\_FLOATING\_POINT\_EQUALITY.name=高危 - 测试浮点数相等  
rule.findbugs.TQ\_EXPLICIT\_UNKNOWN\_SOURCE\_VALUE\_REACHES\_NEVER\_SINK.name=错误用法 - 值不要求有类型标示，但是标记为未知  
rule.findbugs.NP\_NULL\_PARAM\_DEREF.name=错误用法 - 方法调用把null传递给一个非null参数  
rule.findbugs.FB\_MISSING\_EXPECTED\_WARNING.name=试验 - findbugs丢失了期待或需要的警告  
rule.findbugs.DMI\_INVOKING\_HASHCODE\_ON\_ARRAY.name=错误用法 - 在数组上调用了hashCode  
rule.findbugs.QBA\_QUESTIONABLE\_BOOLEAN\_ASSIGNMENT.name=错误用法 - 方法在布尔表达式中分配了boolean文字  
rule.findbugs.SA\_FIELD\_SELF\_COMPARISON.name=错误用法 - 属性自己与自己进行了比较  
rule.findbugs.UR\_UNINIT\_READ\_CALLED\_FROM\_SUPER\_CONSTRUCTOR.name=错误用法 - 父类的构造方法调用未初始化属性的方法  
rule.findbugs.ES\_COMPARING\_PARAMETER\_STRING\_WITH\_EQ.name=不良实践 - 比较字符串参数使用了 == 或 !=  
rule.findbugs.INT\_BAD\_COMPARISON\_WITH\_NONNEGATIVE\_VALUE.name=错误用法 - 错误比较非负值与负数  
rule.findbugs.INT\_BAD\_COMPARISON\_WITH\_SIGNED\_BYTE.name=错误用法 - 错误比较带符号的byte  
rule.findbugs.IO\_APPENDING\_TO\_OBJECT\_OUTPUT\_STREAM.name=错误用法 - 尝试向一个对象输出流添加信息  
rule.findbugs.FI\_MISSING\_SUPER\_CALL.name=不良实践 - Finalizer没有调用父类的finalizer  
rule.findbugs.VA\_FORMAT\_STRING\_EXTRA\_ARGUMENTS\_PASSED.name=错误用法 - 传递了多余实际使用的格式化字符串的参数  
rule.findbugs.HE\_EQUALS\_USE\_HASHCODE.name=不良实践 - 类定义了equals()，但使用了Object.hashCode()  
rule.findbugs.IJU\_BAD\_SUITE\_METHOD.name=错误用法 - TestCase声明了一个错误的suite方法  
rule.findbugs.DMI\_CONSTANT\_DB\_PASSWORD.name=安全风险 - 硬编码了数据库密码  
rule.findbugs.REC\_CATCH\_EXCEPTION.name=高危 - 捕获了没有抛出的异常  
rule.findbugs.PS\_PUBLIC\_SEMAPHORES.name=高危 - 类在公用接口中暴露了同步和信号  
rule.findbugs.EC\_UNRELATED\_INTERFACES.name=错误用法 - 调用equals()比较不同的接口类型  
rule.findbugs.UCF\_USELESS\_CONTROL\_FLOW\_NEXT\_LINE.name=错误用法 - 执行到下一行的无用流程控制  
rule.findbugs.LG\_LOST\_LOGGER\_DUE\_TO\_WEAK\_REFERENCE.name=试验 - OpenJDK中存在潜在的丢失logger的风险，因为弱引用  
rule.findbugs.NP\_UNWRITTEN\_FIELD.name=错误用法 - 读取未初始化的属性  
rule.findbugs.DMI\_UNSUPPORTED\_METHOD.name=高危 - 调用不支持的方法  
rule.findbugs.RCN\_REDUNDANT\_COMPARISON\_OF\_NULL\_AND\_NONNULL\_VALUE.name=高危 - 重复比较非空值和null  
rule.findbugs.EC\_BAD\_ARRAY\_COMPARE.name=错误用法 - 调用equals()，与==效果一样  
rule.findbugs.EI\_EXPOSE\_REP.name=恶意代码漏洞 - 可能通过返回一个可变对象的引用暴露了内部实现  
rule.findbugs.NP\_DEREFERENCE\_OF\_READLINE\_VALUE.name=高危 - 没有判断readLine()的结果是否为空  
rule.findbugs.UPM\_UNCALLED\_PRIVATE\_METHOD.name=性能 - 从未用到的私有方法  
rule.findbugs.NP\_NULL\_ON\_SOME\_PATH.name=错误用法 - 可能出现空指针引用  
rule.findbugs.NP\_EQUALS\_SHOULD\_HANDLE\_NULL\_ARGUMENT.name=不良实践 - equals()方法没有检测null参数  
rule.findbugs.EC\_NULL\_ARG.name=错误用法 - 使用空参数调用equals()  
rule.findbugs.SE\_BAD\_FIELD\_STORE.name=不良实践 - 非序列化值保存在序列化类的实例变量中  
rule.findbugs.VO\_VOLATILE\_REFERENCE\_TO\_ARRAY.name=多线程错误 - 数组的volatile引用不会把数组元素也当做volatile来引用  
rule.findbugs.NP\_SYNC\_AND\_NULL\_CHECK\_FIELD.name=多线程错误 - 同步和空值检测发生在同一个属性上  
rule.findbugs.DM\_EXIT.name=不良实践 - 方法调用了System.exit(...)  
