Guideline Document

(Identify new Bikes)

(https://www.zigwheels.com/)

1. Philomina Nancy Arokiaraj Stephen-890201
2. Poornima Manoharan-890115
3. Shalini Ravi – 890200
4. Shanmugapriya Palanisamy-891373

Table Of Content

1.0 Introduction

The Project Guidelines and Standards (PGS) are a roadmap and toolkit for selecting, identifying, planning, implementing, monitoring, evaluating and closing.The adoption of coding standards results in code consistency, which makes -it easier to understand, develop and maintain the application. In addition by being aware of and following the right coding techniques at a granular level, the programmer can make the code more efficient and performance effective.

2.0 Coding standard

The term coding standard refers to a set of rules related to writing code that have been standardized in some context, but are not enforced by interpreters or compilers. A coding standard defines and seeks to enforce conventions such as programming language subsets, naming conventions, and coding style for the purpose of improving code quality.

As compilers are only required to monitor the code for syntax and constraint errors, the purpose of a coding standard is to prevent errors and unwanted behaviour in the code caused by the programmer writing it

2.1 Enforcement

Coding standards are met with some resistance, because they can be seen as causing extra work, and compliance might force programmers to adopt a style to which they are not accustomed.

Programmers are creatures of habit, which entails that any negatively perceived change requires a convincing rationale or a compulsion.

A coding standard that is not enforced often goes unused. Even if programmers understand and agree with a standard, cutting corners can be tempting or deemed necessary due to time constraints or other inevitabilities. Enforcement can be implemented via automatic enforcement and code reviews.

2.2 Code reviews

Code review refers to peer reviewing code in order to provide quality control. Code reviews are the main cause of aversion to implementing coding standards, as they are time-consuming and potentially confrontational, but the benefits of code reviews are considerable. Code reviews promote sharing of knowledge and actually save resources by weeding out bugs and problematic implementations such as an unmaintainable logical structure that would go unnoticed by a linter.

2.3 Significance of coding standards

While a coding standard helps create more secure and reliable programs, the most significant improvement a coding standard offers is readability. The easier the code is to read, the cheaper it ends up being.

The study was controlled for inherently superior memory by repeating the experiment with randomly placed pieces, which displayed beginner level recall performance in all the players, implying that standardized patterns significantly increase human ability to process visual information

3.0 Naming conventions

Identifier naming is perhaps the most noticeable part of a programmers touch on a piece of code, as it is left completely to the discretion of the person writing the code. Identifiers are used for multiple functional parts of a program:

- Variables
- Methods
- Classes
- Constants

3.1 Variables

- The names of local variables should be in mixed case, starting with a lower-case letter (e.g. packetSize). This also applies to the formal parameters of methods. We recommend not to use names starting with underscore.

- The names of static final constants should be all upper-case words, separated by underscores (e.g. MIN WIDTH). CS-2-25 Avoid names that differ only in case. Never use in the same namespace identifiers that have the same letters, but that are capitalized in different ways

3.2 Methods

Method names are verbs or nouns in mixed-case, starting with a lower-case letter (e.g. size or addElementary)

The following method naming conventions are preferred:

- Methods that convert the type, returning a new object, begin with to (e.g. toString).
- Methods that return a view are called asType (e.g. asList)
- Methods that return a primitive type xxx representation are called xxxValue (e.g. intValue)
- Methods that return a singleton instance are called getInstance.

3.3 Classes

- Class names are nouns, in mixed-case, with an initial upper-case letter and the first letter of each subsequent word capitalized (e.g. CoreFactory).
- Abbreviations and acronyms should be avoided in class names, unless these are more widely used than the full form (e.g. HTML)
- Avoid using class names that appear in the JDK library.
- Class names should not normally be longer than 32 characters, to avoid the need for excessive wrapping of source line.
- Avoid using class names that appear in the JDK library

3.4 Constants

Class constant should be all Uppercase with words separated by Underscores(“-“).

4.0 Code Comment Standard

4.1 Comment

Comments are text notes added to the program to provide explanatory information about the source code. They are used in a programming language to document the program and remind programmers of what tricky things they just did with the code and also helps the later generation for understanding and maintenance of code. All Java source code shall include JavaDoc comments.

It should not be used on consecutive multiple lines for text comments, however, it may be used in single or consecutive multiple lines for commenting out sections of code.

- Header comments at front of each file with purpose, author, date.
- "Paragraph" comments at beginning of each group of code.
- Document "tricks" -- anything unobvious.
- Don't comment code that is already clear.
- Using javadoc is essential for larger projects.

4.2 Code Standard

A coding standard gives a uniform appearance to the codes written by different engineers. It improves readability,and maintainability of the codes and it reduces complexity also.It helps in code reuse and help to detect error easily.

Coding standards help in the development of software programs that are less complex and thereby reduce the errors.

5.0 Indentation:

- Each level should be indented by a tab (8 column tab).
- A line should be limited to a page boundary (code should not wrap around the terminal page). In case of a wrap around, it is often the case that the nesting structure is too complex and the code would be clearer if it were rewritten.
- Insert tabs in variable declarations to align the variables.
- Defining variables in a header file is often a symptom of poor partitioning of code between files.
- Declaring all functions before any are defined allows the implementor to use the top-down ordering without running afoul of the single pass C compilers.
- Indentation refers to the spaces at the beginning of a code line.