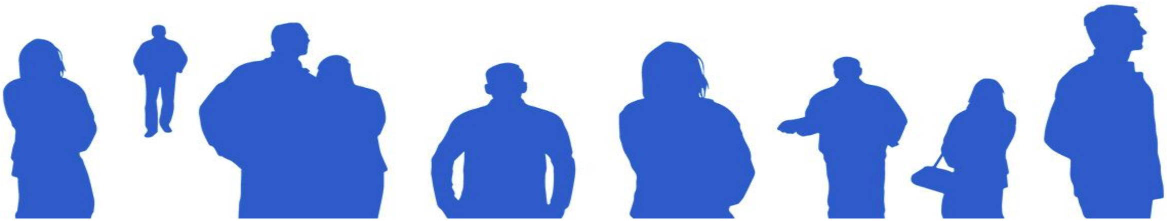

Ministerie van BZK
Operatie BRP
Turfmarkt 147
2511 DC Den Haag

www.operatieBRP.nl
info@operatiebrp.nl

NOTITIE
Normenkader codekwaliteit
Centrale BRP-voorzieningen

Datum
15-04-2015

Versie 1.1 vastgesteld stuurgroep Operatie BRP dd. 23-04-2015

Inleiding

Het doel van deze notitie is het vaststellen van het normenkader voor de codekwaliteit van de centrale BRP-voorzieningen en de migratievoorzieningen. Bij het opstellen van de notitie is gebruik gemaakt van het advies 'Kwaliteitsdoelen softwareontwikkeling' van KPMG aan opdrachtgever Operatie BRP van 2 december 2014. Verder is gebruik gemaakt van ISO norm 25010, die kwaliteitskenmerken van software beschrijft. In dit normenkader ligt de nadruk op de categorieën betrouwbaarheid, beveiligbaarheid en onderhoudbaarheid.

De norm geeft invulling aan een deel van de zogenoemde 'non-functional requirements' (NFR's) die door Agentschap BPR en andere partijen zijn opgesteld. Een groot aantal regels in deze norm draagt bij aan een of meerdere NFR's of vult deze volledig in[1].

Bij het vaststellen van het normenkader zijn de volgende partijen betrokken:
1. Gedelegeerd opdrachtgever Operatie BRP (als verantwoordelijke)
2. Project O&R (als uitvoerder)
3. Agentschap BPR (als toekomstig beheerder)
4. KPMG (vanuit zijn verantwoordelijkheid voor Quality Assurance rond de broncode)

[1] Zie bijlage 3

De notitie is als volgt opgebouwd:

1. Procesinrichting
2. Afbakening
3. Normstelling
4. Omgang met uitkomsten beoordelingen en toetsen

Procesinrichting

Het proces rond het normenkader bestaat uit de volgende onderdelen:

- Vaststelling en beheer van de norm
- Toepassing van de norm
- Toetsing van de norm

Vaststelling en beheer van de norm

De gedelegeerd opdrachtgever stelt de kwaliteitsnorm voor de code (deze notitie) vast in overleg met het project O&R in de rol van uitvoerder en het Agentschap BPR in de rol van toekomstig beheerder. KPMG geeft een kwaliteitsoordeel over de norm die de gedelegeerd opdrachtgever stelt, inclusief eventuele aanbevelingen. De gedelegeerd opdrachtgever legt expliciet vast welke aanbevelingen hij wel en niet in het normenkader verwerkt. Het project O&R maakt de kwaliteitsnorm, het oordeel van KPMG en de reactie van de gedelegeerd opdrachtgever over de aanbevelingen van KPMG openbaar.

Veranderingen van inzicht op het terrein van codekwaliteit kunnen leiden tot wijzigingsvoorstellen op de kwaliteitsnorm. Deze wijzigingen volgen hetzelfde proces als de vaststelling van de norm. Het project O&R bepaalt conform het reguliere wijzigingenproces na een besluit de impact van de wijziging op tijd en geld. De gedelegeerd opdrachtgever beslist of deze impact zodanig is dat hij het besluit over de wijziging van de kwaliteitsnorm aan de stuurgroep Operatie BRP voorlegt.

Toepassing van de norm

Het project O&R past de kwaliteitsnorm toe op alle code die door het project zelf wordt geproduceerd. De omvang daarvan wordt verder beschreven onder de kop Afbakening.

De toepassing van de kwaliteitsnorm krijgt concreet vorm door de inzet van de volgende producten:

- Findbugs
- PMD
- Checkstyle
- SonarQube

De controle op de kwaliteit van de code vindt zo vroeg mogelijk in het proces van het vervaardigen van code plaats. Dit betekent dat een programmeur na het invoeren van nieuwe regels code direct een terugmelding krijgt die aangeeft of de code aan de in de kwaliteitsnorm vastgelegde regels voldoet. Deze werkwijze is efficiënter en effectiever dan een werkwijze waarbij achteraf (bijvoorbeeld aan het eind van iedere dag) een batchgewijze toets plaatsvindt op de geproduceerde code, waarna de programmeur de volgende dag signalen krijgt over de onderdelen van de door hem geproduceerde code die niet voldoen aan de kwaliteitsnorm.

Het is mogelijk dat het oplossen van een situatie waarin niet wordt voldaan aan een specifieke regel uit het normenkader tot kwalitatief minder goede broncode zou leiden. Voor bepaalde normen[2] geldt daarom de regel 'Pas toe of leg uit'. Dit betekent dat per individuele afwijking van een bepaalde regel een uitleg ('explain') gedocumenteerd moet worden waarom de afwijking naar de mening van het project niet wordt opgelost. Het doel is om het aantal 'explains' zo laag mogelijk te houden. Monitoring en eerste inhoudelijke beoordeling van 'explains' is een taak van de ontwikkelteams.

Toetsing van de norm

Periodiek vindt een toetsing plaats door KPMG (verantwoordelijk voor Quality Assurance rond de broncode). Het doel hiervan is tweeledig:

1. toetsen in hoeverre de door het project geproduceerde code voldoet aan het gestelde normenkader;
2. bepalen of het proces van kwaliteitsborging nog steeds op orde is.

Daarmee wordt vervolg gegeven aan aanbevelingen op basis van de eerste toets door KPMG, en wordt gecontroleerd hoe er wordt gestuurd op bevindingen en of de hiervoor beschreven afwijkingen van deze regels ('explains') in voldoende mate worden beargumenteerd. KPMG rapporteert over de uitkomst van deze toetsen.

De gedelegeerd opdrachtgever, het Agentschap BPR en de stuurgroep Operatie BRP krijgen daarmee de beschikking over onafhankelijk opgestelde rapportages over de codekwaliteit.

Afbakening

De kwaliteitsnorm is integraal van toepassing op code die door project O&R zelf is geschreven en die bedoeld is om in productie te gaan. De volgende code wordt buiten beschouwing gelaten:

- de broncode van de componenten die worden aangeduid als 'de BRP generatoren' (de norm is dus *wel* van toepassing op met deze generatoren gegenereerde code)[3];
- third party libraries alsmede code gegenereerd door deze libraries[4,5];
- twee specifieke gevallen van code hergebruik binnen de migratiecomponenten:
    - een door het project zelf aangepast stuk JBoss broncode;
    - (her)gebruikte delen van bestaande GBA-V broncode[6].

Ten aanzien van de norm voor beveiliging gelden enkele nuanceringen van bovengenoemde afbakening. Als er onverhoopt een "beveiligingsissue" in de hierboven

[2] Specifiek: voor de normen 1, 2 en 6 in de tabel in sectie Normstelling. Voor de normen 3, 4, 5 en 7 kan de regel 'Pas toe of leg uit' niet per individuele afwijking worden toegepast, aangezien het daar gaat om percentages die worden gemeten over het geheel van de code. Voor 3, 4, 5 en 7 kan nog wel gelden dat bepaalde delen van de code worden uitgesloten van de norm. Daar hoort dan vanzelfsprekend ook een adequate uitleg bij.

[3] Gebaseerd op de voorlopige aanname dat de BRP-generatoren niet worden overgedragen aan het Agentschap BPR. Als deze aanname in de loop van het project wijzigt, volgt een impactanalyse en wordt op basis van de impact een definitief besluit hieromtrent genomen.

[4] Zowel source code als (na compilatie in te voegen) byte code.

[5] Inclusief mogelijk nog te selecteren libraries die worden ingezet voor de realisatie van beheerfunctionaliteit.

[6] Motivatie voor de gekozen oplossingen en onderbouwing dat de risico's hiervan beperkt zijn zal worden opgenomen in SAD of Technische Ontwerpdocumentatie.

opgesomde code-onderdelen voorkomt, dan zal daar op de volgende manieren mee worden omgegaan:

- In geval van third party libraries:
    1. Voor het eerste gebruik van een library wordt deze getoetst op veiligheidsissues. Alleen libraries waarin geen "beveiligingsissues" voorkomen mogen worden gebruikt.
    2. Als gedurende het gebruik alsnog een "beveiligingsissue" wordt ontdekt dan volgt een afweging met als mogelijke uitkomsten:
        a. de library wordt als ‘bad practice’ bestempeld en dus niet gebruikt, of;
        b. in overleg met de beveiligingsfunctionaris van Agentschap BPR maakt het project de afweging dat het risico (bijvoorbeeld in het licht van de architectuur van de BRP) toch acceptabel is. Dan kan de library met een bijbehorende toelichting in het SAD alsnog gebruikt (blijven) worden.
- In geval van hergebruikte broncode dient het issue, voor zover in het zelf aangepaste deel is ontstaan, ofwel te worden opgelost, ofwel te worden voorzien van een goede ‘explain’. Voor hergebruikte delen van de GBA-V broncode wordt de gedragslijn gevolgd die Agentschap BPR hanteert voor de GBA-V broncode en zal voor de aanpassing hiervan een beroep gedaan worden op Agentschap BPR.

Voor de broncode van de BRP generatoren gelden deze speciale eisen niet omdat deze code nooit in productie gaat, ze is geen onderdeel van de BRP-voorziening maar een hulpmiddel dat gebruikt wordt bij de voorbrenging.

Normstelling

De normen zijn vastgesteld zoals beschreven in onderstaande tabel.

Nr	Aspect	Norm
1	Aantal blocker of critical issues	0 (nul, geen)
2	Aantal issues ten aanzien van veiligheid en betrouwbaarheid	0 (nul, geen)
3	Testdekking op productiecode	Minimaal 80%
4	Documentatie publieke API	Minimaal 95%
5	Code duplicatie	Maximaal 4%
6	Cyclische afhankelijkheden	0 (nul, geen)
7	Rule Compliance Index	Minimaal 97%

De te hanteren meetmethode per aspect is onderdeel van de norm, en is beschreven in bijlage 1 aan het eind van dit document.

"Explains" maken geen deel uit van de telling ten behoeve van de normen 1 tot en met 6. Ten aanzien van norm 7 wordt de specifieke bijdrage van de issues die de "Explains" veroorzaken in de berekening van de Rule Compliance Index in kaart gebracht, de berekening van de index wordt voor deze bijdrage gecorrigeerd door de door "Explains" veroorzaakte bevindingen niet in de telling mee te nemen.
Norm nummer 7 is er op gericht om te borgen dat het totaal aantal issues (inclusief het deel met een ‘explain’) proportioneel blijft ten opzichte van de omvang van de code.