rule.findbugs.RC\_REF\_COMPARISON.name=不良实践 - 怀疑进行了引用比较  
rule.findbugs.SE\_NO\_SUITABLE\_CONSTRUCTOR.name=不良实践 - 类是可序列化的，但是父类没有定义无参数构造方法  
rule.findbugs.DC\_DOUBLECHECK.name=多线程错误 - 可能对属性进行了双重检测  
rule.findbugs.DMI\_LONG\_BITS\_TO\_DOUBLE\_INVOKED\_ON\_INT.name=错误用法 - 在int上调用了Double.longBitsToDouble  
rule.findbugs.RpC\_REPEATED\_CONDITIONAL\_TEST.name=错误用法 - 重复判断条件  
rule.findbugs.WMI\_WRONG\_MAP\_ITERATOR.name=性能 - keySet迭代是低效的，使用entrySet代替  
rule.findbugs.DLS\_DEAD\_LOCAL\_STORE.name=高危 - 未用的局部变量  
rule.findbugs.INT\_BAD\_REM\_BY\_1.name=错误用法 - 整数剩余模1  
rule.findbugs.RV\_RETURN\_VALUE\_IGNORED\_BAD\_PRACTICE.name=不良实践 - 方法忽略异常返回值  
rule.findbugs.SA\_LOCAL\_SELF\_ASSIGNMENT.name=高危 - 局部变量的自我赋值  
rule.findbugs.MS\_SHOULD\_BE\_FINAL.name=恶意代码漏洞 - 属性不是final，但是应该设置成final  
rule.findbugs.SIC\_INNER\_SHOULD\_BE\_STATIC.name=性能 - 应该是一个静态内部类  
rule.findbugs.NP\_GUARANTEED\_DEREF.name=错误用法 - null值一定会被调用  
rule.findbugs.SE\_READ\_RESOLVE\_MUST\_RETURN\_OBJECT.name=不良实践 - readResolve方法必须返回Object  
rule.findbugs.NP\_LOAD\_OF\_KNOWN\_NULL\_VALUE.name=高危 - 加载了已知的null值  
rule.findbugs.BX\_BOXING\_IMMEDIATELY\_UNBOXED\_TO\_PERFORM\_COERCION.name=性能 - 基本数据被装箱又被拆箱  
rule.findbugs.CN\_IMPLEMENTS\_CLONE\_BUT\_NOT\_CLONEABLE.name=不良实践 - 类定义了clone()但没有实现Cloneable  
rule.findbugs.CO\_ABSTRACT\_SELF.name=不良实践 - 抽象类定义了协变的compareTo()方法  
rule.findbugs.BAC\_BAD\_APPLET\_CONSTRUCTOR.name=试验 - 错误的Applet构造方法依赖未初始化的AppletStub  
rule.findbugs.EQ\_GETCLASS\_AND\_CLASS\_CONSTANT.name=不良实践 - equals方法因为子类失败  
rule.findbugs.DB\_DUPLICATE\_SWITCH\_CLAUSES.name=高危 - 在两个switch语句中使用了相同的代码  
rule.findbugs.DB\_DUPLICATE\_BRANCHES.name=高危 - 在两个分支中使用了相同的代码  
rule.findbugs.UOE\_USE\_OBJECT\_EQUALS.name=试验 - 在final类上调用了equals，但是没有覆盖Object的equals方法  
rule.findbugs.FI\_USELESS.name=不良实践 - Finalizer除了调用父类的finalizer以外什么也没做  
rule.findbugs.NP\_ALWAYS\_NULL.name=错误用法 - 调用了null指针  
rule.findbugs.DMI\_VACUOUS\_SELF\_COLLECTION\_CALL.name=错误用法 - 集合的调用不能被感知  
rule.findbugs.DLS\_DEAD\_LOCAL\_STORE\_IN\_RETURN.name=错误用法 - 返回语句中的无用的赋值  
rule.findbugs.IJU\_ASSERT\_METHOD\_INVOKED\_FROM\_RUN\_METHOD.name=错误用法 - 在run方法中的JUnit检验不能报告给JUnit  
rule.findbugs.DMI\_EMPTY\_DB\_PASSWORD.name=安全风险 - 空的数据库密码  
rule.findbugs.DM\_BOOLEAN\_CTOR.name=性能 - 方法调用了低效的Boolean构造方法；使用Boolean.valueOf(...)代替  
rule.findbugs.BC\_IMPOSSIBLE\_DOWNCAST.name=错误用法 - 不可能转型  
rule.findbugs.BC\_EQUALS\_METHOD\_SHOULD\_WORK\_FOR\_ALL\_OBJECTS.name=不良实践 - Equals方法不应该假设任何有关参数类型的事宜  
rule.findbugs.RV\_EXCEPTION\_NOT\_THROWN.name=错误用法 - 异常创建后就丢弃了，没有抛出  
rule.findbugs.VA\_PRIMITIVE\_ARRAY\_PASSED\_TO\_OBJECT\_VARARG.name=错误用法 - 基本类型数组传递给一个期待可变对象类型参数的方法  
rule.findbugs.LI\_LAZY\_INIT\_UPDATE\_STATIC.name=多线程错误 - 错误的延迟初始化和更新静态属性  
rule.findbugs.SA\_FIELD\_SELF\_ASSIGNMENT.name=错误用法 - 属性自身赋值  
rule.findbugs.EQ\_ALWAYS\_FALSE.name=错误用法 - equals方法一直返回false  
rule.findbugs.DMI\_RANDOM\_USED\_ONLY\_ONCE.name=不良实践 - Random对象创建后只用了一次  
rule.findbugs.NM\_CLASS\_NOT\_EXCEPTION.name=不良实践 - Class没有继承Exception，虽然名字像一个异常  
rule.findbugs.SA\_LOCAL\_DOUBLE\_ASSIGNMENT.name=高危 - 给局部变量双重赋值  
rule.findbugs.NP\_NULL\_PARAM\_DEREF\_ALL\_TARGETS\_DANGEROUS.name=错误用法 - 方法调用传递null给非空参数 (ALL\_TARGETS\_DANGEROUS)  
