JT-Bug-Id	Description
Cor 1	BC: Impossible cast
Cor 2	BC: Impossible downcast
Cor 3	BC: Impossible downcast of toArray() result
Cor 4	BC: instanceof will always return false
Cor 5	BIT: Bitwise add of signed byte value
Cor 6	BIT: Incompatible bit masks
Cor 7	BIT: Check to see if ((...) & 0) == 0
Cor 8	BIT: Incompatible bit masks
Cor 9	BIT: Bitwise OR of signed byte value
Cor 10	BIT: Check for sign of bitwise operation
Cor 11	BOA: Class overrides a method implemented in super class Adapter wrongly
Cor 12	BSHIFT: 32 bit int shifted by an amount not in the range -31..31
Cor 13	Bx: Primitive value is unboxed and coerced for ternary operator
Cor 14	Co: compareTo()/compare() returns Integer.MIN_VALUE
Cor 15	DLS: Dead store of class literal
Cor 16	DLS: Overwritten increment
Cor 17	DMI: Reversed method arguments
Cor 18	DMI: Bad constant value for month
Cor 19	DMI: BigDecimal constructed from double that isn't represented precisely
Cor 20	DMI: hasNext method invokes next
Cor 21	DMI: Collections should not contain themselves
Cor 22	DMI: D'oh! A nonsensical method invocation
Cor 23	DMI: Invocation of hashCode on an array
Cor 24	DMI: Double.longBitsToDouble invoked on an int
Cor 25	DMI: Vacuous call to collections
Cor 26	Dm: Can't use reflection to check for presence of annotation without runtime retention
Cor 27	Dm: Futile attempt to change max pool size of ScheduledThreadPoolExecutor
Cor 28	Dm: Creation of ScheduledThreadPoolExecutor with zero core threads
Cor 29	Dm: Useless/vacuous call to EasyMock method
Cor 30	EC: equals() used to compare array and nonarray
Cor 31	EC: Invocation of equals() on an array, which is equivalent to ==
Cor 32	EC: equals(...) used to compare incompatible arrays
Cor 33	EC: Call to equals(null)
Cor 34	EC: Call to equals() comparing unrelated class and interface
Cor 35	EC: Call to equals() comparing different interface types
Cor 36	EC: Call to equals() comparing different types
Cor 37	EC: Using pointer equality to compare different types
Cor 38	Eq: equals method always returns false
Cor 39	Eq: equals method always returns true
Cor 40	Eq: equals method compares class names rather than class objects
Cor 41	Eq: Covariant equals() method defined for enum
Cor 42	Eq: equals() method defined that doesn't override equals(Object)
Cor 43	Eq: equals() method defined that doesn't override Object.equals(Object)
Cor 44	Eq: equals method overrides equals in superclass and may not be symmetric
Cor 45	Eq: Covariant equals() method defined, Object.equals(Object) inherited
Cor 46	FE: Doomed test for equality to NaN
Cor 47	FS: Format string placeholder incompatible with passed argument
Cor 48	FS: The type of a supplied argument doesn't match format specifier

JT-Bug-Id	Description
Cor 49	FS: MessageFormat supplied where printf style format expected
Cor 50	FS: More arguments are passed than are actually used in the format string
Cor 51	FS: Illegal format string
Cor 52	FS: Format string references missing argument
Cor 53	FS: No previous argument for format string
Cor 54	GC: No relationship between generic parameter and method argument
Cor 55	HE: Signature declares use of unhashable class in hashed construct
Cor 56	HE: Use of class without a hashCode() method in a hashed data structure
Cor 57	ICAST: int value converted to long and used as absolute time
Cor 58	ICAST: integral value cast to double and then passed to Math.ceil
Cor 59	ICAST: int value cast to float and then passed to Math.round
Cor 60	IJU: JUnit assertion in run method will not be noticed by JUnit
Cor 61	IJU: TestCase declares a bad suite method
Cor 62	IJU: TestCase has no tests
Cor 63	IJU: TestCase defines setUp that doesn't call super.setUp()
Cor 64	IJU: TestCase implements a non-static suite method
Cor 65	IJU: TestCase defines tearDown that doesn't call super.tearDown()
Cor 66	IL: A collection is added to itself
Cor 67	IL: An apparent infinite loop
Cor 68	IL: An apparent infinite recursive loop
Cor 69	IM: Integer multiply of result of integer remainder
Cor 70	INT: Bad comparison of int value with long constant
Cor 71	INT: Bad comparison of nonnegative value with negative constant
Cor 72	INT: Bad comparison of signed byte
Cor 73	IO: Doomed attempt to append to an object output stream
Cor 74	IP: A parameter is dead upon entry to a method but overwritten
Cor 75	MF: Class defines field that masks a superclass field
Cor 76	MF: Method defines a variable that obscures a field
Cor 77	NP: Null pointer dereference
Cor 78	NP: Null pointer dereference in method on exception path
Cor 79	NP: Method does not check for null argument
Cor 80	NP: close() invoked on a value that is always null
Cor 81	NP: Null value is guaranteed to be dereferenced
Cor 82	NP: Value is null and guaranteed to be dereferenced on exception path
Cor 83	NP: Nonnull field is not initialized
Cor 84	NP: Method call passes null to a nonnull parameter
Cor 85	NP: Method may return null, but is declared @NonNull
Cor 86	NP: A known null value is checked to see if it is an instance of a type
Cor 87	NP: Possible null pointer dereference
Cor 88	NP: Possible null pointer dereference in method on exception path
Cor 89	NP: Method call passes null for nonnull parameter
Cor 90	NP: Method call passes null for nonnull parameter
Cor 91	NP: Non-virtual method call passes null for nonnull parameter
Cor 92	NP: Store of null value into field annotated NonNull
Cor 93	NP: Read of unwritten field
Cor 94	Nm: Class defines equal(Object); should it be equals(Object)?
Cor 95	Nm: Class defines hashcode(); should it be hashCode()?
Cor 96	Nm: Class defines tostring(); should it be toString()?
Cor 97	Nm: Apparent method/constructor confusion

