SOFTWARE ENGINEERING STANDARDS & PRACTICES

Java Coding Standards

9AAD135383

Department	GF-IS ADM Applications Performance & Excellence (APE)
Approver	Giulio Bitella, Global Department Manager
Owner	Tomasz Jastrzębski, Global Leader for Software Engineering (SE)

For the latest distributable version of this and other Software Engineering standards please visit this link to ABB Library.

WHAT IS THIS?

This document presents set of coding standards, design principles and naming conventions that applies to Java language. For the general Best Coding Practices refer to document 9AAD135446.

TABLE OF CONTENTS

INTRODUCTION

This document addresses Java specific standards only and is continuation of the Best Coding Practices document (9AAD135446) which covers:

1. platforms
2. design principles
3. web applications
4. web/api services
5. mobile applications
6. cloud services
7. authentication and authorization
8. data access
9. unit tests
10. logging and tracing
11. use of open source software
12. group standards and policies

DEVELOPMENT

1. In general, the latest Oracle Java EE specification should be followed, e.g. http://www.oracle.com/technetwork/java/javaee/tech/index.html
2. Applications should be developed using the latest Java API. Use of Java and Java EE API versions other than the latest stable is not allowed.
3. During the development, the latest Java EE specification should be used as a reference.
4. Do not use and introduce libraries or frameworks if functionality can be implemented using Java/Java EE API. As a first resort, solution should depend on Java/Java EE API, implementation should be provided by application server runtime.
5. Oracle coding conventions are mandatory with the exception that allows 160 characters line length.
6. Usage of tools for code formatting is advised. Code formatter enabled in IDE on 'save' action is recommended practice.
7. Usage of tools for static analysis is advised. The analysis should be run regularly and its findings must be fixed and aligned with best code practices as early as possible to avoid further code quality degradation.

EFFECTIVE JAVA PRINCIPLES

1. For Java **version 7** and earlier, it is required to use coding principles (item 1-78) described in Bloch, J. (2008). *Effective Java*. 2nd ed. Boston: Addison-Wesley, pp.5-315.

2. For Java **version 8** and newer, it is required to use coding principles (item 1-90) described in Bloch, J. (2018). *Effective Java*. 3rd ed. Boston: Addison-Wesley, pp.23-405.

DATA ACCESS

Use JPA and JTA specifications for data access.

UNIT TESTS

1. Use test driven development. Suggested frameworks: JUnit, Mockito.
2. Tests should be as fast as possible.
3. Static methods should be avoided (e.g. do not use PowerMock framework to mock static methods).

LOGGING AND TRACING

1. The most recommended logging frameworks are log4j and slf4j.
2. Consider using Azure App Insights for Microsoft Azure cloud hosted applications.

SECURITY

Oracle Secure Coding Guidelines should be followed:
http://www.oracle.com/technetwork/java/seccodeguide-139067.html

REFERENCES

1. IS Standard for Application Development Frameworks - 9AAD135404
2. IS Standard for Web Java Script Frameworks - 9AAD133991
3. Best Coding Practices - 9AAD135446
4. Test Strategy - 9AAD134969
5. Unit Testing Field Guide - 9AAD135249
6. Source Code Management Standard - 9AAD134843
7. User Experience & Design Standards and Practices - 9AAD134800

RECOMMENDED READING

1. Martin, R. C. (2016). *Clean code: A handbook of agile software craftsmanship.* Upper Saddle River, NJ: Prentice Hall.
2. Bloch, J. (2018). *Effective Java*. 3rd ed. Boston: Addison-Wesley

REVISION HISTORY

Rev.	Page	Change Description	Author(s)	Date
A	all	approved	Rafał Curyło, Krzysztof Kabała, Tomasz Jastrzębski et al.	2019-04-09

Dev

Information Systems

Applications Performance Excellence Department (APE)

Software Engineering Standards & Practices