rule.findbugs.NP\_TOSTRING\_COULD\_RETURN\_NULL.name=不良实践 - toString方法可能返回null  
rule.findbugs.BC\_BAD\_CAST\_TO\_ABSTRACT\_COLLECTION.name=高危 - 转换成抽象集合值得怀疑  
rule.findbugs.NM\_LCASE\_HASHCODE.name=类定义了hashcode(); 应该是hashCode()吧？  
rule.findbugs.RU\_INVOKE\_RUN.name=多线程错误 - 在线程中调用了run（你的意思是再启动一次么？）  
rule.findbugs.DMI\_INVOKING\_TOSTRING\_ON\_ARRAY.name=错误用法 - 调用了数组的toString  
rule.findbugs.NM\_METHOD\_NAMING\_CONVENTION.name=方法名应该以小写字母开头  
rule.findbugs.RCN\_REDUNDANT\_COMPARISON\_TWO\_NULL\_VALUES.name=高危 - 重复比较两个null值  
rule.findbugs.SA\_LOCAL\_SELF\_COMPUTATION.name=错误用法 - 对一个变量进行无意义的自我计算 (比如 x & x)  
rule.findbugs.MS\_MUTABLE\_HASHTABLE.name=恶意代码漏洞 - 属性是可变的Hashtable  
rule.findbugs.RV\_DONT\_JUST\_NULL\_CHECK\_READLINE.name=高危 - 方法丢掉了readLine的结果，在检测它是非空之后。  
rule.findbugs.ES\_COMPARING\_STRINGS\_WITH\_EQ.name=不良实践 - 使用== 或 !=比较String  
rule.findbugs.DL\_SYNCHRONIZATION\_ON\_SHARED\_CONSTANT.name=多线程错误 - 同步内部String可能导致死锁  
rule.findbugs.MF\_METHOD\_MASKS\_FIELD.name=错误用法 - 方法定义了一个变量，与实例变量同名。  
rule.findbugs.EQ\_SELF\_USE\_OBJECT.name=错误用法 - 协变equals()方法定义，继承了Object.equals(Object)  
rule.findbugs.ICAST\_INT\_CAST\_TO\_FLOAT\_PASSED\_TO\_ROUND.name=错误用法 - int值转换成float，然后传递给Math.round  
rule.findbugs.GC\_UNRELATED\_TYPES.name=错误用法 - 泛型参数与方法参数没有相互关系  
rule.findbugs.BC\_IMPOSSIBLE\_INSTANCEOF.name=错误用法 - instanceof一直返回false  
rule.findbugs.SBSC\_USE\_STRINGBUFFER\_CONCATENATION.name=性能 - 方法在循环中使用+进行字符串拼接  
rule.findbugs.ICAST\_INT\_CAST\_TO\_DOUBLE\_PASSED\_TO\_CEIL.name=错误用法 - int转换成double，然后传递给Math.ceil  
rule.findbugs.UG\_SYNC\_SET\_UNSYNC\_GET.name=多线程错误 - 未同步的getter方法，同步的setter方法  
rule.findbugs.RE\_BAD\_SYNTAX\_FOR\_REGULAR\_EXPRESSION.name=错误用法 - 非法的正则表达式  
rule.findbugs.SA\_FIELD\_SELF\_COMPUTATION.name=错误用法 - 无意义的自我计算 (比如 x & x)  
rule.findbugs.DMI\_SCHEDULED\_THREAD\_POOL\_EXECUTOR\_WITH\_ZERO\_CORE\_THREADS.name=错误用法 - 创建了没有任何线程的ScheduledThreadPoolExecutor  
rule.findbugs.DMI\_USELESS\_SUBSTRING.name=高危 - 调用substring(0)会返回原值  
rule.findbugs.IA\_AMBIGUOUS\_INVOCATION\_OF\_INHERITED\_OR\_OUTER\_METHOD.name=高危 - 被继承的或外部的方法的模棱两可的调用  
rule.findbugs.OS\_OPEN\_STREAM.name=不良实践 - 方法可能在关闭流时失败  
rule.findbugs.HE\_INHERITS\_EQUALS\_USE\_HASHCODE.name=不良实践 - 类继承了equals()，但使用了Object.hashCode()  
rule.findbugs.SE\_NONFINAL\_SERIALVERSIONID.name=不良实践 - serialVersionUID不是final的  
rule.findbugs.EQ\_SELF\_NO\_OBJECT.name=不良实践 - 协变equals()方法定义  
rule.findbugs.SF\_DEAD\_STORE\_DUE\_TO\_SWITCH\_FALLTHROUGH.name=由于switch语句导致的无用存储  
rule.findbugs.SW\_SWING\_METHODS\_INVOKED\_IN\_SWING\_THREAD.name=不良实践 - 某些swing方法需要在swing线程中调用  
rule.findbugs.VA\_FORMAT\_STRING\_ILLEGAL.name=错误用法 - 非法的格式化字符串  
rule.findbugs.DM\_NUMBER\_CTOR.name=性能 - 方法调用低效的数字构造方法;使用静态valueOf代替  
rule.findbugs.RV\_REM\_OF\_RANDOM\_INT.name=高危 - 其余32位带符号随机整数  
rule.findbugs.EQ\_COMPARING\_CLASS\_NAMES.name=错误用法 - equals方法比较类名而不是比较类  
rule.findbugs.ST\_WRITE\_TO\_STATIC\_FROM\_INSTANCE\_METHOD.name=高危 - 通过一个实例方法更新静态属性  
rule.findbugs.TQ\_NEVER\_VALUE\_USED\_WHERE\_ALWAYS\_REQUIRED.name=错误用法 - 某个值使用了注解标记不会限制类型，但是这个限制是必须的  
rule.findbugs.NS\_NON\_SHORT\_CIRCUIT.name=高危 - 可疑的非短路逻辑  
rule.findbugs.VA\_FORMAT\_STRING\_NO\_PREVIOUS\_ARGUMENT.name=错误用法 - 格式化字符串没有前面的参数  
