Java Standards: A Comparative Study

MOHAMMAD M. A. ABDALLAH[1] and MUSTAFA M. Al-RIFAEE[2]

[1, 2] Al-Zaytoonah University of Jordan, Amman, Jordan

[1]m.abdallah@zuj.edu.jo, [2]m.rifaee@zuj.edu.jo

ABSTRACT

Java language is one of the most usable programming language. This make it used to develop many applications and systems by many different developers. This diversity of developers' skills and backgrounds lead to the Java Programming language standards. There are many rules and guidelines for Java programming language. In this research, we include 20 different java standards that are also the common standards, with their characteristics, specifications, and limitations. Also, a comparative study has been made to provide different perspective to researchers and Java developers to help them choose the best standard for their research or application. Furthermore, this research addresses the tools that can be used to check the format of Java code using different standards that listed in this document.

Keywords: *Java, Standard, Java checking tools.*

1. INTRODUCTION

Rule, regarding to Oxford dictionary means "One of a set of explicit or understood regulations or principles governing conduct or procedure within a particular area of activity". It also must be understood and agreed by everyone who get involved in the area of rules. On the other hand, Standards regarding to the International Organization for Standardization (ISO) [1] "A standard is a document that provides requirements, specifications, guidelines or characteristics that can be used consistently to ensure that materials, products, processes and services are fit for their purpose". Programming Language standards are set of guidelines for a specific programming language that recommend programming style, practices and methods for each aspect of a piece program written in this language.

Java programming language is like any other language; it has some rules, notations, and statements. Also, it may be written and applied in different ways. As a consequence, there is a numerous number of possibilities to write a piece of code in Java. Which make it painful to get back to it and try to read, understand or maintain it. Java Standards are the solution for the previous mentioned problems. These terms aim to make the Java programming language readable, maintainable and well documented by applying some rules and guidelines while writing code. The standards are usually covers the file organization, indentation, comments and documentation and declarations …etc. Usually standards target the human maintainer, peer reviewer, quality assurance engineers, and even the developers.

Standards can be categorized in two categories: mandatory rules and recommended guidelines. Rules are necessary and required coding practices, and it violation cause an error. Guideline is suggested coding practices that have been written to recognize the need for individuality and for common coding practices. Its violation may cause a warning or misinterpretation [2].

Programming languages standard is consists of technical definitions, rules and guidelines that function as instructions for designers/manufacturers and operators/users of Java code [3].

Standards sometimes called conventions or styles. In this research, we use only the term "standard" for all of the three terms to make sure that there will be no misunderstanding.

Not all people are agreed that the standards should be used. Verhas [4] argues that the code standards are not important. Since the developer is the main responsible and owner of the code, and the new software development tools helps in understanding and analyzing the code with being formatted in a certain way.

This paper has five sections. Section two explore the most common Java standards. Sections three is listed the Java tools that is applied the Java Standards. Section three discuss the standards of Java programming languages. Section five is a conclusion with recommendations.

2. JAVA STANDARDS

The first standard for Java programming language were found in 1996 by Sun Microsystems [5], short time after java has been released. The Sun standards (also called Oracle's code standards for the Java Programming Language) document has 8 categories that covers files practice, comments and formatting, naming and declarations and some programming practice with examples.

The Sun standards are more likely a basic guideline for Java. Where it is only cover few areas with very general advices on how to write the Java code. In addition, these standards is out dated, and unaware of billions of code that been generated based on new technologies in Java [6]. Moreover, the coding experience of people are also changed and they become more aware of developing a quality code. Therefore, there are many new Java features and tricks that Sun standards do not mention at all in the standard document. However, Sun standards still useful since they cover the main aspects of Java language features. Consequently, the standards that been released after have been built depending on Sun standards. In the same year, Johnson [7] has put his own standards. But, Johnson Standards' are outdated, too general and very basic that did not address the whole Java features. On the other hand, Reddy [8] has a quite similar standards as Sun. but, he added one category more under the name interface. Also, he added subcategories and new rules.

Netscape's software standards [2] are quite short and general. They aim to make the code readable, maintainable and clearer documented. These rules are too general in some terms, which make them losing the benefits of being a standard! Since they can be interpreting in different ways.