In dit normenkader worden uitsluitingen van aangewezen delen van de code voor specifieke normen vastgelegd. Aangewezen delen betreffen de stukken code zoals benoemd in de afbakening en de markeringen ten behoeve van de "explains".
Deze uitsluitingen worden door middel van een technische markering in de code aangebracht ten behoeve van de gebruikte tooling. Deze technische markering kan op zichzelf ook leiden tot een afwijking van een regel als genoemd in Bijlage 2. Deze markeringen tellen niet mee bij tellingen ten behoeve van alle normen (inclusief norm 7) en behoeven geen specifieke "explain". De algemene verantwoording is het technisch mogelijk maken van metingen van de codekwaliteit op basis van het normenkader.

Omgang met uitkomsten beoordelingen en toetsen

Bij beoordelingen (toetsing bij opstellen code, interne reviews en testen) en toetsen (door KPMG) zullen bevindingen ontstaan. Deze worden primair besproken binnen het ontwikkelteam, dat bepaalt hoe de bevinding af te handelen. Als de bevinding terecht wordt bevonden, wordt het werk voor het herstellen van de bevinding geschat en ingepland.

Als er discussie ontstaat over het al dan niet terecht zijn van een bevinding of over de wijze waarop of het tempo waarin deze moet worden opgelost, gelden vijf escalatieniveaus:

1. Ontwikkelteam (waaronder de scrum master)
2. Teamleider en domeinarchitect
3. Projectleider O&R
4. Gedelegeerd opdrachtgever Operatie BRP
5. Stuurgroep

Bij escalatieniveau's 2 tot en met 4 stemt het project af met Agentschap BPR.

Allereerst wordt getracht de discussie binnen het team op te lossen. De scrum master is onderdeel van het team met als speciale verantwoordelijkheid het borgen van het samenwerkingsproces. Als het team geen oplossing kan vinden wordt het issue aan de teamleider en domeinarchitect voorgelegd. De volgende escalatiestap is de projectleider O&R. Deze kan eventueel de lead architect om advies vragen. Als ook na afstemming met projectleider O&R de discussie niet tot afsluiting komt, wordt de bevinding voorgelegd aan de gedelegeerd opdrachtgever die het finale besluit neemt. Deze kan eventueel besluiten om KPMG (Quality Assurance) om advies te vragen of om het onderwerp voor te leggen aan de stuurgroep.

Mogelijk vormt een afwijking aanleiding tot aanpassing van het normenkader. De besluitvorming daarover is de verantwoordelijkheid van de gedelegeerd opdrachtgever Operatie BRP. Het opstellen van een nieuwe versie van het normenkader vindt plaats volgens het proces dat eerder beschreven onder "Vaststelling en beheer van de norm".

Bijlage 1: Meetmethoden

Nr	Aspect	Meetmethode
1	Aantal blocker of critical issues	Indicatoren ‘blocker’ en ‘critical’ in SonarQube versie 4.4 op basis van de door project O&R op 1 december 2014 afgestemde set Findbugs versie 3.0, PMD versie 2.2 en Checkstyle versie 2.1.1 regels. De complete regelset is opgenomen in bijlage 2.
2	Aantal issues ten aanzien van veiligheid en betrouwbaarheid	Findbugs, PMD en Checkstyle regels die betrekking hebben op veiligheid en betrouwbaarheid. Bijlage 2 bevat per regel een aanduiding of deze betrekking heeft op veiligheid of betrouwbaarheid.
3	Testdekking op productiecode	De norm geldt voor twee delen: het geheel van BRP-code en het geheel van Migratie-code. De norm geldt voor zowel line coverage als branch coverage. Voor de BRP-code wordt Cobertura versie 1.6.3 ingezet voor coverage meting, voor de Migratie-code Jacoco versie 2.3. Deze norm niet van toepassing op gegenereerde code. Toelichting: Gegenereerde code gebruikt een standaard patroon dat meerdere malen wordt herhaald. Daarom worden uit efficiency overwegingen alleen unit tests gemaakt voor het template en niet voor elke instantie daarvan in de code. Omdat de meting gebeurt op basis van de code en de tooling de bovengenoemde nuance niet kan meenemen wordt gegenereerde code niet meegeteld.
4	Documentatie publieke API	Indicator ‘Public documented API’ in SonarQube
5	Code duplicatie	Indicator ‘Duplications’ in SonarQube. De BRP software wordt gebouwd met behulp van code generatoren en zoveel mogelijk op basis van generieke patronen. Deze combinatie resulteert in en grotere hoeveelheid code duplicatie dan als deze met de hand geschreven wordt. Deze norm is daarom niet van toepassing op gegenereerde code.
6	Cyclische afhankelijkheden	Indicator ‘Package cycles’ in SonarQube. Een aantal cyclische afhankelijkheden wordt veroorzaakt door architectuurkeuzes, met name in de code delen “BRP Algemeen Model” (expressietaal en het model zelf). Deze cyclische afhankelijkheden zullen door het project worden aangewezen en vallen buiten de norm en tellen niet mee in issue tellingen en explains. Het Software Architectuur Document licht de architectuurkeuzes toe. Buiten deze specifieke categorie cyclische afhankelijkheden geldt de norm 0 in combinatie met “pas toe of leg uit”.

Nr	Aspect	Meetmethode
7	Rule Compliance Index	De RCI wordt berekend aan de hand van de indicatoren blocker, critical, major, minor uit SonarQube. $\text{gewogen aantal issues} = 10 \times \text{aantal blocker issues} + 5 \times \text{aantal critical issues} + 3 \times \text{aantal major issues} + \text{aantal minor issues}$ $RCI = (1 - (\text{gewogen aantal issues}/\text{aantal regels code})) \times 100\%$

Bijlage 2 Regelset

Onderstaande tabel geeft omschrijvingen en categorieën bij de regels die onderdeel uitmaken van de norm.
Bron: bestand 'sonar-java-profiel-OperatieBRP.xml' d.d. 1 december 2014.

In dit normenkader worden uitsluitingen van aangewezen delen van de code voor specifieke normen vastgelegd. Deze uitsluitingen worden door middel van een technische markering (zoals //NOSONAR, CHECKSTYLE:OFF en NOPMD) in de code aangebracht ten behoeve van de gebruikte tooling,. Deze technische markering kan op zichzelf ook leiden tot een afwijking van bijvoorbeeld de hieronder genoemde drie regels. Deze markeringen tellen niet mee bij tellingen ten behoeve van alle normen (inclusief norm 7) en daarom maken deze genoemde drie regels geen onderdeel uit van het normenkader, evenals andere regels die dezelfde situatie veroorzaken.

Het gaat om de volgende regels:

- Avoid use of //NOSONAR marker (Categorie SECURITY)
- CHECKSTYLE:OFF suppression comment filters should not be used (Categorie MAINTAINABILITY)
- NOPMD suppression comment filters should not be used (Categorie MAINTAINABILITY)

Om die reden maken deze regels geen onderdeel uit van de norm.

Regel	Categorie	Configuratiecode
Avoid cycle between java packages	CHANGEABILITY	squid:CycleBetweenPackages
Avoid Duplicate Literals	CHANGEABILITY	pmd:AvoidDuplicateLiterals
Bad practice - Fields of immutable classes should be final	CHANGEABILITY	findbugs:JCIP_FIELD_ISNT_FINAL_IN_IMMUTABLE_CLASS
Bad practice - Superclass uses subclass during initialization	CHANGEABILITY	findbugs:IC_SUPERCLASS_USES_SUBCLASS_DURING_INITIALIZATION
Class Fan Out Complexity	CHANGEABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.metrics.ClassFanOutComplexityCheck
Correctness - Class defines field that masks a superclass field	CHANGEABILITY	findbugs:MF_CLASS_MASKS_FIELD
Default Comes Last	CHANGEABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.DefaultComesLastCheck

Design For Extension	CHANGEABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.design.DesignForExtensionCheck
Dodgy - Ambiguous invocation of either an inherited or outer method	CHANGEABILITY	findbugs:IA_AMBIGUOUS_INVOCATION_OF_INHERITED_OR_OUTER_METHOD
Experimental - Abstract Method is already defined in implemented interface	CHANGEABILITY	findbugs:USM_USELESS_ABSTRACT_METHOD
Experimental - Method accesses a private member variable of owning class	CHANGEABILITY	findbugs:IMA_INEFFICIENT_MEMBER_ACCESS
Experimental - Test for circular dependencies among classes	CHANGEABILITY	findbugs:CD_CIRCULAR_DEPENDENCY
Interface Is Type	CHANGEABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.design.InterfaceIsTypeCheck
Loose coupling	CHANGEABILITY	pmd:LooseCoupling
Loose Coupling (With Type Resolution)	CHANGEABILITY	pmd:LooseCouplingWithTypeResolution
Magic Number	CHANGEABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.MagicNumberCheck
Need Braces	CHANGEABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.blocks.NeedBracesCheck
Nested For Depth	CHANGEABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.NestedForDepthCheck
Nested If Depth	CHANGEABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.NestedIfDepthCheck
Nested Try Depth	CHANGEABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.NestedTryDepthCheck
Replace Hashtable With Map	CHANGEABILITY	pmd:ReplaceHashtableWithMap
Replace Vector With List	CHANGEABILITY	pmd:ReplaceVectorWithList
Return Count	CHANGEABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.ReturnCountCheck
Throws Count	CHANGEABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.design.ThrowsCountCheck
Use Array List Instead Of Vector	CHANGEABILITY	pmd:UseArrayListInsteadOfVector
Useless Overriding Method	CHANGEABILITY	pmd:UselessOverridingMethod
Visibility Modifier	CHANGEABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.design.VisibilityModifierCheck
Avoid Array Loops	EFFICIENCY	pmd:AvoidArrayLoops
Big Integer Instantiation	EFFICIENCY	pmd:BigIntegerInstantiation
Boolean Instantiation	EFFICIENCY	pmd:BooleanInstantiation
Boxed value is unboxed and then immediately reboxed	EFFICIENCY	findbugs:BX_UNBOXING_IMMEDIATELY_REBOXED
Final Field Could Be Static	EFFICIENCY	pmd:FinalFieldCouldBeStatic

Inefficient String Buffering	EFFICIENCY	pmd:InefficientStringBuffering
Instantiation To Get Class	EFFICIENCY	pmd:InstantiationToGetClass
Integer Instantiation	EFFICIENCY	pmd:IntegerInstantiation
Performance - Could be refactored into a named static inner class	EFFICIENCY	findbugs:SIC_INNER_SHOULD_BE_STATIC_ANON
Performance - Could be refactored into a static inner class	EFFICIENCY	findbugs:SIC_INNER_SHOULD_BE_STATIC_NEEDS_THIS
Performance - Huge string constants is duplicated across multiple class files	EFFICIENCY	findbugs:HSC_HUGE_SHARED_STRING_CONSTANT
Performance - Inefficient use of keySet iterator instead of entrySet iterator	EFFICIENCY	findbugs:WMI_WRONG_MAP_ITERATOR
Performance - Maps and sets of URLs can be performance hogs	EFFICIENCY	findbugs:DMI_COLLECTION_OF_URLS
Performance - Method allocates a boxed primitive just to call toString	EFFICIENCY	findbugs:DM_BOXED_PRIMITIVE_TOSTRING
Performance - Method allocates an object, only to get the class object	EFFICIENCY	findbugs:DM_NEW_FOR_GETCLASS
Performance - Method calls static Math class method on a constant value	EFFICIENCY	findbugs:UM_UNNECESSARY_MATH
Performance - Method concatenates strings using + in a loop	EFFICIENCY	findbugs:SBSC_USE_STRINGBUFFER_CONCATENATION
Performance - Method invokes inefficient Boolean constructor; use Boolean.valueOf(...) instead	EFFICIENCY	findbugs:DM_BOOLEAN_CTOR
Performance - Method invokes inefficient floating-point Number constructor; use static valueOf instead	EFFICIENCY	findbugs:DM_FP_NUMBER_CTOR
Performance - Method invokes inefficient new String() constructor	EFFICIENCY	findbugs:DM_STRING_VOID_CTOR
Performance - Method invokes inefficient new String(String) constructor	EFFICIENCY	findbugs:DM_STRING_CTOR
Performance - Method invokes inefficient Number constructor; use static valueOf instead	EFFICIENCY	findbugs:DM_NUMBER_CTOR
Performance - Method invokes toString()	EFFICIENCY	findbugs:DM_STRING_TOSTRING