rule.findbugs.SE\_PRIVATE\_READ\_RESOLVE\_NOT\_INHERITED.name=高危 - 私有readResolve方法不是继承自父类  
rule.findbugs.RCN\_REDUNDANT\_NULLCHECK\_OF\_NONNULL\_VALUE.name=高危 - 对一个已知不是null的值重复进行空值判断  
rule.findbugs.NM\_LCASE\_TOSTRING.name=类定义了tostring(); 应该是toString()吧？  
rule.findbugs.HSC\_HUGE\_SHARED\_STRING\_CONSTANT.name=性能 - 巨大的字符串常量在多个类文件中间重复了  
rule.findbugs.SE\_TRANSIENT\_FIELD\_NOT\_RESTORED.name=不良实践 - Transient属性没有再反序列化时被设置  
rule.findbugs.JLM\_JSR166\_LOCK\_MONITORENTER.name=多线程错误 - 在java.util.concurrent Lock上进行了同步  
rule.findbugs.EQ\_ALWAYS\_TRUE.name=错误用法 - equals方法总是返回true  
rule.findbugs.ISC\_INSTANTIATE\_STATIC\_CLASS.name=不良实践 - 只提供静态方法的类不需要实例化  
rule.findbugs.ICAST\_IDIV\_CAST\_TO\_DOUBLE.name=高危 - int相除的结果转换成double或float  
rule.findbugs.RC\_REF\_COMPARISON\_BAD\_PRACTICE.name=错误用法 - 可疑的引用与常量的比较  
rule.findbugs.FI\_EXPLICIT\_INVOCATION.name=不良实践 - 直接调用finalizer  
rule.findbugs.ESync\_EMPTY\_SYNC.name=多线程错误 - 空的同步块  
rule.findbugs.DMI\_USING\_REMOVEALL\_TO\_CLEAR\_COLLECTION.name=错误用法 - 不要使用removeAll清空集合  
rule.findbugs.SE\_BAD\_FIELD.name=在序列化类中出现了非transient也非serializable的实例属性  
rule.findbugs.NP\_STORE\_INTO\_NONNULL\_FIELD.name=错误用法 - 属性已经标记为NonNull，但是设置了一个null值  
rule.findbugs.IT\_NO\_SUCH\_ELEMENT.name=不良实践 - 迭代器的next()方法不能抛出NoSuchElementException  
rule.findbugs.HRS\_REQUEST\_PARAMETER\_TO\_HTTP\_HEADER.name=安全风险 - HTTP响应拆分漏洞  
rule.findbugs.DMI\_THREAD\_PASSED\_WHERE\_RUNNABLE\_EXPECTED.name=高危 - 希望传入Runnable的地方传入了一个Thread  
rule.findbugs.NP\_GUARANTEED\_DEREF\_ON\_EXCEPTION\_PATH.name=错误用法 - null值会在exception处理中被用到  
rule.findbugs.RCN\_REDUNDANT\_NULLCHECK\_OF\_NULL\_VALUE.name=高危 - 已知的null值被重复的检测  
rule.findbugs.DMI\_CALLING\_NEXT\_FROM\_HASNEXT.name=错误用法 - hasNext方法调用了next方法  
rule.findbugs.HE\_HASHCODE\_USE\_OBJECT\_EQUALS.name=不良实践 - 定义了hashCode()的类使用了Object.equals()  
rule.findbugs.VA\_FORMAT\_STRING\_EXPECTED\_MESSAGE\_FORMAT\_SUPPLIED.name=错误用法 - 需要使用printf样式时使用了MessageFormat  
rule.findbugs.NP\_BOOLEAN\_RETURN\_NULL.name=不良实践 - 方法返回boolean类型返回了null  
rule.findbugs.RI\_REDUNDANT\_INTERFACES.name=高危 - 类实现了父类一样的接口  
rule.findbugs.DL\_SYNCHRONIZATION\_ON\_UNSHARED\_BOXED\_PRIMITIVE.name=多线程错误 - 在装箱的基本属性上使用了同步  
rule.findbugs.STCAL\_STATIC\_CALENDAR\_INSTANCE.name=多线程错误 - 静态Calendar  
rule.findbugs.RR\_NOT\_CHECKED.name=不良实践 - 方法忽略InputStream.read()的返回值  
rule.findbugs.IL\_INFINITE\_RECURSIVE\_LOOP.name=错误用法 - 明显的无限递归循环  
rule.findbugs.DMI\_NONSERIALIZABLE\_OBJECT\_WRITTEN.name=高危 - 非持久化的对象写入了ObjectOutput  
rule.findbugs.GC\_UNCHECKED\_TYPE\_IN\_GENERIC\_CALL.name=不良实践 - 泛型调用中使用了未检查的类型  
rule.findbugs.IMA\_INEFFICIENT\_MEMBER\_ACCESS.name=试验 - 方法访问了一个拥有的类的私有成员  
rule.findbugs.FI\_PUBLIC\_SHOULD\_BE\_PROTECTED.name=恶意代码漏洞 - Finalizer应该是protected，不能是public  
rule.findbugs.RV\_CHECK\_FOR\_POSITIVE\_INDEXOF.name=高危 - 方法检查检查String.indexOf的结果是否为正数  
rule.findbugs.ICAST\_QUESTIONABLE\_UNSIGNED\_RIGHT\_SHIFT.name=高危 - 无符号右移转换为short/byte  
rule.findbugs.DM\_STRING\_VOID\_CTOR.name=性能 - 方法调用了低效的new String()构造方法  
rule.findbugs.RE\_POSSIBLE\_UNINTENDED\_PATTERN.name=错误用法 - "."用作正则表达式  
rule.findbugs.WL\_USING\_GETCLASS\_RATHER\_THAN\_CLASS\_LITERAL.name=多线程错误 - 在getClass上使用同步而不是在class文字上  
