JAVA CODING STANDRDS

SUST, Computer Science Faculty, Software Engineering Department, 4th year.

بسم الله الرحمن الرحيم

Sudan University for Science and Technology

Faculty of Computer Science and Information Technology

Department of Software Engineering – 4th year

Software Maintenance Lab H.W

Prepared by:

- Alaa Azhari Ahmed Mohammed.

What are coding standards?:

A set of rules, techniques, and best practices to create cleaner, more readable, more efficient code with minimal errors. They offer a uniform format by which software engineers can use to build sophisticated and highly functional code.

Purpose and scope:

An effective mechanism of institutionalizing production of quality code is to develop programming standard and enforce the same through code reviews. This document delves into some fundamental Java programming techniques and provides a rich collection of coding practices to be followed by JAVA/J2EE based application development teams The best practices are primarily targeted towards improvement in the readability and maintainability of code with keen attention to performance enhancements. By employing such practices the application development team can demonstrate their proficiency, profound knowledge and professionalism.

Advantages of implementing coding standards:

- Offers uniformity to the code created by different engineers.
- Enables the creation of reusable code.
- Makes it easier to detect errors.
- Make code simpler, more readable, and easier to maintain.
- Boost programmer efficiency and generates faster results.

Java coding standards:

- Use descriptive names for all variables, function names, constants, and other identifiers.
- Use single letter identifiers only for the counter in loops.
- Class names start with an upper case letter.
- Variable names start with a lower case letter (Use camelCase style), (Variables include parameters, local variables, and data fields. Exception: use UPPER_CASE for constants - final variables).
- For reasons of encapsulation, fields should not be declared public. All fields should be declared private unless necessary otherwise.
- Method names start with a lower case letter.
- Multi-word identifiers are internally capitalized.
- Do not use hyphens or underscores to separate multi-word identifiers (except for constants, which have all upper case letters).

- Class members must be accessed privately, define the variables as private, protected, or public where you need.
- Never leave a Catch block empty.
- Use comments to explain and document your work.
- Make sure that you have use the fundamental concept of OOP (encapsulation, polymorphism, inheritance).
- Each line should contain at most one statement. While compound statements are statements that contain lists of statements enclosed in braces.
- Use classes, objects, methods to modular the code to be easy to maintain.
- Left and Right braces: The starting brace should be at the end of the conditional and the ending brace must be on a separate line and aligned with the conditional.
- White Space (blank Lines): Blank lines improve readability by setting of sections of code that are logically related.
- Implementation Comments: Java codes should have implementation comments delimited by /*...*/ or //. For commenting out code a double slash i.e. // is recommended, while for multiple or single-line comments given as overview of code, the c-style comments i.e. /* */ should be used. For clarity in code, comments should be followed by a blank line. Code should have four styles of implementation comments as follows and anything that goes against a standard should always be document.
- Every method should include a header at the top of the source code that documents all of the information that is critical to understanding it. Detailed description of the method may include the intent of method i.e. what and why the method does, what a method should be passed as

parameters and what it returns, any Known bugs, exceptions that the method throws, information on visibility decisions., how a method changes the object, pre and post conditions, side effects , dependencies , implementation notes , who should be calling this method , whether the method should or should not be overridden , where to invoke super when overriding , control flow or state dependencies that need to exist before calling this method.

- In the declaration you need to follow this guidelines:
  - Instance variables should be placed in the sequence: First public instance variables, protected, package level with no access modifier and then private.
  - Next the class constructors should be declared.
  - This should be followed by the inner classes, if applicable
  - Class methods should be grouped by functionality rather than by scope or accessibility to make reading and understanding the code easier.
  - Declarations for local variables should be only at the beginning of blocks e.g. at the beginning of a try-catch construct.
- The ‘while’ keyword should appear on its own line, immediately followed by the conditional expression, as same as for loop.
- The DO ..WHILE form of the while construct should appear as: The statement block is placed on the next line. The closing curly brace starts in a new line, indented to match its corresponding opening statement.
- When using the comma operator in the initialization or update clause of a for statement, avoid the complexity of using more than three variables.

- The ‘switch’ construct should use the same layout format as the ‘if’ construct. The ‘switch’ keyword should appear on its own line, immediately followed by its test expression. The keyword ‘switch’ and the parenthesis should be separated by a space.
- In the try/catch construct the ‘try’ keyword should be followed by the open brace in its own line. This is followed by the statement body and the close brace on its own line. This may follow any number of ‘catch’ phrases - consisting of the ‘catch’ keyword and the exception expression on its own line with the ‘catch’ body; followed by the close brace on its own line. Try-catch should be accomplished with finally block to destroys all Objects not required. There should not be any empty try-catch blocks.
- For a good design, the rule is to be as restrictive as possible when setting the visibility of a method. If a method doesn’t have to be private then it should have default access modifier, if it doesn’t have to be default then it should be made protected and if it doesn’t have to protected only then it should be made public.
- Class names should be simple full English descriptor nouns, in mixed case starting with the first letter capitalized and the first letter of each internal word also capitalized.
- Package or default visibility may be used for classes internal to a component while public visibility may be used for other components.
- The Java convention is to name interfaces using mixed case with the first letter of each word capitalized like classes.

Resources and references:

1) https://www.browserstack.com/guide/coding-standards-best-practices

2) https://www2.hawaii.edu/~walbritt/ics211/materials/standard.htm

3) https://nea.gov.bh/Attachments/eServices%20Standards/Java_standards_V1.0.pdf