ChiMu guidelines [9] is a mixture of different guidelines from different resources that covers most of the features all other standards do. These guidelines explain in first two chapters some basic concepts of object oriented programming with references in case if the reader want to get more knowledge. Even though, ChiMu guidelines still explain each one of the guidelines. It is quite good but in some cases, they are too detailed.

Macadamian Inc. [10] has made a standards guidelines for both C++ and Java in one documents. They gather the similarities in both programming languages and do one standard guidelines for both of them. It is useful for developers who write in both languages and also useful to write a program that may be converted from one language to another. However, Macadamian standards still wide general and do not cover all the aspect of Java languages.

Infospheres [11] is quite old standards for Java, they were built in 1999. The Infospheres categorized their standards into five categories: Structure; which include identifying and calling standards for Packages, Program files, classes and interfaces, and methods. Second category is Documentation; which include the commenting and embedded HTML issues and uses standards. The third category is Naming which discuss the naming style for packages, classes, methods …etc., in term to be clear and correct. Tools and Examples are the fifth category. Tools that support Infospheres standards such as: Emacs, and some examples that show how these standards are applicable.

Infospheres standards focus on the main features and parts of the Java programming languages, such as: packages, classes as mentioned earlier. These standards also have some examples and tools which make them more applicable and more learnable especially for junior developers. Also, make them more able to be distributed in wider area of the java coding world.

On the other hands, the AmbySoft standards are more specified and detailed. The AmbySoft standards have guidelines for variables and parameters, classes, methods and packages. Moreover, they listed some advices for Java code reusability and testing [12]. These standards then published in a book called "The Elements of Java Style" [13]. This book provides a set of rules for Java practitioners to follow. While illustrating these rules with parallel examples of correct and incorrect usage, the book provides a collection of standards, conventions, and guidelines for writing solid Java code which will be easy to understand, maintain, and enhance.

The Computer Science Department in Rochester Institute of Technology (RIT) [14] had used Sun standards with some modifications, such as Classes, Methods, White Spaces and some more. However, there standards still very general and simple which may be an advantage for new Java developers or learner.

Wilson [15] has expanded the sun java standards targeting mainly Java developers, but also team managers, systems architect and technical writers. In his technical notes, he covered the coding style of import statements, declarations, and tabbing and indentation. Moreover, Wilson discussed the naming and documentation and commenting. He also mentioned some good practices for some cases in Java.

In Wilson's technical notes, as in Sun standards, he addressed and discuss the standards in too general aspects and only considered the basics of Java language. So, it is not targeting the professional Java community, but indeed it is quite useful for new developers or technical supports people.

In 2003, Yohanan [16] has made a contribution on Sun standards. He added and removed some subtitles to customize Sun standards to be more reliable with his projects and team.

The BSSC standards present rules and recommendations about the use of the language constructs of Java. These standards describe what is possible and not necessarily what is desirable or acceptable, especially for large software engineering projects intended for usual java programs or safety-critical systems [17].

The BSSC standards have 146 rules, these rules are mandatory for developers to follow and if any rule breaks then it may cause a harm for the program. However, the 77 recommendations are optional or good to use. Therefore, the recommendations break may not cause a harm but it may cause some warning and code misunderstanding.

Unlike other standards, BSSC standards did not categorized Java programming languages features in any way. The standards are only divided in two main sections: Rules and Recommendations. The rules cover all the features that mentioned in previous standards like; classes, naming, and declaration. Where recommendations cover writing format advices of the same features.

The best thing about BSSC standards are that they are 'direct to the point'. "All rules and recommendations have a short title and an explanation. Many rules and recommendations are also followed by a rationale section justifying the application of the rule. Rules and recommendations may also contain examples showing how to apply them, or illustrating the consequences of not applying them" [17]. However, these standards are targeting the Java developers that are working with European Space Agency. So, there standards may not be as clear as it should to any other Java developers.

Google has its own style for Java programming language [18]. In its style document, Google has included all the features of Java programming language. In addition to the standards and standards of Java. Therefore, Google Java style is more as full document that combine what the developers need to write a readable, maintainable, and well documented Java program.

Google Java style is built based on Sun Microsystems Java standards. Therefore, Google Java style has categorized the style as same as Sun Microsystems Standards; Source File basics and structure, Formatting, Naming, and Programming practices. However, the Google style only insert the 'hard-and-fast rules' and avoiding giving advice to reduce the number of guideline which make it easier to understand and applied.