rule.findbugs.ICAST\_BAD\_SHIFT\_AMOUNT.name=错误用法 - 右移的数值不在0..31范围内  
rule.findbugs.SF\_SWITCH\_FALLTHROUGH.name=Switch语句中一个case失败以后进入了下一个case  
rule.findbugs.DP\_DO\_INSIDE\_DO\_PRIVILEGED.name=不良实践 - 方法调用应该在doPrivileged块中  
rule.findbugs.NO\_NOTIFY\_NOT\_NOTIFYALL.name=多线程错误 - 使用notify()而不是notifyAll()  
rule.findbugs.SS\_SHOULD\_BE\_STATIC.name=性能 - 未读取的属性：这个属性是否应该是static的？  
rule.findbugs.DM\_RUN\_FINALIZERS\_ON\_EXIT.name=不良实践 - 方法调用了危险的runFinalizersOnExit方法  
rule.findbugs.MS\_FINAL\_PKGPROTECT.name=恶意代码漏洞 - 属性应该同时是final 和 package protected的  
rule.findbugs.BC\_BAD\_CAST\_TO\_CONCRETE\_COLLECTION.name=高危 - 转换成具体集合可能有问题  
rule.findbugs.BIT\_IOR\_OF\_SIGNED\_BYTE.name=错误用法 - 在带符号的byte值上进行位OR运算  
rule.findbugs.MSF\_MUTABLE\_SERVLET\_FIELD.name=多线程错误 - 可变的servlet属性  
rule.findbugs.SE\_BAD\_FIELD\_INNER\_CLASS.name=不良实践 - 非serializable类有一个可序列化的内部类  
rule.findbugs.BIT\_ADD\_OF\_SIGNED\_BYTE.name=错误用法 - 在带符号的byte值上进行位add运算  
rule.findbugs.FI\_FINALIZER\_ONLY\_NULLS\_FIELDS.name=不良实践 - Finalizer中子还有null属性  
rule.findbugs.DE\_MIGHT\_IGNORE.name=不良实践 - 方法可能忽略异常  
rule.findbugs.XSS\_REQUEST\_PARAMETER\_TO\_SERVLET\_WRITER.name=安全风险 - Servlet反射跨域脚本漏洞  
rule.findbugs.ML\_SYNC\_ON\_FIELD\_TO\_GUARD\_CHANGING\_THAT\_FIELD.name=多线程错误 - 使用同步的属性可能发生改变  
rule.findbugs.SQL\_BAD\_PREPARED\_STATEMENT\_ACCESS.name=错误用法 - 方法尝试访问PreparedStatement参数的索引是0  
rule.findbugs.DM\_CONVERT\_CASE.name=国际化 - 考虑使用国际化参数型的调用方法版本  
rule.findbugs.SE\_TRANSIENT\_FIELD\_OF\_NONSERIALIZABLE\_CLASS.name=高危 - 类的transient属性不能序列化  
rule.findbugs.NN\_NAKED\_NOTIFY.name=多线程错误 - 未使用同步包裹的notify  
rule.findbugs.VA\_FORMAT\_STRING\_MISSING\_ARGUMENT.name=错误用法 - 格式化字符串引用的参数丢失  
rule.findbugs.IC\_SUPERCLASS\_USES\_SUBCLASS\_DURING\_INITIALIZATION.name=不良实践 - 父类初始化时使用了子类  
rule.findbugs.SA\_LOCAL\_SELF\_COMPARISON.name=错误用法 - 自己和自己的值比较  
rule.findbugs.IM\_BAD\_CHECK\_FOR\_ODD.name=高危 - 对奇偶判断不能用于负数  
rule.findbugs.NP\_CLOSING\_NULL.name=错误用法 - close()调用了一个永远是null的值  
rule.findbugs.XSS\_REQUEST\_PARAMETER\_TO\_JSP\_WRITER.name=安全风险 - JSP反射调用跨域脚本漏洞  
rule.findbugs.IS\_FIELD\_NOT\_GUARDED.name=多线程错误 - 属性不能保证同步访问  
rule.findbugs.DM\_GC.name=性能 - 直接调用垃圾回收，特别是在性能测试中的可疑代码  
rule.findbugs.IM\_MULTIPLYING\_RESULT\_OF\_IREM.name=错误用法 - 整数乘法的结果是整数  
rule.findbugs.SE\_COMPARATOR\_SHOULD\_BE\_SERIALIZABLE.name=不良实践 - Comparator没有实现Serializable  
rule.findbugs.HE\_HASHCODE\_NO\_EQUALS.name=不良实践 - 类定义了hashCode()但是没定义equals()  
rule.findbugs.MF\_CLASS\_MASKS\_FIELD.name=错误用法 - 类定义了属性覆盖了父类的属性  
rule.findbugs.NM\_VERY\_CONFUSING\_INTENTIONAL.name=不良实践 - 非常迷惑的方法名称（可能是内部方法）  
rule.findbugs.SR\_NOT\_CHECKED.name=不良实践 - 方法忽略了InputStream.skip()的结果  
rule.findbugs.LI\_LAZY\_INIT\_STATIC.name=多线程错误 - 错误的对static属性进行了延迟初始化  
rule.findbugs.DMI\_ANNOTATION\_IS\_NOT\_VISIBLE\_TO\_REFLECTION.name=错误用法 - 不能使用反射检测没有标记为runtime rentention的注解的存在  
rule.findbugs.ODR\_OPEN\_DATABASE\_RESOURCE\_EXCEPTION\_PATH.name=不良实践 - 方法可能在关闭database资源时因为异常失败  
rule.findbugs.BIT\_IOR.name=错误用法 - 不兼容的位掩码（BIT\_IOR）  
rule.findbugs.ODR\_OPEN\_DATABASE\_RESOURCE.name=不良实践 - 方法可能在关闭database资源时失败  
rule.findbugs.IP\_PARAMETER\_IS\_DEAD\_BUT\_OVERWRITTEN.name=错误用法 - 参数没有被使用，但是被重新赋值了  