method on a String		
Performance - Method uses toArray() with zero-length array argument	EFFICIENCY	findbugs:ITA_INEFFICIENT_TO_ARRAY
Performance - Primitive value is boxed and then immediately unboxed	EFFICIENCY	findbugs:BX_BOXING_IMMEDIATELY_UNBOXED
Performance - Primitive value is boxed then unboxed to perform primitive coercion	EFFICIENCY	findbugs:BX_BOXING_IMMEDIATELY_UNBOXED_TO_PERFORM_COERCION
Performance - Should be a static inner class	EFFICIENCY	findbugs:SIC_INNER_SHOULD_BE_STATIC
Performance - The equals and hashCode methods of URL are blocking	EFFICIENCY	findbugs:DMI_BLOCKING_METHODS_ON_URL
Performance - Use the nextInt method of Random rather than nextDouble to generate a random integer	EFFICIENCY	findbugs:DM_NEXTINT_VIA_NEXTDOUBLE
String Instantiation	EFFICIENCY	pmd:StringInstantiation
Super Clone	EFFICIENCY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.SuperCloneCheck
Super Finalize	EFFICIENCY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.SuperFinalizeCheck
Unnecessary Case Change	EFFICIENCY	pmd:UnnecessaryCaseChange
Unnecessary Local Before Return	EFFICIENCY	pmd:UnnecessaryLocalBeforeReturn
Unused Null Check In Equals	EFFICIENCY	pmd:UnusedNullCheckInEquals
Use Arrays As List	EFFICIENCY	pmd:UseArraysAsList
Use Index Of Char	EFFICIENCY	pmd:UseIndexOfChar
Abstract Class Name	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.naming.AbstractClassNameCheck
Annotation Use Style	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.annotation.AnnotationUseStyleCheck
Anon Inner Length	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.sizes.AnonInnerLengthCheck
Array Trailing Comma	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.ArrayTrailingCommaCheck
Array Type Style	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.ArrayTypeStyleCheck
Avoid Instanceof Checks In Catch Clause	MAINTAINABILITY	pmd:AvoidInstanceofChecksInCatchClause
Avoid Nested Blocks	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.blocks.AvoidNestedBlocksCheck
Bad practice - Class defines clone() but doesn't implement Cloneable	MAINTAINABILITY	findbugs:CN_IMPLEMENTS_CLONE_BUT_NOT_CLONEABLE

Bad practice - Class implements Cloneable but does not define or use clone method	MAINTAINABILITY	findbugs:CN_IDIOM
Bad practice - Class is not derived from an Exception, even though it is named as such	MAINTAINABILITY	findbugs:NM_CLASS_NOT_EXCEPTION
Bad practice - Class names shouldn't shadow simple name of implemented interface	MAINTAINABILITY	findbugs:NM_SAME_SIMPLE_NAME_AS_INTERFACE
Bad practice - Class names shouldn't shadow simple name of superclass	MAINTAINABILITY	findbugs:NM_SAME_SIMPLE_NAME_AS_SUPERCLASS
Bad practice - Confusing method names	MAINTAINABILITY	findbugs:NM_CONFUSING
Bad practice - Empty finalizer should be deleted	MAINTAINABILITY	findbugs:FI_EMPTY
Bad practice - Finalizer does nothing but call superclass finalizer	MAINTAINABILITY	findbugs:FI_USELESS
Bad practice - Finalizer nulls fields	MAINTAINABILITY	findbugs:FI_FINALIZER_NULLS_FIELDS
Bad practice - Finalizer only nulls fields	MAINTAINABILITY	findbugs:FI_FINALIZER_ONLY_NULLS_FIELDS
Bad practice - Method doesn't override method in superclass due to wrong package for parameter	MAINTAINABILITY	findbugs:NM_WRONG_PACKAGE_INTENTIONAL
Bad practice - Needless instantiation of class that only supplies static methods	MAINTAINABILITY	findbugs:ISC_INSTANTIATE_STATIC_CLASS
Bad practice - serialVersionUID isn't final	MAINTAINABILITY	findbugs:SE_NONFINAL_SERIALVERSIONID
Bad practice - serialVersionUID isn't long	MAINTAINABILITY	findbugs:SE_NONLONG_SERIALVERSIONID
Bad practice - serialVersionUID isn't static	MAINTAINABILITY	findbugs:SE_NONSTATIC_SERIALVERSIONID
Bad practice - Unchecked type in generic call	MAINTAINABILITY	findbugs:GC_UNCHECKED_TYPE_IN_GENERIC_CALL
Bad practice - Very confusing method names (but perhaps intentional)	MAINTAINABILITY	findbugs:NM_VERY_CONFUSING_INTENTIONAL
Boolean Expression Complexity	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.metrics.BooleanExpressionComplexityCheck
Class names should start with an upper case letter	MAINTAINABILITY	findbugs:NM_CLASS_NAMING_CONVENTION

Class Type(Generic) Parameter Name	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.naming.ClassTypeParameterNameCheck
Clone method must implement Cloneable	MAINTAINABILITY	pmd:CloneMethodMustImplementCloneable
Clone Throws Clone Not Supported Exception	MAINTAINABILITY	pmd:CloneThrowsCloneNotSupportedException
Collapsible If Statements	MAINTAINABILITY	pmd:CollapsibleIfStatements
Correctness - A known null value is checked to see if it is an instance of a type	MAINTAINABILITY	findbugs:NP_NULL_INSTANCEOF
Correctness - Call to equals() with null argument	MAINTAINABILITY	findbugs:EC_NULL_ARG
Correctness - Can't use reflection to check for presence of annotation without runtime retention	MAINTAINABILITY	findbugs:DMI_ANNOTATION_IS_NOT_VISIBLE_TO_REFLECTION
Correctness - Covariant equals() method defined for enum	MAINTAINABILITY	findbugs:EQ_DONT_DEFINE_EQUALS_FOR_ENUM
Correctness - Covariant equals() method defined, Object.equals(Object) inherited	MAINTAINABILITY	findbugs:EQ_SELF_USE_OBJECT
Correctness - Dead store of class literal	MAINTAINABILITY	findbugs:DLS_DEAD_STORE_OF_CLASS_LITERAL
Correctness - Double assignment of field	MAINTAINABILITY	findbugs:SA_FIELD_DOUBLE_ASSIGNMENT
Correctness - Field only ever set to null	MAINTAINABILITY	findbugs:UWF_NULL_FIELD
Correctness - Method call passes null for nonnull parameter	MAINTAINABILITY	findbugs:NP_NULL_PARAM_DEREF
Correctness - Method call passes null for nonnull parameter (ALL_TARGETS_DANGEROUS)	MAINTAINABILITY	findbugs:NP_NULL_PARAM_DEREF_ALL_TARGETS_DANGEROUS
Correctness - Method call passes null to a nonnull parameter	MAINTAINABILITY	findbugs:NP_NONNULL_PARAM_VIOLATION
Correctness - Nullcheck of value previously dereferenced	MAINTAINABILITY	findbugs:RCN_REDUNDANT_NULLCHECK_WOULD_HAVE_BEEN_A_NPE
Correctness - TestCase declares a bad suite method	MAINTAINABILITY	findbugs:IJU_BAD_SUITE_METHOD
Correctness - TestCase has no tests	MAINTAINABILITY	findbugs:IJU_NO_TESTS
Correctness - TestCase implements a	MAINTAINABILITY	findbugs:IJU_SUITE_NOT_STATIC

non-static suite method		
Correctness - Uncallable method defined in anonymous class	MAINTAINABILITY	findbugs:UMAC_UNCALLABLE_METHOD_OF_ANONYMOUS_CLASS
Correctness - Unnecessary type check done using instanceof operator	MAINTAINABILITY	findbugs:SIO_SUPERFLUOUS_INSTANCEOF
Correctness - Unneeded use of currentThread() call, to call interrupted()	MAINTAINABILITY	findbugs:STI_INTERRUPTED_ON_CURRENTTHREAD
Correctness - Unwritten field	MAINTAINABILITY	findbugs:UWF_UNWRITTEN_FIELD
Correctness - Useless assignment in return statement	MAINTAINABILITY	findbugs:DLS_DEAD_LOCAL_STORE_IN_RETURN
Correctness - Useless control flow to next line	MAINTAINABILITY	findbugs:UCF_USELESS_CONTROL_FLOW_NEXT_LINE
Correctness - Very confusing method names	MAINTAINABILITY	findbugs:NM_VERY_CONFUSING
Declaration Order	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.DeclarationOrderCheck
Dodgy - Class implements same interface as superclass	MAINTAINABILITY	findbugs:RI_REDUNDANT_INTERFACES
Dodgy - Class is final but declares protected field	MAINTAINABILITY	findbugs:CI_CONFUSED_INHERITANCE
Dodgy - Class too big for analysis	MAINTAINABILITY	findbugs:SKIPPED_CLASS_TOO_BIG
Dodgy - Dead store of null to local variable	MAINTAINABILITY	findbugs:DLS_DEAD_LOCAL_STORE_OF_NULL
Dodgy - Dead store to local variable	MAINTAINABILITY	findbugs:DLS_DEAD_LOCAL_STORE
Dodgy - Exception is caught when Exception is not thrown	MAINTAINABILITY	findbugs:REC_CATCH_EXCEPTION
Dodgy - Invocation of substring(0), which returns the original value	MAINTAINABILITY	findbugs:DMI_USELESS_SUBSTRING
Dodgy - Load of known null value	MAINTAINABILITY	findbugs:NP_LOAD_OF_KNOWN_NULL_VALUE
Dodgy - Method checks to see if result of String.indexOf is positive	MAINTAINABILITY	findbugs:RV_CHECK_FOR_POSITIVE_INDEXOF
Dodgy - private readResolve method not inherited by subclasses	MAINTAINABILITY	findbugs:SE_PRIVATE_READ_RESOLVE_NOT_INHERITED
Dodgy - Redundant comparison of non-null value to null	MAINTAINABILITY	findbugs:RCN_REDUNDANT_COMPARISON_OF_NULL_AND_NONNULL_VALUE

