




SonarQube

 This page was automatically generated and should not be edited.

 The information on this page was provided by outside contributors and has not been verified by SEI CERT.

 The table below can be re-ordered, by clicking column headers.

Tool Version: 6.7

Checker	Guideline
AssignmentInSubExpressionCheck	EXP51-J. Do not perform assignments in conditional expressions
EmptyStatementUsageCheck	MSC51-J. Do not place a semicolon immediately following an if, for, or while condition
HiddenFieldCheck	DCL51-J. Do not shadow or obscure identifiers in subscopes
S106	ERR02-J. Prevent exceptions while logging data
S112	ERR07-J. Do not throw RuntimeException, Exception, or Throwable
S00121	EXP52-J. Use braces for the body of an if, for, or while statement
S128	MSC52-J. Finish every set of statements associated with a case label with a break statement
S818	DCL50-J. Use visually distinct identifiers
S864	EXP53-J. Use parentheses for precedence of operation
S881	EXP05-J. Do not follow a write by a subsequent write or read of the same object within an expression
S899	EXP00-J. Do not ignore values returned by methods
S899	FIO02-J. Detect and handle file-related errors
S1111	MET12-J. Do not use finalizers
S1113	MET12-J. Do not use finalizers
S1114	MET12-J. Do not use finalizers
S1143	ERR04-J. Do not complete abruptly from a finally block
S1147	ERR09-J. Do not allow untrusted code to terminate the JVM
S1163	ERR05-J. Do not let checked exceptions escape from a finally block
S1166	ERR00-J. Do not suppress or ignore checked exceptions
S1168	MET55-J. Return an empty array or collection instead of a null value for methods that return an array or collection
S1174	MET12-J. Do not use finalizers
S1181	ERR08-J. Do not catch NullPointerException or any of its ancestors
S1182	MET53-J. Ensure that the clone() method calls super.clone()
S1193	ERR51-J. Prefer user-defined exceptions over more general exception types
S1206	MET09-J. Classes that define an equals() method must also define a hashCode() method
S1217	THI00-J. Do not invoke Thread.run()
S1313	MSC03-J. Never hard code sensitive information
S1314	DCL50-J. Use visually distinct identifiers
S1444	OBJ10-J. Do not use public static nonfinal fields
S1449	STR02-J. Specify an appropriate locale when comparing locale-dependent data
S1659	DCL52-J. Do not declare more than one variable per declaration
S1696	ERR08-J. Do not catch NullPointerException or any of its ancestors
S1698	EXP03-J. Do not use the equality operators when comparing values of boxed primitives
S1698	EXP50-J. Do not confuse abstract object equality with reference equality

S1699	MET05-J. Ensure that constructors do not call overridable methods
S1854	MSC56-J. Detect and remove superfluous code and values
S1860	LCK01-J. Do not synchronize on objects that may be reused
S1872	OBJ09-J. Compare classes and not class names
S1874	MET02-J. Do not use deprecated or obsolete classes or methods
S1943	STR04-J. Use compatible character encodings when communicating string data between JVMs
S1943	STR50-J. Use the appropriate method for counting characters in a string
S1989	ERR01-J. Do not allow exceptions to expose sensitive information
S2057	SER00-J. Enable serialization compatibility during class evolution
S2059	SER05-J. Do not serialize instances of inner classes
S2061	SER01-J. Do not deviate from the proper signatures of serialization methods
S2066	SER05-J. Do not serialize instances of inner classes
S2068	MSC03-J. Never hard code sensitive information
S2076	IDS07-J. Sanitize untrusted data passed to the Runtime.exec() method
S2077	IDS00-J. Prevent SQL injection
S2078	IDS54-J. Prevent LDAP injection
S2093	ERR54-J. Use a try-with-resources statement to safely handle closeable resources
S2095	FIO04-J. Release resources when they are no longer needed
S2111	NUM10-J. Do not construct BigDecimal objects from floating-point literals
S2151	MET12-J. Do not use finalizers
S2159	EXP02-J. Do not use the Object.equals() method to compare two arrays
S2162	MET08-J. Preserve the equality contract when overriding the equals() method
S2168	LCK10-J. Use a correct form of the double-checked locking idiom
S2175	EXP04-J. Do not pass arguments to certain Java Collections Framework methods that are a different type than the collection parameter type
S2184	NUM50-J. Convert integers to floating point for floating-point operations
S2189	MSC01-J. Do not use an empty infinite loop
S2197	NUM51-J. Do not assume that the remainder operator always returns a nonnegative result for integral operands
S2201	EXP00-J. Do not ignore values returned by methods
S2225	EXP01-J. Do not use a null in a case where an object is required
S2245	MSC02-J. Generate strong random numbers
S2251	MSC54-J. Avoid inadvertent wrapping of loop counters
S2259	EXP01-J. Do not use a null in a case where an object is required
S2274	THI03-J. Always invoke wait() and await() methods inside a loop
S2276	LCK09-J. Do not perform operations that can block while holding a lock
S2278	MSC61-J. Do not use insecure or weak cryptographic algorithms
S2384	OBJ05-J. Do not return references to private mutable class members
S2384	OBJ06-J. Defensively copy mutable inputs and mutable internal components
S2384	OBJ13-J. Ensure that references to mutable objects are not exposed
S2386	OBJ01-J. Limit accessibility of fields
S2386	OBJ13-J. Ensure that references to mutable objects are not exposed
S2390	DCL00-J. Prevent class initialization cycles
S2442	LCK03-J. Do not synchronize on the intrinsic locks of high-level concurrency objects
S2445	LCK00-J. Use private final lock objects to synchronize classes that may interact with untrusted code
S2446	THI02-J. Notify all waiting threads rather than a single thread
S2447	EXP01-J. Do not use a null in a case where an object is required
S2631	IDS08-J. Sanitize untrusted data included in a regular expression

S2637	EXP01-J. Do not use a null in a case where an object is required
S2653	ENV06-J. Production code must not contain debugging entry points
S2674	FIO10-J. Ensure the array is filled when using read() to fill an array
S2681	EXP52-J. Use braces for the body of an if, for, or while statement
S2693	TSM02-J. Do not use background threads during class initialization
S2755	IDS17-J. Prevent XML External Entity Attacks
S2886	VNA01-J. Ensure visibility of shared references to immutable objects
S3011	SEC05-J. Do not use reflection to increase accessibility of classes, methods, or fields
S3014	THI01-J. Do not invoke ThreadGroup methods
S3034	NUM52-J. Be aware of numeric promotion behavior
S3067	LCK02-J. Do not synchronize on the class object returned by getClass()
S3346	EXP06-J. Expressions used in assertions must not produce side effects
S3518	NUM02-J. Ensure that division and remainder operations do not result in divide-by-zero errors
S3649	IDS00-J. Prevent SQL injection
S4435	IDS17-J. Prevent XML External Entity Attacks
S5042	IDS04-J. Safely extract files from ZipInputStream