Politecnico di Milano

A.A 2016-2017

Code Inspection Document

Version 1.0

PowerEnjoy

Instructor : Prof. Di Nitto

Authors:
Amico Simone
Chianella Claudia Beatrice
Giovanakis Yannick

CONTENTS

1. INTRODUCTION

The code inspection is a systematic examination of computer source. It is intended to find mistakes overlooked during the initial development phase, with the aim of developing the overall quality of the software.

This document indexes all coding mistakes found following a standard code inspection check-list for the requested classes listed in the underling section 1.1

1.1 Class overview

The Apache OFBiz Project is an open source product for the automation of enterprise processes that includes framework components and business applications for ERP,CRM and other business-oriented functionalities.[1]

Among the many, the following classes were assigned for code inspection:

- **ProductStoreCartAwareEvents.java**

    - *Location*: apache-ofbiz-16.11.01/applications/order/src/main /java/org/apache/ofbiz/order/shoppingcart/product /ProductStoreCartAwareEvents.java

    - *Class role*: see section 2.1

    - *Total number of issues found*: TBD

- **ProductDisplayWorker.java**

    - *Location*: apache-ofbiz-16.11.01/applications/order/src/main /java/org/apache/ofbiz/order/shoppingcart/product /ProductDisplayWorker.java

    - *Class role*: see section 2.2

    - *Total number of issues found*: TBD

---

[1]Full source code: `http://mirror.nohup.it/apache/ofbiz/apache-ofbiz-16.11. 01.zip`

2. FUNCTIONAL ROLE OF CLASSES

This section analyses the meaning and functionality for each of the inspected classes.

2.1 ProductStoreCartAwareEvents.java

This java class is composed of two static methods. The first static method takes a `HttpServletRequest` and `HttpServletResponse` as input. The method extracts the `productStoreId` String parameter from the request and tries to invoke the second static method. Depending on the success or failure of the second method , the the first method return either the string "success" or "error"

The second static method takes extracted `productStoreID` String and the `HttpServletRequest` as input. The method than :

- returns immediately if the `productStoreID` is `null` or if it equals the `HttpSession`'s old `productStoreID`.

- throws an exception if `productStoreID` is not related to a valid productStore , if the `webSite` cannot be found in relation to the request or if `webSite` exists but does not allow changing of the `productStoreID`.

- if the points above are not matched , sets the `productStoreID` of the `session` to the one passed as parameter ,clears the old product catalogue and eventually sets the session's `currency`, `locale` and `time-zone` depending on the `productStore`. Finally the method creates a new `shoppingCart` for the session without erasing the old one.

2.2 ProductDisplayWorker.java

3. LIST OF ISSUES

3.1 ProductStoreCartAwareEvents.java

3.1.1 Naming conventions

- All class names, interface names, method names, class variables, method variables, and constants used have meaningful names and do what the name suggests. (setCurrencyUomIfNone che cazzo vor d?)

- The only one-character variable (Exception e) is a temporary throw-away variable.

- Class names are nouns, in mixed case, with the first letter of each word in capitalized.

- There are no interfaces.

- Method names are verbs, with the first letter of each addition word capitalized.

- All attributes, are mixed case, no one begin with an underscore. All the remaining words in the variable name have their first letter capitalized.

- There are no constant.

3.1.2 Indention

- Four spaces are used for indentation and it is done consistently.

- No tabs are used to indent.

3.1.3 Braces

- In all the class is used the Kernighan and Ritchie style (first brace is on the same line of the instruction that opens the new block).

- All if, while, do-while, try-catch, and for statements that have only one statement to execute are surrounded by curly braces.

3.1.4 File Organisation

- Either blank lines or comments are used to separate sections.

- Line length exceeds 80 characters many times. Six times it exceeds also 120 characters (3 of them are comments).

3.1.5 Wrapping Lines

- Line break occurs only after a comma or an operator.

- ???????????? Higher-level breaks are used ?????????????????????

- A new statement is aligned with the beginning of the expression at the same level as the previous line.

3.1.6 Comments

- Comments are used to adequately explain what the class, interface, methods, and blocks of code are doing.

- ?? Commented out code contains a reason for being commented out and a date it can be removed from the source file if determined it is no longer needed.??????????????????????????????

3.2   Java Source Files

3.3   Package and Import Statements

3.4   Class and Interface Declarations

3.5   Initialisation and Declaration

3.6   Method Calls

3.7   Arrays

3.8   Object Comparison

3.9   Output Format

3.10   Computation , Comparisons and Assignments

3.11   Exceptions

3.12   Flow Control

3.13   Files

3.14 ProductDisplayWorker.java

3.14.1 Naming conventions

- All class names, interface names, method names, class variables, method variables, and constants used have meaningful names and do what the name suggests. (nqdbl, nqint, curpcms, curpcm che cazzo vor d?)

- The only one-character variable (Exception e) is a temporary throw-away variable.

- Class names are nouns, in mixed case, with the first letter of each word in capitalized.

- There are no interfaces.

- Method names are verbs, with the first letter of each addition word capitalized.

- All attributes, are mixed case, no one begin with an underscore. All the remaining words in the variable name have their first letter capitalized, except one (cartiter, which should have been cartIter).

- There are no constant.

3.14.2 Indention

- Four spaces are used for indentation and it is done consistently.

- No tabs are used to indent.

3.14.3 Braces

- In all the class is used the Kernighan and Ritchie style (first brace is on the same line of the instruction that opens the new block).

- All while, do-while, try-catch, and for statements that have only one statement to execute are surrounded by curly braces. There are 10 if statements which contain onlu one statement and are written in one line without braces.

3.14.4 File Organisation

- Either blank lines or comments are used to separate sections.

- Line length exceeds 80 characters many times. 10 times it exceeds also 120 characters One of them it's over 200.

3.14.5 Wrapping Lines

- Line break occurs only after a comma or an operator.

- ???????????? Higher-level breaks are used ????????????????????

- A new statement is aligned with the beginning of the expression at the same level as the previous line.

3.14.6 Comments

- Comments are used to adequately explain what the class, interface, methods, and blocks of code are doing.

- ?? Commented out code contains a reason for being commented out and a date it can be removed from the source file if determined it is no longer needed.?????????????????????????????????

4. OTHER ISSUES

Section dedicated to other issues found

5. APPENDICES

5.1 References

The following tools where used in the creation of this document:

- *TexMaker 4.5* as Editor

5.2 Effort Spent

- Simone Amico   12h
- Chianella Claudia Beatrice   12 h
- Giovanakis Yannick   12 h