Dodgy - Redundant comparison of two null values	MAINTAINABILITY	findbugs:RCN_REDUNDANT_COMPARISON_TWO_NULL_VALUES
Dodgy - Redundant nullcheck of value known to be non-null	MAINTAINABILITY	findbugs:RCN_REDUNDANT_NULLCHECK_OF_NONNULL_VALUE
Dodgy - Redundant nullcheck of value known to be null	MAINTAINABILITY	findbugs:RCN_REDUNDANT_NULLCHECK_OF_NULL_VALUE
Dodgy - Unchecked/unconfirmed cast	MAINTAINABILITY	findbugs:BC_UNCONFIRMED_CAST
Dodgy - Useless control flow	MAINTAINABILITY	findbugs:UCF_USELESS_CONTROL_FLOW
Dodgy - Vacuous bit mask operation on integer value	MAINTAINABILITY	findbugs:INT_VACUOUS_BIT_OPERATION
Dodgy - Vacuous comparison of integer value	MAINTAINABILITY	findbugs:INT_VACUOUS_COMPARISON
Dont Import Java Lang	MAINTAINABILITY	pmd:DontImportJavaLang
Empty Finalizer	MAINTAINABILITY	pmd:EmptyFinalizer
Empty Finally Block	MAINTAINABILITY	pmd:EmptyFinallyBlock
Empty If Stmt	MAINTAINABILITY	pmd:EmptyIfStmt
Empty Static Initializer	MAINTAINABILITY	pmd:EmptyStaticInitializer
Empty Switch Statements	MAINTAINABILITY	pmd:EmptySwitchStatements
Empty Synchronized Block	MAINTAINABILITY	pmd:EmptySynchronizedBlock
Empty Try Block	MAINTAINABILITY	pmd:EmptyTryBlock
Empty While Stmt	MAINTAINABILITY	pmd:EmptyWhileStmt
Executable Statement Count	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.sizes.ExecutableStatementCountCheck
Experimental - Method superfluously delegates to parent class method	MAINTAINABILITY	findbugs:USM_USELESS_SUBCLASS_METHOD
Experimental - Missing expected or desired warning from FindBugs	MAINTAINABILITY	findbugs:FB_MISSING_EXPECTED_WARNING
Experimental - Unexpected/undesired warning from FindBugs	MAINTAINABILITY	findbugs:FB_UNEXPECTED_WARNING
Field names should start with a lower case letter	MAINTAINABILITY	findbugs:NM_FIELD_NAMING_CONVENTION
File Length	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.sizes.FileLengthCheck
File Tab Character	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.whitespace.FileTabCharacterCheck

Final Class	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.design.FinalClassCheck
Generic Whitespace	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.whitespace.GenericWhitespaceCheck
Hide Utility Class Constructor	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.design.HideUtilityClassConstructorCheck
Idempotent Operations	MAINTAINABILITY	pmd:IdempotentOperations
Illegal Type	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.IllegalTypeCheck
Import Order	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.imports.ImportOrderCheck
Indentation	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.indentation.IndentationCheck
Inner Type Last	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.design.InnerTypeLastCheck
Javadoc Method	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.javadoc.JavadocMethodCheck
Javadoc Package	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.javadoc.JavadocPackageCheck
Javadoc Style	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.javadoc.JavadocStyleCheck
Javadoc Type	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.javadoc.JavadocTypeCheck
Javadoc Variable	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.javadoc.JavadocVariableCheck
JavaNCSS	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.metrics.JavaNCSSCheck
Left Curly	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.blocks.LeftCurlyCheck
Line Length	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.sizes.LineLengthCheck
Local Final Variable Name	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.naming.LocalFinalVariableNameCheck
Method Count	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.sizes.MethodCountCheck
Method Length	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.sizes.MethodLengthCheck
Method Name	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.naming.MethodNameCheck
Method names should start with a lower case letter	MAINTAINABILITY	findbugs:NM_METHOD_NAMING_CONVENTION
Method Param Pad	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.whitespace.MethodParamPadCheck
Method Type(Generic) Parameter Name	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.naming.MethodTypeParameterNameCheck
Missing Deprecated	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.annotation.MissingDeprecatedCheck
Missing Static Method In Non Instantiatable Class	MAINTAINABILITY	pmd:MissingStaticMethodInNonInstantiatableClass
Modifier Order	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.modifier.ModifierOrderCheck
Multiple Variable Declarations	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.MultipleVariableDeclarationsCheck

Naming - Avoid dollar signs	MAINTAINABILITY	pmd:AvoidDollarSigns
Naming - Class naming conventions	MAINTAINABILITY	pmd:ClassNamingConventions
Naming - Method with same name as enclosing class	MAINTAINABILITY	pmd:MethodWithSameNameAsEnclosingClass
Naming - Suspicious constant field name	MAINTAINABILITY	pmd:SuspiciousConstantFieldName
Naming - Suspicious equals method name	MAINTAINABILITY	pmd:SuspiciousEqualsMethodName
Naming - Suspicious Hashcode method name	MAINTAINABILITY	pmd:SuspiciousHashcodeMethodName
Ncss Method Count	MAINTAINABILITY	pmd:NcssMethodCount
Ncss Type Count	MAINTAINABILITY	pmd:NcssTypeCount
No Whitespace Before	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.whitespace.NoWhitespaceBeforeCheck
One Statement Per Line	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.OneStatementPerLineCheck
Operator Wrap	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.whitespace.OperatorWrapCheck
Outer Type Filename	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.OuterTypeFilenameCheck
Outer Type Number	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.sizes.OuterTypeNumberCheck
Package Annotation	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.annotation.PackageAnnotationCheck
Package Declaration	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.PackageDeclarationCheck
Package name	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.naming.PackageNameCheck
Parameter Name	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.naming.ParameterNameCheck
Paren Pad	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.whitespace.ParenPadCheck
Performance - Private method is never called	MAINTAINABILITY	findbugs:UPM_UNCALLED_PRIVATE_METHOD
Performance - Unread field	MAINTAINABILITY	findbugs:URF_UNREAD_FIELD
Performance - Unread field: should this field be static?	MAINTAINABILITY	findbugs:SS_SHOULD_BE_STATIC
Performance - Unused field	MAINTAINABILITY	findbugs:UUF_UNUSED_FIELD
Redundant import	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.imports.RedundantImportCheck
Redundant Modifier	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.modifier.RedundantModifierCheck
Redundant Throws	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.RedundantThrowsCheck
Reversed method arguments	MAINTAINABILITY	findbugs:DMI_ARGUMENTS_WRONG_ORDER

Right Curly	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.blocks.RightCurlyCheck
Signature Declare Throws Exception	MAINTAINABILITY	pmd:SignatureDeclareThrowsException
Signature Declare Throws Exception (With Type Resolution)	MAINTAINABILITY	pmd:SignatureDeclareThrowsExceptionWithTypeResolution
Simplify Boolean Expression	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.SimplifyBooleanExpressionCheck
Simplify Boolean Return	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.SimplifyBooleanReturnCheck
Simplify Conditional	MAINTAINABILITY	pmd:SimplifyConditional
Singular Field	MAINTAINABILITY	pmd:SingularField
String To String	MAINTAINABILITY	pmd:StringToString
Suppress Warnings	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.annotation.SuppressWarningsCheck
Trailing Comment	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.TrailingCommentCheck
Typecast Paren Pad	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.whitespace.TypecastParenPadCheck
Unconditional If Statement	MAINTAINABILITY	pmd:UnconditionalIfStatement
Unnecessary Parentheses	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.UnnecessaryParenthesesCheck
Unused formal parameter	MAINTAINABILITY	pmd:UnusedFormalParameter
Unused Imports	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.imports.UnusedImportsCheck
Unused local variable	MAINTAINABILITY	pmd:UnusedLocalVariable
Unused Modifier	MAINTAINABILITY	pmd:UnusedModifier
Unused Private Field	MAINTAINABILITY	pmd:UnusedPrivateField
Unused private method	MAINTAINABILITY	pmd:UnusedPrivateMethod
Upper Ell	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.UpperEllCheck
Use String Buffer Length	MAINTAINABILITY	pmd:UseStringBufferLength
Useless String Value Of	MAINTAINABILITY	pmd:UselessStringValueOf
Whitespace After	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.whitespace.WhitespaceAfterCheck
Whitespace Around	MAINTAINABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.whitespace.WhitespaceAroundCheck
Avoid Assert As Identifier	PORTABILITY	pmd:AvoidAssertAsIdentifier
Avoid Enum As Identifier	PORTABILITY	pmd:AvoidEnumAsIdentifier
Bad practice - Class is Serializable, but doesn't define serialVersionUID	PORTABILITY	findbugs:SE_NO_SERIALVERSIONID

Bad practice - Use of identifier that is a keyword in later versions of Java	PORTABILITY	findbugs:NM_FUTURE_KEYWORD_USED_AS_IDENTIFIER
Bad practice - Use of member identifier that is a keyword in later versions of Java	PORTABILITY	findbugs:NM_FUTURE_KEYWORD_USED_AS_MEMBER_IDENTIFIER
Correctness - File.separator used for regular expression	PORTABILITY	findbugs:RE_CANT_USE_FILE_SEPARATOR_AS_REGULAR_EXPRESSION
Dodgy - Code contains a hard coded reference to an absolute pathname	PORTABILITY	findbugs:DMI_HARDCODED_ABSOLUTE_FILENAME
Dodgy - Method directly allocates a specific implementation of xml interfaces	PORTABILITY	findbugs:XFB_XML_FACTORY_BYPASS
Dont Import Sun	PORTABILITY	pmd:DontImportSun
Experimental - Potential lost logger changes due to weak reference in OpenJDK	PORTABILITY	findbugs:LG_LOST_LOGGER_DUE_TO_WEAK_REFERENCE
Internationalization - Consider using Locale parameterized version of invoked method	PORTABILITY	findbugs:DM_CONVERT_CASE
Newline At End Of File	PORTABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.NewlineAtEndOfFileCheck
Reliance on default encoding	PORTABILITY	findbugs:DM_DEFAULT_ENCODING
Replace Enumeration With Iterator	PORTABILITY	pmd:ReplaceEnumerationWithIterator
System Println	PORTABILITY	pmd:SystemPrintln
Adding elements of an entry set may fail due to reuse of Entry objects	RELIABILITY	findbugs:DMI_ENTRY_SETS_MAY_REUSE_ENTRY_OBJECTS
An increment to a volatile field isn't atomic	RELIABILITY	findbugs:VO_VOLATILE_INCREMENT
Avoid Calling Finalize	RELIABILITY	pmd:AvoidCallingFinalize
Avoid Catching NPE	RELIABILITY	pmd:AvoidCatchingNPE
Avoid Catching Throwable	RELIABILITY	pmd:AvoidCatchingThrowable
Avoid Decimal Literals In Big Decimal Constructor	RELIABILITY	pmd:AvoidDecimalLiteralsInBigDecimalConstructor
Avoid Print Stack Trace	RELIABILITY	pmd:AvoidPrintStackTrace
Avoid Rethrowing Exception	RELIABILITY	pmd:AvoidRethrowingException
Avoid Star Import	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.imports.AvoidStarImportCheck

Avoid Static Import	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.imports.AvoidStaticImportCheck
Avoid Throwing Raw Exception Types	RELIABILITY	pmd:AvoidThrowingRawExceptionTypes
Bad comparison of int value with long constant	RELIABILITY	findbugs:INT_BAD_COMPARISON_WITH_INT_VALUE
Bad practice - Abstract class defines covariant compareTo() method	RELIABILITY	findbugs:CO_ABSTRACT_SELF
Bad practice - Abstract class defines covariant equals() method	RELIABILITY	findbugs:EQ_ABSTRACT_SELF
Bad practice - Certain swing methods needs to be invoked in Swing thread	RELIABILITY	findbugs:SW_SWING_METHODS_INVOKED_IN_SWING_THREAD
Bad practice - Check for sign of bitwise operation	RELIABILITY	findbugs:BIT_SIGNED_CHECK
Bad practice - Class defines compareTo(...) and uses Object.equals()	RELIABILITY	findbugs:EQ_COMPARETO_USE_OBJECT_EQUALS
Bad practice - Class defines equals() and uses Object.hashCode()	RELIABILITY	findbugs:HE_EQUALS_USE_HASHCODE
Bad practice - Class defines equals() but not hashCode()	RELIABILITY	findbugs:HE_EQUALS_NO_HASHCODE
Bad practice - Class defines hashCode() and uses Object.equals()	RELIABILITY	findbugs:HE_HASHCODE_USE_OBJECT_EQUALS
Bad practice - Class defines hashCode() but not equals()	RELIABILITY	findbugs:HE_HASHCODE_NO_EQUALS
Bad practice - Class inherits equals() and uses Object.hashCode()	RELIABILITY	findbugs:HE_INHERITS_EQUALS_USE_HASHCODE
Bad practice - Class is Externalizable but doesn't define a void constructor	RELIABILITY	findbugs:SE_NO_SUITABLE_CONSTRUCTOR_FOR_EXTERNALIZATION
Bad practice - Class is Serializable but its superclass doesn't define a void constructor	RELIABILITY	findbugs:SE_NO_SUITABLE_CONSTRUCTOR
Bad practice - clone method does not call super.clone()	RELIABILITY	findbugs:CN_IDIOM_NO_SUPER_CALL
Bad practice - Clone method may return null	RELIABILITY	findbugs:NP_CLONE_COULD_RETURN_NULL
Bad practice - Comparator doesn't implement Serializable	RELIABILITY	findbugs:SE_COMPARATOR_SHOULD_BE_SERIALIZABLE