Geotechnical has introduce a Java style based on number of sources, individual experience, local requirements/needs, as well as suggestions. This style has some more specific rules than any other standards or styles and formatted it in a way that make it easier to use and referred to. Also, the Geotechnical style is focuses mainly on programming style and does not contain any Java technical recommendations at all [19].

Geotechnical Java style covers the same main features as all other standards, styles, and standards mentioned in this paper. However, it includes example in some rules to make sure that reader is fully understand them.

The Geotechnical style is the most modern style for Java. But, still not covering all the Java issues and features.

Fred Long et al. [20] have collected 75 guidelines from expert guidelines, recommendations, and code examples. These guidelines aim to produce a reliable, robust, fast, maintainable, and secure code.

For each guideline, there are sub rules. The authors explain each rule in words with a noncompliant code example to show how the common mistake, compliant examples to explore all the suggested solutions, and a risk assessment show the severity, likelihood, cost, priority, and level for rule violation. Moreover, the tools that able to detect the rule violations. References and related guidelines are also addressed for each rule.

NASA has Jet Propulsion Laboratory which considered Java code standards, under the name JPL java coding [21]. JPL has categorized the java standards into three main categories: Critical, Important and Advisory. Critical; these rules must always be followed and violations of these rules must be corrected as soon as possible. Important; these rules should be followed and violations of these rules should be corrected where practical. Advisory; these rules represent good practice. Violations of these rules are allowed but not recommended.

The Java feature; Methods for example, may have rules with the three categories; critical, important and advisory. However, the JPL document arrange the rules depending on the categories not depending on the features. Therefore, if a developer want to study the rules for Java method, for example, he or she needs to look through the document and collect the rules information across the document.

New England Java Standards [22] has released a standards document with 60 rules that categories in three categories; Standards which means the rules that widely accepted and little tolerance to violation. Styles which address the recommendations and Conventions that explain the good practices of the code using the standards and the style.

Gosling et. al. [23] considered Java SE edition 8, they start with a description and notation used to present the lexical and syntactic grammar for programming languages in general. Then they start get in more details and more specific for Java. Subsequently, they explain the Java features specifications in more details.

Gosling et. al. addressed the language issues in detail that none of the previous standards did. However, they have created a big document (792 pages) to be used as a quick reference for Java standards. Moreover, the document has included so many information that may not be useful for developers that are looking for direct and clear list of standards to use. On the other hand, the document has an example for every rule, which may reduce the ambiguity of rules, and make them clearer.

3. TOOLS

There are number of tools that support checking and applying the Java standards, styles, and standards.

Java Pre-Processor (JPP) [24] is a parsing pre-processor for Java 1.X programming language. JPP configuration is corresponding to the Infospheres Java coding standards.

The main functionalities the JPP provides is the code beautification, that order the code depending on Infospheres rules. It also Evaluate the code by checking the standards of any piece of code, code complexity analysis, documentation analysis, and object-oriented design principles. It also provides Class and interfaces specifications and testing, and documentation generation.

The Java Formatter [25] is based on a Java parser engine that allows control the formatting details of code blocks, comments and documentations. Java Formatter is ideal for making messy and obtuse code clear. It allows source code to be presented in a consistent format of developer choosing, it provides many source code format style and operations options as well as batch or individual file processing.

Java Card Modelling Language (JCML) [26] is a java applet that run for Java Card programs, these program that is run on devices with severe storage and processing restrictions. In this applet, the researchers proposed a verification presents a runtime verification approach based on Design by Contract to improve the safety of Java Card applications.

In [27], a method had been described for representing Java security constraints using the Alloy modeling language.

Checkstyle [28] is a development tool to help programmers write Java code that adheres to a coding standard. It is ideal for projects that want to enforce a coding standard. Checkstyle is highly configurable and can be made to support almost any coding standard. As example configuration files are supplied supporting the Sun Code Standards, Google Java Style.

PMD [29] is a static Java source code analyzer. It uses rule-sets to define when a piece of source is erroneous. PMD includes a set of built-in rules and supports the ability to write custom rules. Typically, issues reported by PMD are not true errors, but rather inefficient code, i.e. the application could still function properly even if they were not corrected.

While PMD does not officially stand for anything, it has several unofficial names, the most appropriate probably being Programming Mistake Detector.