rule.findbugs.SQL\_PREPARED\_STATEMENT\_GENERATED\_FROM\_NONCONSTANT\_STRING.name=安全风险 - 使用非常量字符串创建了一个PreparedStatement  
rule.findbugs.UUF\_UNUSED\_FIELD.name=性能 - 无用的属性  
rule.findbugs.RCN\_REDUNDANT\_NULLCHECK\_WOULD\_HAVE\_BEEN\_A\_NPE.name=错误用法 - 对一个已经使用的值进行了null检测  
rule.findbugs.EQ\_OTHER\_USE\_OBJECT.name=错误用法 - equals()方法定义没有覆盖Object.equals(Object)  
rule.findbugs.SP\_SPIN\_ON\_FIELD.name=多线程错误 - 方法对一个属性循环复制  
rule.findbugs.SI\_INSTANCE\_BEFORE\_FINALS\_ASSIGNED.name=不良实践 - 在所有静态final属性赋值之前static初始化块创建了一个实例  
rule.findbugs.NP\_ALWAYS\_NULL\_EXCEPTION.name=错误用法 - 方法的异常路径中引用了空指针  
rule.findbugs.MS\_EXPOSE\_REP.name=恶意代码漏洞 - Public static方法可能因为返回了一个数组而暴露内部实现  
rule.findbugs.VA\_FORMAT\_STRING\_BAD\_CONVERSION\_TO\_BOOLEAN.name=高危 - 非布尔参数使用%b进行了格式化  
rule.findbugs.MS\_PKGPROTECT.name=恶意代码漏洞 - 属性应该是package protected  
rule.findbugs.NP\_NONNULL\_RETURN\_VIOLATION.name=错误用法 - 方法可能返回null，但是声明了 @NonNull  
rule.findbugs.J2EE\_STORE\_OF\_NON\_SERIALIZABLE\_OBJECT\_INTO\_SESSION.name=不良实践 - 在HttpSession中保存了不可序列化的对象  
rule.findbugs.NM\_SAME\_SIMPLE\_NAME\_AS\_SUPERCLASS.name=不良实践 - 类名不应该和父类的名称相同  
rule.findbugs.DMI\_BLOCKING\_METHODS\_ON\_URL.name=性能 - URL的equals 和 hashCode 方法会被堵塞  
rule.findbugs.HE\_SIGNATURE\_DECLARES\_HASHING\_OF\_UNHASHABLE\_CLASS.name=错误用法 - 在哈希构造中声明了不能哈希的类  
rule.findbugs.UR\_UNINIT\_READ.name=错误用法 - 构造方法中对未初始化的类进行了读取  
rule.findbugs.WA\_NOT\_IN\_LOOP.name=多线程错误 - Wait不在循环中  
rule.findbugs.DMI\_FUTILE\_ATTEMPT\_TO\_CHANGE\_MAXPOOL\_SIZE\_OF\_SCHEDULED\_THREAD\_POOL\_EXECUTOR.name=错误用法 - 尝试修改ScheduledThreadPoolExecutor的最大数  
rule.findbugs.RV\_RETURN\_VALUE\_IGNORED2.name=错误用法 - 方法忽略了返回值  
rule.findbugs.NM\_FIELD\_NAMING\_CONVENTION.name=属性名应该以小写字母开头  
rule.findbugs.FB\_UNEXPECTED\_WARNING.name=试验 - 未期待/未期望的findbugs警告  
rule.findbugs.BX\_UNBOXED\_AND\_COERCED\_FOR\_TERNARY\_OPERATOR.name=错误用法 - 基础类型拆箱后用于三元表达式  
rule.findbugs.DP\_CREATE\_CLASSLOADER\_INSIDE\_DO\_PRIVILEGED.name=不良实践 - Classloaders应该只在doPrivileged块中被创建  
rule.findbugs.NM\_WRONG\_PACKAGE.name=错误用法 - 方法没有覆盖父类的方法，因为参数的包不正确  
rule.findbugs.IL\_CONTAINER\_ADDED\_TO\_ITSELF.name=错误用法 - 一个集合被添加到自身  
rule.findbugs.CI\_CONFUSED\_INHERITANCE.name=高危 - 类是final的，但是定义了protected属性  
rule.findbugs.HE\_USE\_OF\_UNHASHABLE\_CLASS.name=错误用法 - 在哈希数据结构中使用的类没有定义hashCode()方法  
rule.findbugs.IJU\_SUITE\_NOT\_STATIC.name=错误用法 - TestCase实现了非static suite方法  
rule.findbugs.RS\_READOBJECT\_SYNC.name=多线程错误 - 类的readObject()方法不是同步的  
rule.findbugs.AM\_CREATES\_EMPTY\_JAR\_FILE\_ENTRY.name=不良实践 - 创建了一个空的jar方法入口  
rule.findbugs.VA\_FORMAT\_STRING\_BAD\_CONVERSION\_FROM\_ARRAY.name=错误用法 - 使用格式化字符串对数组进行了无用的格式化  
rule.findbugs.SF\_DEAD\_STORE\_DUE\_TO\_SWITCH\_FALLTHROUGH\_TO\_THROW.name=switch语句失败抛出异常导致无用的存储  
rule.findbugs.RV\_ABSOLUTE\_VALUE\_OF\_HASHCODE.name=错误用法 - 错误的尝试计算带符号的32位hashcodde的绝对值  
rule.findbugs.EQ\_DONT\_DEFINE\_EQUALS\_FOR\_ENUM.name=错误用法 - 枚举定义了协变equals()方法定义  
rule.findbugs.SA\_FIELD\_DOUBLE\_ASSIGNMENT.name=错误用法 - 对属性进行双重赋值  
rule.findbugs.DMI\_COLLECTION\_OF\_URLS.name=性能 - URL的Maps 和 sets可能是性能大问题  
rule.findbugs.NM\_SAME\_SIMPLE\_NAME\_AS\_INTERFACE.name=不良实践 - 类名不应该和实现的接口名相同  