JT-Bug-Id	Description
Cor 98	Nm: Very confusing method names
Cor 99	Nm: Method doesn't override method in superclass due to wrong package for parameter
Cor 100	QBA: Method assigns boolean literal in boolean expression
Cor 101	RC: Suspicious reference comparison
Cor 102	RCN: Nullcheck of value previously dereferenced
Cor 103	RE: Invalid syntax for regular expression
Cor 104	RE: File.separator used for regular expression
Cor 105	RE: "." used for regular expression
Cor 106	RV: Random value from 0 to 1 is coerced to the integer 0
Cor 107	RV: Bad attempt to compute absolute value of signed 32-bit hashcode
Cor 108	RV: Bad attempt to compute absolute value of signed random integer
Cor 109	RV: Code checks for specific values returned by compareTo
Cor 110	RV: Exception created and dropped rather than thrown
Cor 111	RV: Method ignores return value
Cor 112	RpC: Repeated conditional tests
Cor 113	SA: Self assignment of field
Cor 114	SA: Self comparison of field with itself
Cor 115	SA: Nonsensical self computation involving a field (e.g., x & x)
Cor 116	SA: Self assignment of local rather than assignment to field
Cor 117	SA: Self comparison of value with itself
Cor 118	SA: Nonsensical self computation involving a variable (e.g., x & x)
Cor 119	SF: Dead store due to switch statement fall through
Cor 120	SF: Dead store due to switch statement fall through to throw
Cor 121	SIC: Deadly embrace of non-static inner class and thread local
Cor 122	SIO: Unnecessary type check done using instanceof operator
Cor 123	SQL: Method attempts to access a prepared statement parameter with index 0
Cor 124	SQL: Method attempts to access a result set field with index 0
Cor 125	STI: Unneeded use of currentThread() call, to call interrupted()
Cor 126	STI: Static Thread.interrupted() method invoked on thread instance
Cor 127	Se: Method must be private in order for serialization to work
Cor 128	Se: The readResolve method must not be declared as a static method.
Cor 129	TQ: Value annotated as carrying a type qualifier used where a value that must not carry that qualifier is required
Cor 130	TQ: Comparing values with incompatible type qualifiers
Cor 131	TQ: Value that might not carry a type qualifier is always used in a way requires that type qualifier
Cor 132	TQ: Value that might carry a type qualifier is always used in a way prohibits it from having that type qualifier
Cor 133	TQ: Value annotated as never carrying a type qualifier used where value carrying that qualifier is required
Cor 134	TQ: Value without a type qualifier used where a value is required to have that qualifier
Cor 135	UMAC: Uncallable method defined in anonymous class
Cor 136	UR: Uninitialized read of field in constructor
Cor 137	UR: Uninitialized read of field method called from constructor of superclass
Cor 138	USELESS_STRING: Invocation of toString on an unnamed array
Cor 139	USELESS_STRING: Invocation of toString on an array
Cor 140	USELESS_STRING: Array formatted in useless way using format string

JT-Bug-Id	Description
Cor 141	UwF: Field only ever set to null
Cor 142	UwF: Unwritten field
Cor 143	VA: Primitive array passed to function expecting a variable number of object arguments