Sun standards still can be used especially for Java beginner developers or in teaching Java as in the standards of Computer Science department in Rochester Institute of Technology [14].

However, the Sun Standard is adapted by many other standards and styles as seen earlier and also there are some tools like Eclipse Formatter [30] and The Apache Jakarta Project (JMeter) [31].

4. COMPARATIVE STUDY

There are different types of standards for Java programming Language. However, they are all have the same objective: create a robust Java program. As seen in the previous sections, a comparative table between the java standards can be produced to facilitate for Java programmer to prefer which one is the best use and applied in their applications. In table 1, the pervious standards are compared using six specifications: Author(s) name, date issued, standards document number of pages, standards structure, number of standards rules' categories, number of rules and whether there is an example for each rule or not in the document.

The comparative categories are targeting mostly the new Java programmer. Therefore, it does not go deep into technical terms. Moreover, it is difficult, specially for large standards documents, to go through all rules and compare it with other standards rules in other documents.

As seen in table 1, there are some notes regarding the Java standards. Which can be listed as follows:

The Java standards are very generic and can be understood differently by different developers. Which can lead to the opposite direction of the standards objectives.

All the rules are written in text format, there are no equations, notations or any symbolic form. Which may cause different interpretation when applied.

In the standards documents that use text (not numeric or pointed structure), most of the rules are embedded in paragraphs. As a result, it unorganized and its challenging in some cases to extract the rule.

In the most of the documents, there are no weights for the rules. i.e. all rules have the same level of importance.

Most of the standards are made by individuals, not companies or standards associations.

Most of the tools are using SUN or Google standards. Still, some tools such as PMD is using rules that approved by developers in addition to SUN and Google standards. Which make it the tool that has plenty of rules to apply.

5. CONCLUSIONS

This paper listed the most common used Java standards, styles, and standards with tools that support them. This study shows that there are number of rules and guidelines for Java programming code. However, they are all covers the main features of Java in similar way. As a conclusion of this research paper, the best Java Standards can be used is that developed and run by SUN or Google, using PMD.

ACKNOLWLEDGMENT

This work is part of a funded project by Al-Zaytoonah University of Jordan, under the agreement number 11/10/2016.

Table 1: Java Standards List

Specifications *Standards*	*Author*	*Date issued*	*Document size (in pages)*	*Rules structure (numbered or text)*	*Number of Rule Categories*	*Number of Rules*	*Examples explain the rules*
Java Code convention	SUN microsystems	1996	20	Text	8	Around 81	Yes
Java coding standards	Philip Johnson	1996	2	Text	8	Around 16	Yes
CHiMu	ChiMu corporation	1997	54	Text	15	Around 105	No
Netscape	Netscape	1998	13	Text	6	Around 44	Yes
Coding convention for C++ and java	Macadamian	1998	10	Text	5	Around 76	No
Infospheres	CalTech	1999	24	Text	4	Around 91	No
Ambysoft	Scott W. Ambler	2000	76	Text	9	Around 76	Yes
Java coding standards	CS department in RIT	2000	4	Text	10	Around 54	No
Java coding style guide	Achut Reddy	2000	23	Text	7	Around 188	Yes
Java coding standards	Exolab Arnaud Blandin et.al.	2002	39	Pointed 90 rules 15 recommendations	4	105	Yes
Coding standards for java	New England Java Users group	2002	62	Numbered 9 standards, 29 style 22 conventions	3	60	Yes
BSSC	European Space Agency	2005	113	Numbered 146 rules 77 recommendations	11	223	No
Java coding standards	Department of Veterans Affairs	2009	72	Text	13	Around 64	Yes
Java standards	Bahraini e-government Sharmila Naveen	2010	31	Text + points	8	Around 135	Yes
The CERT® Oracle® Secure Coding Standard for Java	Fred Long, Dhruv Mohindra, Robert C. Seacord, Dean F. Sutherland and David Svoboda	2011	699	Text	17	N/A	Yes
JPL java coding	Jet Propulsion Laboratory (NASA)	2014	330	Points	3 (Critical, Important, Advisory)	218	Yes
The java language specifications SE 8ed	James Gosling, Bill Joy Guy Steele, Gilad Bracha and Alex Buckley	2014	644	Text	16	N/A	Yes
Java programming style guide lines	Geotechnical Software Service	2015	16	Numbered	5	86	Yes
Code convention for java	Steve Yohanan	2016	67	Points	9	Around 112	Yes
Google Java Style	Google	N/A	18	Numbered	7	27	Yes

REFERENCES

[1] I. O. f. Standardization. (2017, 21/1/2017). Standards. Available: http://www.iso.org/iso/home/standards.htm

[2] N. Software, "Netscape's Software coding standards guide for Java," ed: Source Format, 1998.