Bad practice - Comparison of String objects using == or !=	RELIABILITY	findbugs:ES_COMPARING_STRINGS_WITH_EQ
Bad practice - Comparison of String parameter using == or !=	RELIABILITY	findbugs:ES_COMPARING_PARAMETER_STRING_WITH_EQ
Bad practice - Covariant compareTo() method defined	RELIABILITY	findbugs:CO_SELF_NO_OBJECT
Bad practice - Covariant equals() method defined	RELIABILITY	findbugs:EQ_SELF_NO_OBJECT
Bad practice - Creates an empty jar file entry	RELIABILITY	findbugs:AM_CREATES_EMPTY_JAR_FILE_ENTRY
Bad practice - Creates an empty zip file entry	RELIABILITY	findbugs:AM_CREATES_EMPTY_ZIP_FILE_ENTRY
Bad practice - Dubious catching of IllegalMonitorStateException	RELIABILITY	findbugs:IMSE_DONT_CATCH_IMSE
Bad practice - Equals checks for noncompatible operand	RELIABILITY	findbugs:EQ_CHECK_FOR_OPERAND_NOT_COMPATIBLE_WITH_THIS
Bad practice - equals method fails for subtypes	RELIABILITY	findbugs:EQ_GETCLASS_AND_CLASS_CONSTANT
Bad practice - Equals method should not assume anything about the type of its argument	RELIABILITY	findbugs:BC_EQUALS_METHOD_SHOULD_WORK_FOR_ALL_OBJECTS
Bad practice - equals() method does not check for null argument	RELIABILITY	findbugs:NP_EQUALS_SHOULD_HANDLE_NULL_ARGUMENT
Bad practice - Explicit invocation of finalizer	RELIABILITY	findbugs:FI_EXPLICIT_INVOCATION
Bad practice - Finalizer does not call superclass finalizer	RELIABILITY	findbugs:FI_MISSING_SUPER_CALL
Bad practice - Finalizer nullifies superclass finalizer	RELIABILITY	findbugs:FI_NULLIFY_SUPER
Bad practice - Iterator next() method can't throw NoSuchElementException	RELIABILITY	findbugs:IT_NO_SUCH_ELEMENT
Bad practice - Method ignores exceptional return value	RELIABILITY	findbugs:RV_RETURN_VALUE_IGNORED_BAD_PRACTICE
Bad practice - Method ignores results of InputStream.read()	RELIABILITY	findbugs:RR_NOT_CHECKED
Bad practice - Method ignores results of	RELIABILITY	findbugs:SR_NOT_CHECKED

InputStream.skip()		
Bad practice - Method may fail to close database resource	RELIABILITY	findbugs:ODR_OPEN_DATABASE_RESOURCE
Bad practice - Method may fail to close database resource on exception	RELIABILITY	findbugs:ODR_OPEN_DATABASE_RESOURCE_EXCEPTION_PATH
Bad practice - Method may fail to close stream	RELIABILITY	findbugs:OS_OPEN_STREAM
Bad practice - Method may fail to close stream on exception	RELIABILITY	findbugs:OS_OPEN_STREAM_EXCEPTION_PATH
Bad practice - Method might drop exception	RELIABILITY	findbugs:DE_MIGHT_DROP
Bad practice - Method might ignore exception	RELIABILITY	findbugs:DE_MIGHT_IGNORE
Bad practice - Method with Boolean return type returns explicit null	RELIABILITY	findbugs:NP_BOOLEAN_RETURN_NULL
Bad practice - Non-serializable class has a serializable inner class	RELIABILITY	findbugs:SE_BAD_FIELD_INNER_CLASS
Bad practice - Non-serializable value stored into instance field of a serializable class	RELIABILITY	findbugs:SE_BAD_FIELD_STORE
Bad practice - Serializable inner class	RELIABILITY	findbugs:SE_INNER_CLASS
Bad practice - Static initializer creates instance before all static final fields assigned	RELIABILITY	findbugs:SI_INSTANCE_BEFORE_FINALS_ASSIGNED
Bad practice - Store of non serializable object into HttpSession	RELIABILITY	findbugs:J2EE_STORE_OF_NON_SERIALIZABLE_OBJECT_INTO_SESSION
Bad practice - Suspicious reference comparison	RELIABILITY	findbugs:RC_REF_COMPARISON
Bad practice - The readResolve method must be declared with a return type of Object.	RELIABILITY	findbugs:SE_READ_RESOLVE_MUST_RETURN_OBJECT
Bad practice - toString method may return null	RELIABILITY	findbugs:NP_TOSTRING_COULD_RETURN_NULL
Bad practice - Transient field that isn't set by deserialization.	RELIABILITY	findbugs:SE_TRANSIENT_FIELD_NOT_RESTORED

Bad practice - Usage of GetResource may be unsafe if class is extended	RELIABILITY	findbugs:UI_INHERITANCE_UNSAFE_GETRESOURCE
BigDecimal constructed from double that isn't represented precisely	RELIABILITY	findbugs:DMI_BIGDECIMAL_CONSTRUCTED_FROM_DOUBLE
Broken Null Check	RELIABILITY	pmd:BrokenNullCheck
Class Cast Exception With To Array	RELIABILITY	pmd:ClassCastExceptionWithToArray
Class defines equal(Object); should it be equals(Object)?	RELIABILITY	findbugs:NM_BAD_EQUAL
Class defines hashcode(); should it be hashCode()?	RELIABILITY	findbugs:NM_LCASE_HASHCODE
Class defines tostring(); should it be toString()?	RELIABILITY	findbugs:NM_LCASE_TOSTRING
Close Resource	RELIABILITY	pmd:CloseResource
Code checks for specific values returned by compareTo	RELIABILITY	findbugs:RV_CHECK_COMPARETO_FOR_SPECIFIC_RETURN_VALUE
Compare Objects With Equals	RELIABILITY	pmd:CompareObjectsWithEquals
compareTo()/compare() returns Integer.MIN_VALUE	RELIABILITY	findbugs:CO_COMPARETO_RESULTS_MIN_VALUE
Comparing values with incompatible type qualifiers	RELIABILITY	findbugs:TQ_COMPARING_VALUES_WITH_INCOMPATIBLE_TYPE_QUALIFIERS
Constructor Calls Overridable Method	RELIABILITY	pmd:ConstructorCallsOverridableMethod
Correctness - "." used for regular expression	RELIABILITY	findbugs:RE_POSSIBLE_UNINTENDED_PATTERN
Correctness - A collection is added to itself	RELIABILITY	findbugs:IL_CONTAINER_ADDED_TO_ITSELF
Correctness - A parameter is dead upon entry to a method but overwritten	RELIABILITY	findbugs:IP_PARAMETER_IS_DEAD_BUT_OVERWRITTEN
Correctness - An apparent infinite loop	RELIABILITY	findbugs:IL_INFINITE_LOOP
Correctness - An apparent infinite recursive loop	RELIABILITY	findbugs:IL_INFINITE_RECURSIVE_LOOP
Correctness - Apparent method/constructor confusion	RELIABILITY	findbugs:NM_METHOD_CONSTRUCTOR_CONFUSION
Correctness - Array formatted in useless way using format string	RELIABILITY	findbugs:VA_FORMAT_STRING_BAD_CONVERSION_FROM_ARRAY

Correctness - Bad attempt to compute absolute value of signed 32-bit hashcode	RELIABILITY	findbugs:RV_ABSOLUTE_VALUE_OF_HASHCODE
Correctness - Bad attempt to compute absolute value of signed 32-bit random integer	RELIABILITY	findbugs:RV_ABSOLUTE_VALUE_OF_RANDOM_INT
Correctness - Bad comparison of nonnegative value with negative constant	RELIABILITY	findbugs:INT_BAD_COMPARISON_WITH_NONNEGATIVE_VALUE
Correctness - Bad comparison of signed byte	RELIABILITY	findbugs:INT_BAD_COMPARISON_WITH_SIGNED_BYTE
Correctness - Bad constant value for month	RELIABILITY	findbugs:DMI_BAD_MONTH
Correctness - Bitwise add of signed byte value	RELIABILITY	findbugs:BIT_ADD_OF_SIGNED_BYTE
Correctness - Bitwise OR of signed byte value	RELIABILITY	findbugs:BIT_IOR_OF_SIGNED_BYTE
Correctness - Call to equals() comparing different interface types	RELIABILITY	findbugs:EC_UNRELATED_INTERFACES
Correctness - Call to equals() comparing different types	RELIABILITY	findbugs:EC_UNRELATED_TYPES
Correctness - Call to equals() comparing unrelated class and interface	RELIABILITY	findbugs:EC_UNRELATED_CLASS_AND_INTERFACE
Correctness - Check for sign of bitwise operation	RELIABILITY	findbugs:BIT_SIGNED_CHECK_HIGH_BIT
Correctness - Check to see if ((...) & 0) == 0	RELIABILITY	findbugs:BIT_AND_ZZ
Correctness - Class overrides a method implemented in super class Adapter wrongly	RELIABILITY	findbugs:BOA_BADLY_OVERRIDDEN_ADAPTER
Correctness - close() invoked on a value that is always null	RELIABILITY	findbugs:NP_CLOSING_NULL
Correctness - Collections should not contain themselves	RELIABILITY	findbugs:DMI_COLLECTIONS_SHOULD_NOT_CONTAIN_THEMSELVES
Correctness - Creation of ScheduledThreadPoolExecutor with zero core threads	RELIABILITY	findbugs:DMI_SCHEDULED_THREAD_POOL_EXECUTOR_WITH_ZERO_CORE_THREADS