rule.findbugs.UWF\_FIELD\_NOT\_INITIALIZED\_IN\_CONSTRUCTOR.name=错误用法 - 构造方法没有初始化属性  
rule.findbugs.TQ\_MAYBE\_SOURCE\_VALUE\_REACHES\_NEVER\_SINK.name=错误用法 - 值可能带着一个类型修饰符，一直使用的方式是与这个类型修饰符相悖的  
rule.findbugs.SE\_NONLONG\_SERIALVERSIONID.name=不良实践 - serialVersionUID不是long型  
rule.findbugs.RV\_REM\_OF\_HASHCODE.name=高危 - hashCode的结果可能是负数negative  
rule.findbugs.NS\_DANGEROUS\_NON\_SHORT\_CIRCUIT.name=高危 - 潜在危险使用非短跳出逻辑  
rule.findbugs.USM\_USELESS\_SUBCLASS\_METHOD.name=试验 - 方法代理到父类方法  
rule.findbugs.ITA\_INEFFICIENT\_TO\_ARRAY.name=性能 - 方法使用了toArray()对空数组参数  
rule.findbugs.DM\_MONITOR\_WAIT\_ON\_CONDITION.name=多线程错误 - 根据条件监控wait()的调用  
rule.findbugs.BOA\_BADLY\_OVERRIDDEN\_ADAPTER.name=错误用法 - 类覆盖了父类中实现的方法适配器错误  
rule.findbugs.IC\_INIT\_CIRCULARITY.name=高危 - 初始化死循环  
rule.findbugs.NM\_FUTURE\_KEYWORD\_USED\_AS\_IDENTIFIER.name=不良实践 - 使用的标记是未来java版本中的关键字  
rule.findbugs.EC\_UNRELATED\_CLASS\_AND\_INTERFACE.name=错误用法 - equals()比较无关的类和接口  
rule.findbugs.TQ\_MAYBE\_SOURCE\_VALUE\_REACHES\_ALWAYS\_SINK.name=错误用法 - 值可能没有携带一个类修饰符，但是使用中会一直需要这个类修饰符  
rule.findbugs.FE\_TEST\_IF\_EQUAL\_TO\_NOT\_A\_NUMBER.name=错误用法 - 测试是否与NaN相等  
rule.findbugs.NM\_CONFUSING.name=不良实践 - 迷惑的方法名  
rule.findbugs.VA\_FORMAT\_STRING\_ARG\_MISMATCH.name=错误用法 - 格式化字符串参数的数目与占位符不相等  
rule.findbugs.NP\_NULL\_ON\_SOME\_PATH\_EXCEPTION.name=错误用法 - 方法的异常路径中可能引用空指针  
rule.findbugs.DL\_SYNCHRONIZATION\_ON\_BOXED\_PRIMITIVE.name=多线程错误 - 同步装箱基本类型可能导致死锁  
rule.findbugs.IJU\_TEARDOWN\_NO\_SUPER.name=错误用法 - TestCase定义的tearDown没有调用super.tearDown()  
rule.findbugs.SE\_READ\_RESOLVE\_IS\_STATIC.name=错误用法 - readResolve方法没有生命为static方法  
rule.findbugs.NP\_NULL\_ON\_SOME\_PATH\_MIGHT\_BE\_INFEASIBLE.name=高危 - 这个路径上空指针引用是不可实现的  
rule.findbugs.UMAC\_UNCALLABLE\_METHOD\_OF\_ANONYMOUS\_CLASS.name=错误用法 - 匿名内部类中定义的不可调用的方法  
rule.findbugs.VA\_FORMAT\_STRING\_BAD\_CONVERSION.name=错误用法 - 提供的参数类型与格式化标记不符  
rule.findbugs.EC\_ARRAY\_AND\_NONARRAY.name=错误用法 - equals()用来比较数组和非数组  
rule.findbugs.NM\_BAD\_EQUAL.name=类定义了equal(Object)，应该是equals(Object)吧？  
rule.findbugs.EC\_UNRELATED\_TYPES\_USING\_POINTER\_EQUALITY.name=错误用法 - 使用指针比较不同的类型  
rule.findbugs.STI\_INTERRUPTED\_ON\_CURRENTTHREAD.name=错误用法 - currentThread()调用的不必要用法，调用了interrupted()  
rule.findbugs.RE\_CANT\_USE\_FILE\_SEPARATOR\_AS\_REGULAR\_EXPRESSION.name=错误用法 - File.separator用做正则表达式  
rule.findbugs.MWN\_MISMATCHED\_WAIT.name=多线程错误 - 不匹配的wait()  
rule.findbugs.IL\_INFINITE\_LOOP.name=错误用法 - 明显的无限循环  
rule.findbugs.NP\_IMMEDIATE\_DEREFERENCE\_OF\_READLINE.name=高危 - 立刻使用了readLine()的结果  
rule.findbugs.SC\_START\_IN\_CTOR.name=多线程错误 - 构造方法调用了Thread.start()  
rule.findbugs.STCAL\_STATIC\_SIMPLE\_DATE\_FORMAT\_INSTANCE.name=多线程错误 - 静态DateFormat  
rule.findbugs.HE\_EQUALS\_NO\_HASHCODE.name=不良实践 - 类定义了equals()，但是没有hashCode()  
rule.findbugs.UL\_UNRELEASED\_LOCK.name=多线程错误 - 方法没有在任何路径中释放锁  
rule.findbugs.PZLA\_PREFER\_ZERO\_LENGTH\_ARRAYS.name=高危 - 考虑返回空数组而不是null  
rule.findbugs.SKIPPED\_CLASS\_TOO\_BIG.name=高危 - 类太大不能分析  
rule.findbugs.NP\_ARGUMENT\_MIGHT\_BE\_NULL.name=错误用法 - 方法没有检查空参数  
rule.findbugs.UM\_UNNECESSARY\_MATH.name=性能 - 方法常量调用了静态Math类方法  
rule.findbugs.NM\_WRONG\_PACKAGE\_INTENTIONAL.name=不良实践 - 方法没有覆盖父类的方法因为参数包名错误  