[3] A. S. o. M. Engineering. (2010, Codes and Standards at glence. The ASME Codes and Standards Writing Guide. Available: http://www.ieee.org/documents/astmarticles.pdf

[4] P. Verhas. (2015, 2016/5/7). Do we need coding conventions? Available: https://www.javacodegeeks.com/2015/11/do-we-need-coding-conventions.html

[5] S. Microsystems, "Java Code Conventions," ed. USA: Sun Microsystems, 1996.

[6] T. Nurkiewicz. (2012, 2016/05/07). Java Coding Conventions considered harmful. Available: https://www.javacodegeeks.com/2012/10/java-coding-conventions-considered-harmful.html

[7] P. Johnson. (1996, 17/2/2017). Java Coding Standard. Available: http://www.sourceformat.com/coding-standard-java-johnson.htm

[8] A. Reddy, Java™ Coding Style Guide, 1998.

[9] C. Corporation, "ChiMu OO and Java Development: Guidelines and Resource," ed, 1998.

[10] M. T. Inc., "Coding Conventions for C++ and Java applications," ed: Macadamian Technologies Inc, 1998.

[11] Caltech, "The Infospheres Java Coding Standard," ed, 1999.

[12] AmbySoft, "Writing Robust Java Code: The AmbySoft Inc. Coding Standards for Java ", ed: AmbySoft, 2000.

[13] A. Vermeulen, S. W. Ambler, G. Bumgardner, E. Metz, T. Misfeldt, J. Shur, et al., The Elements of Java Style: CAMBRIDGE UNIVERSITY PRESS, 2000.

[14] T. C. S. D. a. RIT, "Java Coding Standard," ed: The Computer Science Department at RIT, 2000.

[15] B. Wilson, "Java Coding Convention ", ed, 2000.

[16] S. Yohanan, "Code Conventions for Java," ed, 2003.

[17] E. S. A. B. f. S. S. a. Control, "Java Coding Standards," ed, 2005.

[18] Google, "Google Java Style," ed: Google, 2014.

[19] G. S. Services, "Java Programming Style Guidelines," ed, 2015.

[20] F. Long, D. Mohindra, R. C. Seacord, D. F. Sutherland, and D. Svoboda, Java™ Coding Guidelines: 75 Recommendations for Reliable and Secure Programs: Pearson Education, 2014.

[21] J. P. Laboratory, "JPL Java Coding Standard," ed: California Institute of Technology, 2014.

[22] N. E. J. U. Group, "NEJUG Standards," ed, 2002.

[23] J. Gosling, B. Joy, G. L. S. Jr., G. Bracha, and A. Buckley, The Java Language Specification, Java SE 8 Edition, 1 ed.: Addison-Wesley Professional, 2014.

[24] J. R. Kiniry and E. Cheong, "JPP: A Java Pre-Processor," California Institute of Technology1998.

[25] D. Software. (2014, 2016/05/07). Java Source Code Formatter. Available: http://www.sourceformat.com/java-formatter.htm

[26] U. S. d. Costa, A. M. Moreira, M. A. Musicante, and P. A. S. Neto, "JCML: A specification language for the runtime verification of Java Card programs," Science of Computer Programming, vol. 77, pp. 533-550, 2012.

[27] M. C. Reynolds, "Modeling the Java Bytecode Verifier," Science of Computer Programming, vol. 78, pp. 327-342, 3/1/ 2013.

[28] Checkstyle. (2016, 2016/05/07). CheckStyle. Available: http://checkstyle.sourceforge.net/

[29] SourceForge, "PMD," vol. 5.5.3, ed. https://github.com/pmd/pmd: SourceForge.com, 2017.

[30] Eclipse. (2016, 2016/05/06). Coding Conventions. Available: https://wiki.eclipse.org/Coding_Conventions

[31] J. P. team. (2016, 2016/05/06). Jakarta Project. Available: http://jakarta.apache.org/