Correctness - Deadly embrace of non-static inner class and thread local	RELIABILITY	findbugs:SIC_THREADLOCAL_DEADLY_EMBRACE
Correctness - Don't use removeAll to clear a collection	RELIABILITY	findbugs:DMI_USING_REMOVEALL_TO_CLEAR_COLLECTION
Correctness - Doomed attempt to append to an object output stream	RELIABILITY	findbugs:IO_APPENDING_TO_OBJECT_OUTPUT_STREAM
Correctness - Doomed test for equality to NaN	RELIABILITY	findbugs:FE_TEST_IF_EQUAL_TO_NOT_A_NUMBER
Correctness - Double.longBitsToDouble invoked on an int	RELIABILITY	findbugs:DMI_LONG_BITS_TO_DOUBLE_INVOKED_ON_INT
Correctness - equals method always returns false	RELIABILITY	findbugs:EQ_ALWAYS_FALSE
Correctness - equals method always returns true	RELIABILITY	findbugs:EQ_ALWAYS_TRUE
Correctness - equals method compares class names rather than class objects	RELIABILITY	findbugs:EQ_COMPARING_CLASS_NAMES
Correctness - equals method overrides equals in superclass and may not be symmetric	RELIABILITY	findbugs:EQ_OVERRIDING_EQUALS_NOT_SYMMETRIC
Correctness - equals() method defined that doesn't override equals(Object)	RELIABILITY	findbugs:EQ_OTHER_NO_OBJECT
Correctness - equals() method defined that doesn't override Object.equals(Object)	RELIABILITY	findbugs:EQ_OTHER_USE_OBJECT
Correctness - equals() used to compare array and nonarray	RELIABILITY	findbugs:EC_ARRAY_AND_NONARRAY
Correctness - equals(...) used to compare incompatible arrays	RELIABILITY	findbugs:EC_INCOMPATIBLE_ARRAY_COMPARE
Correctness - Exception created and dropped rather than thrown	RELIABILITY	findbugs:RV_EXCEPTION_NOT_THROWN
Correctness - Field not initialized in constructor	RELIABILITY	findbugs:UWF_FIELD_NOT_INITIALIZED_IN_CONSTRUCTOR
Correctness - Format string placeholder incompatible with passed argument	RELIABILITY	findbugs:VA_FORMAT_STRING_BAD_ARGUMENT
Correctness - Format string references missing argument	RELIABILITY	findbugs:VA_FORMAT_STRING_MISSING_ARGUMENT

Correctness - Futile attempt to change max pool size of ScheduledThreadPoolExecutor	RELIABILITY	findbugs:DMI_FUTILE_ATTEMPT_TO_CHANGE_MAXPOOL_SIZE_OF_SCHEDULED_THREAD_POOL_EXECUTOR
Correctness - hasNext method invokes next	RELIABILITY	findbugs:DMI_CALLING_NEXT_FROM_HASNEXT
Correctness - Illegal format string	RELIABILITY	findbugs:VA_FORMAT_STRING_ILLEGAL
Correctness - Impossible cast	RELIABILITY	findbugs:BC_IMPOSSIBLE_CAST
Correctness - Impossible downcast	RELIABILITY	findbugs:BC_IMPOSSIBLE_DOWNCAST
Correctness - Impossible downcast of toArray() result	RELIABILITY	findbugs:BC_IMPOSSIBLE_DOWNCAST_OF_TOARRAY
Correctness - Incompatible bit masks (BIT_AND)	RELIABILITY	findbugs:BIT_AND
Correctness - Incompatible bit masks (BIT_IOR)	RELIABILITY	findbugs:BIT_IOR
Correctness - instanceof will always return false	RELIABILITY	findbugs:BC_IMPOSSIBLE_INSTANCEOF
Correctness - int value cast to double and then passed to Math.ceil	RELIABILITY	findbugs:ICAST_INT_CAST_TO_DOUBLE_PASSED_TO_CEIL
Correctness - int value cast to float and then passed to Math.round	RELIABILITY	findbugs:ICAST_INT_CAST_TO_FLOAT_PASSED_TO_ROUND
Correctness - Integer multiply of result of integer remainder	RELIABILITY	findbugs:IM_MULTIPLYING_RESULT_OF_IREM
Correctness - Integer remainder modulo 1	RELIABILITY	findbugs:INT_BAD_REM_BY_1
Correctness - Integer shift by an amount not in the range 0..31	RELIABILITY	findbugs:ICAST_BAD_SHIFT_AMOUNT
Correctness - Invalid syntax for regular expression	RELIABILITY	findbugs:RE_BAD_SYNTAX_FOR_REGULAR_EXPRESSION
Correctness - Invocation of equals() on an array, which is equivalent to ==	RELIABILITY	findbugs:EC_BAD_ARRAY_COMPARE
Correctness - Invocation of hashCode on an array	RELIABILITY	findbugs:DMI_INVOKING_HASHCODE_ON_ARRAY
Correctness - Invocation of toString on an anonymous array	RELIABILITY	findbugs:DMI_INVOKING_TOSTRING_ON_ANONYMOUS_ARRAY
Correctness - Invocation of toString on	RELIABILITY	findbugs:DMI_INVOKING_TOSTRING_ON_ARRAY

an array		
Correctness - JUnit assertion in run method will not be noticed by JUnit	RELIABILITY	findbugs:IJU_ASSERT_METHOD_INVOKED_FROM_RUN_METHOD
Correctness - MessageFormat supplied where printf style format expected	RELIABILITY	findbugs:VA_FORMAT_STRING_EXPECTED_MESSAGE_FORMAT_SUPPLIED
Correctness - Method assigns boolean literal in boolean expression	RELIABILITY	findbugs:QBA_QUESTIONABLE_BOOLEAN_ASSIGNMENT
Correctness - Method attempts to access a prepared statement parameter with index 0	RELIABILITY	findbugs:SQL_BAD_PREPARED_STATEMENT_ACCESS
Correctness - Method attempts to access a result set field with index 0	RELIABILITY	findbugs:SQL_BAD_RESULTSET_ACCESS
Correctness - Method defines a variable that obscures a field	RELIABILITY	findbugs:MF_METHOD_MASKS_FIELD
Correctness - Method does not check for null argument	RELIABILITY	findbugs:NP_ARGUMENT_MIGHT_BE_NULL
Correctness - Method doesn't override method in superclass due to wrong package for parameter	RELIABILITY	findbugs:NM_WRONG_PACKAGE
Correctness - Method ignores return value	RELIABILITY	findbugs:RV_RETURN_VALUE_IGNORED2
Correctness - Method ignores return value	RELIABILITY	findbugs:RV_RETURN_VALUE_IGNORED
Correctness - Method may return null, but is declared @NonNull	RELIABILITY	findbugs:NP_NONNULL_RETURN_VIOLATION
Correctness - Method must be private in order for serialization to work	RELIABILITY	findbugs:SE_METHOD_MUST_BE_PRIVATE
Correctness - Method performs math using floating point precision	RELIABILITY	findbugs:FL_MATH_USING_FLOAT_PRECISION
Correctness - More arguments are passed that are actually used in the format string	RELIABILITY	findbugs:VA_FORMAT_STRING_EXTRA_ARGUMENTS_PASSED
Correctness - No previous argument for format string	RELIABILITY	findbugs:VA_FORMAT_STRING_NO_PREVIOUS_ARGUMENT
Correctness - No relationship between generic parameter and method	RELIABILITY	findbugs:GC_UNRELATED_TYPES

argument		
Correctness - Nonsensical self computation involving a field (e.g., x & x)	RELIABILITY	findbugs:SA_FIELD_SELF_COMPUTATION
Correctness - Nonsensical self computation involving a variable (e.g., x & x)	RELIABILITY	findbugs:SA_LOCAL_SELF_COMPUTATION
Correctness - Non-virtual method call passes null for nonnull parameter	RELIABILITY	findbugs:NP_NULL_PARAM_DEREF_NONVIRTUAL
Correctness - Null pointer dereference	RELIABILITY	findbugs:NP_ALWAYS_NULL
Correctness - Null pointer dereference in method on exception path	RELIABILITY	findbugs:NP_ALWAYS_NULL_EXCEPTION
Correctness - Null value is guaranteed to be dereferenced	RELIABILITY	findbugs:NP_GUARANTEED_DEREF
Correctness - Number of format-string arguments does not correspond to number of placeholders	RELIABILITY	findbugs:VA_FORMAT_STRING_ARG_MISMATCH
Correctness - Overwritten increment	RELIABILITY	findbugs:DLS_OVERWRITTEN_INCREMENT
Correctness - Possible null pointer dereference	RELIABILITY	findbugs:NP_NULL_ON_SOME_PATH
Correctness - Possible null pointer dereference in method on exception path	RELIABILITY	findbugs:NP_NULL_ON_SOME_PATH_EXCEPTION
Correctness - Primitive array passed to function expecting a variable number of object arguments	RELIABILITY	findbugs:VA_PRIMITIVE_ARRAY_PASSED_TO_OBJECT_VARARG
Correctness - Primitive value is unboxed and coerced for ternary operator	RELIABILITY	findbugs:BX_UNBOXED_AND_COERCED_FOR_TERNARY_OPERATOR
Correctness - Random value from 0 to 1 is coerced to the integer 0	RELIABILITY	findbugs:RV_01_TO_INT
Correctness - Read of unwritten field	RELIABILITY	findbugs:NP_UNWRITTEN_FIELD
Correctness - Repeated conditional tests	RELIABILITY	findbugs:RpC_REPEATED_CONDITIONAL_TEST
Correctness - Return value of putIfAbsent ignored, value passed to putIfAbsent reused	RELIABILITY	findbugs:RV_RETURN_VALUE_OF_PUTIFABSENT_IGNORED

Correctness - Self assignment of field	RELIABILITY	findbugs:SA_FIELD_SELF_ASSIGNMENT
Correctness - Self comparison of field with itself	RELIABILITY	findbugs:SA_FIELD_SELF_COMPARISON
Correctness - Self comparison of value with itself	RELIABILITY	findbugs:SA_LOCAL_SELF_COMPARISON
Correctness - Signature declares use of unhashable class in hashed construct	RELIABILITY	findbugs:HE_SIGNATURE_DECLARES_HASHING_OF_UNHASHABLE_CLASS
Correctness - Static Thread.interrupted() method invoked on thread instance	RELIABILITY	findbugs:STI_INTERRUPTED_ON_UNKNOWNTHREAD
Correctness - Store of null value into field annotated NonNull	RELIABILITY	findbugs:NP_STORE_INTO_NONNULL_FIELD
Correctness - Suspicious reference comparison of Boolean values	RELIABILITY	findbugs:RC_REF_COMPARISON_BAD_PRACTICE_BOOLEAN
Correctness - Suspicious reference comparison to constant	RELIABILITY	findbugs:RC_REF_COMPARISON_BAD_PRACTICE
Correctness - TestCase defines setUp that doesn't call super.setUp()	RELIABILITY	findbugs:IJU_SETUP_NO_SUPER
Correctness - TestCase defines tearDown that doesn't call super.tearDown()	RELIABILITY	findbugs:IJU_TEARDOWN_NO_SUPER
Correctness - The readResolve method must not be declared as a static method.	RELIABILITY	findbugs:SE_READ_RESOLVE_IS_STATIC
Correctness - The type of a supplied argument doesn't match format specifier	RELIABILITY	findbugs:VA_FORMAT_STRING_BAD_CONVERSION
Correctness - Uninitialized read of field in constructor	RELIABILITY	findbugs:UR_UNINIT_READ
Correctness - Uninitialized read of field method called from constructor of superclass	RELIABILITY	findbugs:UR_UNINIT_READ_CALLED_FROM_SUPER_CONSTRUCTOR
Correctness - Use of class without a hashCode() method in a hashed data structure	RELIABILITY	findbugs:HE_USE_OF_UNHASHABLE_CLASS
Correctness - Using pointer equality to compare different types	RELIABILITY	findbugs:EC_UNRELATED_TYPES_USING_POINTER_EQUALITY
Correctness - Vacuous call to collections	RELIABILITY	findbugs:DMI_VACUOUS_SELF_COLLECTION_CALL
Correctness - Value annotated as	RELIABILITY	findbugs:TQ_ALWAYS_VALUE_USED_WHERE_NEVER_REQUIRED