rule.findbugs.NP\_NONNULL\_PARAM\_VIOLATION.name=错误用法 - 方法传递null给一个非空参数  
rule.findbugs.BIT\_AND\_ZZ.name=错误用法 - 检查 ((...) & 0) == 0 是否成立  
rule.findbugs.HRS\_REQUEST\_PARAMETER\_TO\_COOKIE.name=安全风险 - HTTP cookie可能来自不被信任的输入  
rule.findbugs.SQL\_BAD\_RESULTSET\_ACCESS.name=错误用法 - 方法尝试访问一个ResultSet，但是使用的index是0  
rule.findbugs.INT\_VACUOUS\_COMPARISON.name=高危 - 整形的空比较  
rule.findbugs.EQ\_COMPARETO\_USE\_OBJECT\_EQUALS.name=不良实践 - 类定义了compareTo(...)，但使用了Object.equals()  
rule.findbugs.DMI\_HARDCODED\_ABSOLUTE\_FILENAME.name=高危 - 类包含一个硬编码的绝对路径  
rule.findbugs.DMI\_COLLECTIONS\_SHOULD\_NOT\_CONTAIN\_THEMSELVES.name=错误用法 - 集合不应该包含自身  
rule.findbugs.URF\_UNREAD\_FIELD.name=性能 - 读不到的属性  
rule.findbugs.DLS\_OVERWRITTEN\_INCREMENT.name=错误用法 - 重写了自增  
rule.findbugs.BIT\_SIGNED\_CHECK.name=不良实践 - 检测带符号的位运算  
rule.findbugs.UWF\_NULL\_FIELD.name=错误用法 - 属性曾经设置为null  
rule.findbugs.DE\_MIGHT\_DROP.name=不良实践 - 方法可能抛出异常  
rule.findbugs.DMI\_BAD\_MONTH.name=错误用法 - 对于month错误的常量值  
rule.findbugs.MS\_MUTABLE\_ARRAY.name=恶意代码漏洞 - 属性是可变数组  
rule.findbugs.SE\_INNER\_CLASS.name=不良实践 - 可序列化的内部类  
rule.findbugs.OS\_OPEN\_STREAM\_EXCEPTION\_PATH.name=不良实践 - 方法可能在关闭流时因为异常而失败  
rule.findbugs.AT\_OPERATION\_SEQUENCE\_ON\_CONCURRENT\_ABSTRACTION.name=并发抽象的顺序调用可能不是原子的  
rule.findbugs.BX\_UNBOXING\_IMMEDIATELY\_REBOXED.name=装箱的值被拆箱，然后立刻重新装箱了  
rule.findbugs.CO\_COMPARETO\_RESULTS\_MIN\_VALUE.name=compareTo()或compare()返回Integer.MIN\_VALUE  
rule.findbugs.DLS\_DEAD\_LOCAL\_STORE\_SHADOWS\_FIELD.name=无用的局部变量可能与实例属性同名  
rule.findbugs.DMI\_ARGUMENTS\_WRONG\_ORDER.name=方法参数顺序翻转  
rule.findbugs.DMI\_BIGDECIMAL\_CONSTRUCTED\_FROM\_DOUBLE.name=由double构造BigDecimal时没有指定精确度  
rule.findbugs.DMI\_DOH.name=一个明显不合理的方法调用  
rule.findbugs.DMI\_ENTRY\_SETS\_MAY\_REUSE\_ENTRY\_OBJECTS.name=一个入口的添加可能因为重用Entry对象导致失败  
rule.findbugs.DM\_DEFAULT\_ENCODING.name=信任默认字符编码  
rule.findbugs.ICAST\_INT\_2\_LONG\_AS\_INSTANT.name=int转换成long，用来当做绝对时间  
rule.findbugs.INT\_BAD\_COMPARISON\_WITH\_INT\_VALUE.name=错误比较int值和long常量  
rule.findbugs.JML\_JSR166\_CALLING\_WAIT\_RATHER\_THAN\_AWAIT.name=在util.concurrent抽象中使用监控样式的wait方法  
rule.findbugs.NP\_UNWRITTEN\_PUBLIC\_OR\_PROTECTED\_FIELD.name=读取了未初始化的public或protected属性  
rule.findbugs.OBL\_UNSATISFIED\_OBLIGATION\_EXCEPTION\_EDGE.name=方法可能因为checked exception导致清理流或资源失败  
rule.findbugs.PZ\_DONT\_REUSE\_ENTRY\_OBJECTS\_IN\_ITERATORS.name=不用在iterator里重用entry对象  
rule.findbugs.RV\_CHECK\_COMPARETO\_FOR\_SPECIFIC\_RETURN\_VALUE.name=compareTo返回的值与指定的值进行检测  
rule.findbugs.RV\_NEGATING\_RESULT\_OF\_COMPARETO.name=调换了compareTo()/compare()结果的正负  
rule.findbugs.RV\_RETURN\_VALUE\_IGNORED\_INFERRED.name=方法忽略了返回值，这是正常的吗？  
rule.findbugs.SA\_LOCAL\_SELF\_ASSIGNMENT\_INSTEAD\_OF\_FIELD.name=局部变量给自己赋值而不是赋值到实例变量中  
rule.findbugs.URF\_UNREAD\_PUBLIC\_OR\_PROTECTED\_FIELD.name=未被读取的public/protected属性  
rule.findbugs.UUF\_UNUSED\_PUBLIC\_OR\_PROTECTED\_FIELD.name=未被使用的public 或 protected 属性  
rule.findbugs.UWF\_UNWRITTEN\_PUBLIC\_OR\_PROTECTED\_FIELD.name=未初始化的public 或 protected 属性  
rule.findbugs.VA\_FORMAT\_STRING\_USES\_NEWLINE.name=格式化代码应该使用%n代替\n  
rule.findbugs.VO\_VOLATILE\_INCREMENT.name=volatile的自增操作不是原子的