JAVA CODING GUIDELINES

	Fundamentals
01	There is no code duplication
02	The code is well formatted
03	Make `public static` fields `final` (*to avoid caller changing the value*)
04	Use checked exceptions for recoverable conditions and runtime exceptions for programming errors
05	Favor the use of standard exceptions
06	Check parameters for validity
07	In public classes, use accessor methods, not public fields
08	Minimize the scope of local variables
09	Refer to objects by their interfaces
10	Always override `hashCode` when you override `equals`
11	Use `enums` instead of `int` constants
12	Use `JavaDoc` for classes and methods
13	Use the correct type for the right situation
14	All attributes have appropriate access modifiers (`private`, `protected`, `public`)
15	No overflow or underflow are possible during a computation
16	Each `switch` statement have a `default` case
17	No spelling or grammatical errors in any text printed or displayed
18	All comments are consistent with the code
19	Are there enough comments in the code?
20	When commiting, a relevant comment is used
21	No `System.out.println` statement is used.
22	Make a class `final` and the object immutable where possible.
23	Relevant Unit tests are written
24	Source files are encoded in `UTF-8`
25	Don't reinvent the wheel, use libraries (`commons-utils`, `commons-collections`, etc.)
26	No magic numbers or hard coded values
27	Conditional expressions are easy to understand
	Code Smells
01	Class and methods are small and focus on doing one thing.
02	Parameters List does not exceed 4 parameters
03	There is no dead code (*unused parameters, unneeded files/code*)
	Eclipse configuration*
01	Use an Eclipse Formatter
02	Activate Eclipse Save Actions
	Performance
01	Keep Synchronized Sections Small
02	Beware the performance of string concatenation - use `StringBuilder` instead
03	Avoid long lived objects like `ThreaLocal` and `static` variables holding references to lots of short lived objects.

	Naming Conventions
01	Package standard name : `com.[company].[project_name]`
02	Services : `*Service`
03	Interfaces : `I*`
04	Controllers : `*Controller`
05	Converters : `*Converter`
06	DTOs : `*Dto`
07	Constants classes : `*Constants`, `WebConstants`
08	Custom Exceptions : `*Exception`
09	Repositories : `*Repository`
10	DAO : `I*DAO`, `*DAO`
11	Methods : Method names should be verbs or verb phrases, `run();` `isUpperCase();` `getBackground();` `findTotalCost(int Years);`
12	Unit tests : `[CLASS_NAME]Test`
	Exceptions
01	No return codes are returned, Exceptions are used instead
02	No `null` is returned - empty collections or exceptions are used instead
03	Don't ignore exceptions (*log or re-throw*)
04	`e.printStackTrace()` is never used
05	Do not throw `RuntimeException`, `Exception` or `Throwable`
06	Use `try-with-resources` to safely handle closeable resources
	Input Validation
01	Validate inputs (*for valid data, size, range, boundary conditions, etc.*)
02	Validate output from untrusted objects as input
	Security
01	Make class final if not being used for inheritance
02	Release resources (*Streams, Connections, etc.*) in all cases
03	Purge sensitive information from exceptions (exposing file path, internals of the system, configuration)
04	Do not log highly sensitive informations
05	Avoid dynamic SQL, use prepared statement
06	Guard sensitive data during serialization
07	Encrypt passwords and sensitive data

(*) : https://code-memento.github.io/docs/java/eclipse-workspace.html