carrying a type qualifier used where a value that must not carry that qualifier is required		
Correctness - Value annotated as never carrying a type qualifier used where value carrying that qualifier is required	RELIABILITY	findbugs:TQ_NEVER_VALUE_USED_WHERE_ALWAYS_REQUIRED
Correctness - Value is null and guaranteed to be dereferenced on exception path	RELIABILITY	findbugs:NP_GUARANTEED_DEREF_ON_EXCEPTION_PATH
Correctness - Value required to have type qualifier, but marked as unknown	RELIABILITY	findbugs:TQ_EXPLICIT_UNKNOWN_SOURCE_VALUE_REACHES_ALWAYS_SINK
Correctness - Value required to not have type qualifier, but marked as unknown	RELIABILITY	findbugs:TQ_EXPLICIT_UNKNOWN_SOURCE_VALUE_REACHES_NEVER_SINK
Correctness - Value that might carry a type qualifier is always used in a way prohibits it from having that type qualifier	RELIABILITY	findbugs:TQ_MAYBE_SOURCE_VALUE_REACHES_NEVER_SINK
Correctness - Value that might not carry a type qualifier is always used in a way requires that type qualifier	RELIABILITY	findbugs:TQ_MAYBE_SOURCE_VALUE_REACHES_ALWAYS_SINK
Covariant Equals	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.CovariantEqualsCheck
Dead store due to switch statement fall through	RELIABILITY	findbugs:SF_DEAD_STORE_DUE_TO_SWITCH_FALLTHROUGH
Dead store due to switch statement fall through to throw	RELIABILITY	findbugs:SF_DEAD_STORE_DUE_TO_SWITCH_FALLTHROUGH_TO_THROW
Dead store to local variable that shadows field	RELIABILITY	findbugs:DLS_DEAD_LOCAL_STORE_SHADOWS_FIELD
Dodgy - Call to unsupported method	RELIABILITY	findbugs:DMI_UNSUPPORTED_METHOD
Dodgy - Check for oddness that won't work for negative numbers	RELIABILITY	findbugs:IM_BAD_CHECK_FOR_ODD
Dodgy - Class doesn't override equals in superclass	RELIABILITY	findbugs:EQ_DOESNT_OVERRIDE_EQUALS
Dodgy - Class exposes synchronization and semaphores in its public interface	RELIABILITY	findbugs:PS_PUBLIC_SEMAPHORES
Dodgy - Class extends Servlet class and	RELIABILITY	findbugs:MTIA_SUSPECT_SERVLET_INSTANCE_FIELD

uses instance variables		
Dodgy - Class extends Struts Action class and uses instance variables	RELIABILITY	findbugs:MTIA_SUSPECT_STRUTS_INSTANCE_FIELD
Dodgy - Complicated, subtle or wrong increment in for-loop	RELIABILITY	findbugs:QF_QUESTIONABLE_FOR_LOOP
Dodgy - Computation of average could overflow	RELIABILITY	findbugs:IM_AVERAGE_COMPUTATION_COULD_OVERFLOW
Dodgy - Consider returning a zero length array rather than null	RELIABILITY	findbugs:PZLA_PREFER_ZERO_LENGTH_ARRAYS
Dodgy - Dereference of the result of readLine() without nullcheck	RELIABILITY	findbugs:NP_DEREFERENCE_OF_READLINE_VALUE
Dodgy - Double assignment of local variable	RELIABILITY	findbugs:SA_LOCAL_DOUBLE_ASSIGNMENT
Dodgy - Immediate dereference of the result of readLine()	RELIABILITY	findbugs:NP_IMMEDIATE_DEREFERENCE_OF_READLINE
Dodgy - Initialization circularity	RELIABILITY	findbugs:IC_INIT_CIRCULARITY
Dodgy - instanceof will always return true	RELIABILITY	findbugs:BC_VACUOUS_INSTANCEOF
Dodgy - int division result cast to double or float	RELIABILITY	findbugs:ICAST_IDIV_CAST_TO_DOUBLE
Dodgy - Method discards result of readLine after checking if it is nonnull	RELIABILITY	findbugs:RV_DONT_JUST_NULL_CHECK_READLINE
Dodgy - Method uses the same code for two branches	RELIABILITY	findbugs:DB_DUPLICATE_BRANCHES
Dodgy - Method uses the same code for two switch clauses	RELIABILITY	findbugs:DB_DUPLICATE_SWITCH_CLAUSES
Dodgy - Non serializable object written to ObjectOutput	RELIABILITY	findbugs:DMI_NONSERIALIZABLE_OBJECT_WRITTEN
Dodgy - Non-Boolean argument formatted using %b format specifier	RELIABILITY	findbugs:VA_FORMAT_STRING_BAD_CONVERSION_TO_BOOLEAN
Dodgy - Parameter must be nonnull but is marked as nullable	RELIABILITY	findbugs:NP_PARAMETER_MUST_BE_NONNULL_BUT_MARKED_AS_NULLABLE
Dodgy - Possible null pointer dereference due to return value of called method	RELIABILITY	findbugs:NP_NULL_ON_SOME_PATH_FROM_RETURN_VALUE
Dodgy - Possible null pointer dereference	RELIABILITY	findbugs:NP_NULL_ON_SOME_PATH_MIGHT_BE_INFEASIBLE

on path that might be infeasible		
Dodgy - Potentially dangerous use of non-short-circuit logic	RELIABILITY	findbugs:NS_DANGEROUS_NON_SHORT_CIRCUIT
Dodgy - Questionable cast to abstract collection	RELIABILITY	findbugs:BC_BAD_CAST_TO_ABSTRACT_COLLECTION
Dodgy - Questionable cast to concrete collection	RELIABILITY	findbugs:BC_BAD_CAST_TO_CONCRETE_COLLECTION
Dodgy - Questionable use of non-short-circuit logic	RELIABILITY	findbugs:NS_NON_SHORT_CIRCUIT
Dodgy - Remainder of 32-bit signed random integer	RELIABILITY	findbugs:RV_REM_OF_RANDOM_INT
Dodgy - Remainder of hashCode could be negative	RELIABILITY	findbugs:RV_REM_OF_HASHCODE
Dodgy - Result of integer multiplication cast to long	RELIABILITY	findbugs:ICAST_INTEGER_MULTIPLY_CAST_TO_LONG
Dodgy - Self assignment of local variable	RELIABILITY	findbugs:SA_LOCAL_SELF_ASSIGNMENT
Dodgy - Test for floating point equality	RELIABILITY	findbugs:FE_FLOATING_POINT_EQUALITY
Dodgy - Thread passed where Runnable expected	RELIABILITY	findbugs:DMI_THREAD_PASSED_WHERE_RUNNABLE_EXPECTED
Dodgy - Transient field of class that isn't Serializable.	RELIABILITY	findbugs:SE_TRANSIENT_FIELD_OF_NONSERIALIZABLE_CLASS
Dodgy - Unsigned right shift cast to short/byte	RELIABILITY	findbugs:ICAST_QUESTIONABLE_UNSIGNED_RIGHT_SHIFT
Dodgy - Unusual equals method	RELIABILITY	findbugs:EQ_UNUSUAL
Dodgy - Write to static field from instance method	RELIABILITY	findbugs:ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD
D'oh! A nonsensical method invocation	RELIABILITY	findbugs:DMI_DOH
Don't reuse entry objects in iterators	RELIABILITY	findbugs:PZ_DONT_REUSE_ENTRY_OBJECTS_IN_ITERATORS
Empty Block	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.blocks.EmptyBlockCheck
Empty Catch Block	RELIABILITY	pmd:EmptyCatchBlock
Empty For Initializer Pad	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.whitespace.EmptyForInitializerPadCheck
Empty For Iterator Pad	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.whitespace.EmptyForIteratorPadCheck
Empty Statement	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.EmptyStatementCheck

Equals Avoid Null	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.EqualsAvoidNullCheck
Equals Hash Code	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.EqualsHashCodeCheck
Equals Null	RELIABILITY	pmd:EqualsNull
Experimental - Bad Applet Constructor relies on uninitialized AppletStub	RELIABILITY	findbugs:BAC_BAD_APPLET_CONSTRUCTOR
Experimental - Calls to equals on a final class that doesn't override Object's equals method	RELIABILITY	findbugs:UOE_USE_OBJECT_EQUALS
Experimental - Method may fail to clean up stream or resource	RELIABILITY	findbugs:OBL_UNSATISFIED_OBLIGATION
Explicit Initialization	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.ExplicitInitializationCheck
Fall Through	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.FallThroughCheck
Field isn't final but should be refactored to be so	RELIABILITY	findbugs:MS_SHOULD_BE_REFACTORED_TO_BE_FINAL
Final Local Variable	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.FinalLocalVariableCheck
Final Parameters	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.FinalParametersCheck
Finalize Does Not Call Super Finalize	RELIABILITY	pmd:FinalizeDoesNotCallSuperFinalize
Finalize Overloaded	RELIABILITY	pmd:FinalizeOverloaded
Hidden Field	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.HiddenFieldCheck
Illegal Catch	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.IllegalCatchCheck
Illegal Import	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.imports.IllegalImportCheck
Illegal Instantiation	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.IllegalInstantiationCheck
Illegal Throws	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.IllegalThrowsCheck
Inner Assignment	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.InnerAssignmentCheck
int value converted to long and used as absolute time	RELIABILITY	findbugs:ICAST_INT_2_LONG_AS_INSTANT
Method ignores return value, is this OK?	RELIABILITY	findbugs:RV_RETURN_VALUE_IGNORED_INFERRED
Method may fail to clean up stream or resource on checked exception	RELIABILITY	findbugs:OBL_UNSATISFIED_OBLIGATION_EXCEPTION_EDGE
Missing Override	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.annotation.MissingOverrideCheck
Missing Switch Default	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.MissingSwitchDefaultCheck

Modified Control Variable	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.ModifiedControlVariableCheck
Multithreaded correctness - A thread was created using the default empty run method	RELIABILITY	findbugs:DM_USELESS_THREAD
Multithreaded correctness - A volatile reference to an array doesn't treat the array elements as volatile	RELIABILITY	findbugs:VO_VOLATILE_REFERENCE_TO_ARRAY
Multithreaded correctness - Call to static Calendar	RELIABILITY	findbugs:STCAL_INVOKE_ON_STATIC_CALENDAR_INSTANCE
Multithreaded correctness - Call to static DateFormat	RELIABILITY	findbugs:STCAL_INVOKE_ON_STATIC_DATE_FORMAT_INSTANCE
Multithreaded correctness - Class's readObject() method is synchronized	RELIABILITY	findbugs:RS_READOBJECT_SYNC
Multithreaded correctness - Class's writeObject() method is synchronized but nothing else is	RELIABILITY	findbugs:WS_WRITEOBJECT_SYNC
Multithreaded correctness - Condition.await() not in loop	RELIABILITY	findbugs:WA_AWAIT_NOT_IN_LOOP
Multithreaded correctness - Constructor invokes Thread.start()	RELIABILITY	findbugs:SC_START_IN_CTOR
Multithreaded correctness - Empty synchronized block	RELIABILITY	findbugs:ESync_EMPTY_SYNC
Multithreaded correctness - Field not guarded against concurrent access	RELIABILITY	findbugs:IS_FIELD_NOT_GUARDED
Multithreaded correctness - Inconsistent synchronization	RELIABILITY	findbugs:IS_INCONSISTENT_SYNC
Multithreaded correctness - Inconsistent synchronization	RELIABILITY	findbugs:IS2_INCONSISTENT_SYNC
Multithreaded correctness - Incorrect lazy initialization and update of static field	RELIABILITY	findbugs:LI_LAZY_INIT_UPDATE_STATIC
Multithreaded correctness - Incorrect lazy initialization of static field	RELIABILITY	findbugs:LI_LAZY_INIT_STATIC
Multithreaded correctness - Invokes run on a thread (did you mean to start it instead?)	RELIABILITY	findbugs:RU_INVOKE_RUN

Multithreaded correctness - Method calls Thread.sleep() with a lock held	RELIABILITY	findbugs:SWL_SLEEP_WITH_LOCK_HELD
Multithreaded correctness - Method does not release lock on all exception paths	RELIABILITY	findbugs:UL_UNRELEASED_LOCK_EXCEPTION_PATH
Multithreaded correctness - Method does not release lock on all paths	RELIABILITY	findbugs:UL_UNRELEASED_LOCK
Multithreaded correctness - Method spins on field	RELIABILITY	findbugs:SP_SPIN_ON_FIELD
Multithreaded correctness - Method synchronizes on an updated field	RELIABILITY	findbugs:ML_SYNC_ON_UPDATED_FIELD
Multithreaded correctness - Mismatched notify()	RELIABILITY	findbugs:MWN_MISMATCHED_NOTIFY
Multithreaded correctness - Mismatched wait()	RELIABILITY	findbugs:MWN_MISMATCHED_WAIT
Multithreaded correctness - Monitor wait() called on Condition	RELIABILITY	findbugs:DM_MONITOR_WAIT_ON_CONDITION
Multithreaded correctness - Mutable servlet field	RELIABILITY	findbugs:MSF_MUTABLE_SERVLET_FIELD
Multithreaded correctness - Naked notify	RELIABILITY	findbugs:NN_NAKED_NOTIFY
Multithreaded correctness - Possible double check of field	RELIABILITY	findbugs:DC_DOUBLECHECK
Multithreaded correctness - Static Calendar	RELIABILITY	findbugs:STCAL_STATIC_CALENDAR_INSTANCE
Multithreaded correctness - Static DateFormat	RELIABILITY	findbugs:STCAL_STATIC_SIMPLE_DATE_FORMAT_INSTANCE
Multithreaded correctness - Sychronization on getClass rather than class literal	RELIABILITY	findbugs:WL_USING_GETCLASS_RATHER_THAN_CLASS_LITERAL
Multithreaded correctness - Synchronization on Boolean could lead to deadlock	RELIABILITY	findbugs:DL_SYNCHRONIZATION_ON_BOOLEAN
Multithreaded correctness - Synchronization on boxed primitive could lead to deadlock	RELIABILITY	findbugs:DL_SYNCHRONIZATION_ON_BOXED_PRIMITIVE

Multithreaded correctness - Synchronization on boxed primitive values	RELIABILITY	findbugs:DL_SYNCHRONIZATION_ON_UNSHARED_BOXED_PRIMITIVE
Multithreaded correctness - Synchronization on field in futile attempt to guard that field	RELIABILITY	findbugs:ML_SYNC_ON_FIELD_TO_GUARD_CHANGING_THAT_FIELD
Multithreaded correctness - Synchronization on interned String could lead to deadlock	RELIABILITY	findbugs:DL_SYNCHRONIZATION_ON_SHARED_CONSTANT
Multithreaded correctness - Synchronization performed on java.util.concurrent Lock	RELIABILITY	findbugs:JLM_JSR166_LOCK_MONITORENTER
Multithreaded correctness - Synchronize and null check on the same field.	RELIABILITY	findbugs:NP_SYNC_AND_NULL_CHECK_FIELD
Multithreaded correctness - Unconditional wait	RELIABILITY	findbugs:UW_UNCOND_WAIT
Multithreaded correctness - Unsynchronized get method, synchronized set method	RELIABILITY	findbugs:UG_SYNC_SET_UNSYNC_GET
Multithreaded correctness - Using notify() rather than notifyAll()	RELIABILITY	findbugs:NO_NOTIFY_NOT_NOTIFYALL
Multithreaded correctness - Wait not in loop	RELIABILITY	findbugs:WA_NOT_IN_LOOP
Multithreaded correctness - Wait with two locks held	RELIABILITY	findbugs:TLW_TWO_LOCK_WAIT
Mutable Exception	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.design.MutableExceptionCheck
Negating the result of compareTo()/compare()	RELIABILITY	findbugs:RV_NEGATING_RESULT_OF_COMPARETO
No Clone	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.NoCloneCheck
No Finalizer	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.NoFinalizerCheck
Nonnull field is not initialized	RELIABILITY	findbugs:NP_NONNULL_FIELD_NOT_INITIALIZED_IN_CONSTRUCTOR
Non-transient non-serializable instance field in serializable class	RELIABILITY	findbugs:SE_BAD_FIELD
Parameter Assignment	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.ParameterAssignmentCheck

Performance - Explicit garbage collection; extremely dubious except in benchmarking code	RELIABILITY	findbugs:DM_GC
Read of unwritten public or protected field	RELIABILITY	findbugs:NP_UNWRITTEN_PUBLIC_OR_PROTECTED_FIELD
Self assignment of local rather than assignment to field	RELIABILITY	findbugs:SA_LOCAL_SELF_ASSIGNMENT_INSTEAD_OF_FIELD
Sequence of calls to concurrent abstraction may not be atomic	RELIABILITY	findbugs:AT_OPERATION_SEQUENCE_ON_CONCURRENT_ABSTRACTION
String Buffer Instantiation With Char	RELIABILITY	pmd:StringBufferInstantiationWithChar
String Literal Equality	RELIABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.coding.StringLiteralEqualityCheck
Switch statement found where default case is missing	RELIABILITY	findbugs:SF_SWITCH_NO_DEFAULT
Switch statement found where one case falls through to the next case	RELIABILITY	findbugs:SF_SWITCH_FALLTHROUGH
Unchecked/unconfirmed cast of return value from method	RELIABILITY	findbugs:BC_UNCONFIRMED_CAST_OF_RETURN_VALUE
Unread public/protected field	RELIABILITY	findbugs:URF_UNREAD_PUBLIC_OR_PROTECTED_FIELD
Unused public or protected field	RELIABILITY	findbugs:UUF_UNUSED_PUBLIC_OR_PROTECTED_FIELD
Unwritten public or protected field	RELIABILITY	findbugs:UWF_UNWRITTEN_PUBLIC_OR_PROTECTED_FIELD
Useless Operation On Immutable	RELIABILITY	pmd:UselessOperationOnImmutable
Using monitor style wait methods on util.concurrent abstraction	RELIABILITY	findbugs:JML_JSR166_CALLING_WAIT_RATHER_THAN_AWAIT
Absolute path traversal in servlet	SECURITY	findbugs:PT_ABSOLUTE_PATH_TRAVERSAL
Bad practice - Classloaders should only be created inside doPrivileged block	SECURITY	findbugs:DP_CREATE_CLASSLOADER_INSIDE_DO_PRIVILEGED
Bad practice - Method invoked that should be only be invoked inside a doPrivileged block	SECURITY	findbugs:DP_DO_INSIDE_DO_PRIVILEGED
Bad practice - Method invokes dangerous method runFinalizersOnExit	SECURITY	findbugs:DM_RUN_FINALIZERS_ON_EXIT
Bad practice - Method invokes System.exit(...)	SECURITY	findbugs:DM_EXIT
Bad practice - Random object created	SECURITY	findbugs:DMI_RANDOM_USED_ONLY_ONCE

and used only once		
Malicious code vulnerability - Field is a mutable array	SECURITY	findbugs:MS_MUTABLE_ARRAY
Malicious code vulnerability - Field is a mutable Hashtable	SECURITY	findbugs:MS_MUTABLE_HASHTABLE
Malicious code vulnerability - Field isn't final and can't be protected from malicious code	SECURITY	findbugs:MS_CANNOT_BE_FINAL
Malicious code vulnerability - Field isn't final but should be	SECURITY	findbugs:MS_SHOULD_BE_FINAL
Malicious code vulnerability - Field should be both final and package protected	SECURITY	findbugs:MS_FINAL_PKGPROTECT
Malicious code vulnerability - Field should be moved out of an interface and made package protected	SECURITY	findbugs:MS_OOI_PKGPROTECT
Malicious code vulnerability - Field should be package protected	SECURITY	findbugs:MS_PKGPROTECT
Malicious code vulnerability - Finalizer should be protected, not public	SECURITY	findbugs:FI_PUBLIC_SHOULD_BE_PROTECTED
Malicious code vulnerability - May expose internal representation by incorporating reference to mutable object	SECURITY	findbugs:EI_EXPOSE_REP2
Malicious code vulnerability - May expose internal representation by returning reference to mutable object	SECURITY	findbugs:EI_EXPOSE_REP
Malicious code vulnerability - May expose internal static state by storing a mutable object into a static field	SECURITY	findbugs:EI_EXPOSE_STATIC_REP2
Malicious code vulnerability - Public static method may expose internal representation by returning array	SECURITY	findbugs:MS_EXPOSE_REP
Preserve Stack Trace	SECURITY	pmd:PreserveStackTrace
Relative path traversal in servlet	SECURITY	findbugs:PT_RELATIVE_PATH_TRAVERSAL

Security - A prepared statement is generated from a nonconstant String	SECURITY	findbugs:SQL_PREPARED_STATEMENT_GENERATED_FROM_NONCONSTANT_STRING
Security - Array is stored directly	SECURITY	pmd:ArrayIsStoredDirectly
Security - Empty database password	SECURITY	findbugs:DMI_EMPTY_DB_PASSWORD
Security - Hardcoded constant database password	SECURITY	findbugs:DMI_CONSTANT_DB_PASSWORD
Security - HTTP cookie formed from untrusted input	SECURITY	findbugs:HRS_REQUEST_PARAMETER_TO_COOKIE
Security - HTTP Response splitting vulnerability	SECURITY	findbugs:HRS_REQUEST_PARAMETER_TO_HTTP_HEADER
Security - JSP reflected cross site scripting vulnerability	SECURITY	findbugs:XSS_REQUEST_PARAMETER_TO_JSP_WRITER
Security - Nonconstant string passed to execute method on an SQL statement	SECURITY	findbugs:SQL_NONCONSTANT_STRING_PASSED_TO_EXECUTE
Security - Servlet reflected cross site scripting vulnerability	SECURITY	findbugs:XSS_REQUEST_PARAMETER_TO_SEND_ERROR
Security - Servlet reflected cross site scripting vulnerability	SECURITY	findbugs:XSS_REQUEST_PARAMETER_TO_SERVLET_WRITER
Cyclomatic Complexity	TESTABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.metrics.CyclomaticComplexityCheck
Exception As Flow Control	TESTABILITY	pmd:ExceptionAsFlowControl
NPath Complexity	TESTABILITY	checkstyle:com.puppycrawl.tools.checkstyle.checks.metrics.NPathComplexityCheck
Boxing/unboxing to parse a primitive		findbugs:DM_BOXED_PRIMITIVE_FOR_PARSING
Format string should use %n rather than \n		findbugs:VA_FORMAT_STRING_USES_NEWLINE
Method relaxes nullness annotation on return value		findbugs:NP_METHOD_RETURN_RELAXING_ANNOTATION
Method tightens nullness annotation on parameter		findbugs:NP_METHOD_PARAMETER_TIGHTENS_ANNOTATION
Useless increment in return statement		findbugs:DLS_DEAD_LOCAL_INCREMENT_IN_RETURN
Value without a type qualifier used where a value is required to have that qualifier		findbugs:TQ_UNKNOWN_VALUE_USED_WHERE_ALWAYS_STRICTLY_REQUIRED

Bijlage 3: ingevulde NFR's

De volgende NFR's worden volledig ingevuld door de norm:

Code	Requirement
RD-OH-001	De broncode voldoet aan codeerrichtlijnen en wordt geautomatiseerd getest op deze codeerrichtlijnen. De gehanteerde codeerrichtlijnen zijn representatief voor binnen het vakgebied gebruikelijke richtlijnen voor professionele softwareontwikkeling.
RD-BEV-001	95% van de methodes van publieke interfaces is gedocumenteerd.