POLITECNICO DI MILANO

COMPUTER SCIENCE AND ENGINEERING

MASTER OF SCIENCE

Code Inspection

PowerEnjoy

Software engineering II project

Giovanni Agugini, Matteo Foglio, Tommaso Massari

Contents

1 Introduction

1.1 Revision History

Table 1: Revision History

Version	Date	Author(s)	Summary
1.0	28/01/2017	Giovanni Agugini, Matteo Foglio e Tommaso Massari	Initial release

1.2 Purpose and Scope

The purpose of this document is to provide a result of a code inspection on the source code of an Apache project. This kind of inspection, born in the 70's thanks to IBM engineer Michael Fagan and then developed by the software community, insures more quality and certainty on the software and its components. Moreover, the company who committed the software is really interested in inspecting the source code as many statistics say that it can let the company save a huge percentage of maintenance and rehabilitation costs. The purpose of the document is therefore underlining occurred errors, noticing bad ways of coding and giving advice on the code improvement. The conducted analysis concerns only the source code and the language programming issues, thus the document wants to be a guideline for developers to correct its proper code. This inspection is normally part of the natural lifecycle of a software and in fact the errors made by developers are not intended to be objects for a performance evaluation by the management.

2 Class Assignment

The class assigned to our group is ShoppingListServices. It can be found at "../apache-ofbiz-16.11.01/applications/order/src/main/java/org/apache/ofbiz/order/shoppinglist/ShoppingListServices.java "

3 Functional Role

The class provides methods used to manage shopping list. The lifecycle of a shopping list seems to be covered from its creation to its rom its creation to its elimination. There is also the possibility of autosaving the shopping list.

Moreover the class covers the relationship between shopping list and cart, providing some methods that allow to convert one to the other.

There are different methods:

- autoDeleteAutoSaveShoppingList
- createListReorders

- makeListFromOrder
- makeShoppingListCart
- setShoppingListRecurrence
- splitShipmentMethodString
- updateShoppingListQuantitiesFromOrder

In addition we can hilight the overlap of the method makeShoppingListCart showing that a shopping cart can be created in different ways.

4 Issues found

1

"All class names, interface names, method names, class variables, method variables, and constants used should have meaningful names and do what the name suggests."

Issues found on lines:

- 70, 120, 245, 266 544, 575 a parameter of type DispatchContext is used with different names "dctx" "ctx"
- 131 for variable name "eli" of type EntityListIterator
- 245 for variable name "dctx", not meaningful
- 280 for variable name "serviceCtx" not completely meaningful
- 315 for variable name "orh", not meaningful
- 327 for variable name "ctx", not meaningful
- 332 for variable name "ConfigId", not completely meaningful
- 357 for variable name "slCtx", not meaningful
- 372 for variable name "slUpResp", not meaningful
- 431 for vairable name "currencyUom" of type String
- 477 for varible name "reserv

7

"Constants are declared using all uppercase with words separated by an underscore. Examples: MIN WIDTH; MAX HEIGHT."

Issues found on lines:

- 67, 68

8

"Three or four spaces are used for indentation and done so consistently."

Issues found on lines:

- 613, 614: used 8 spaces instead of 4
- 458, 459, 460: used 8 spaces instead of 4

12

“Blank lines and optional comments are used to separate sections (be- ginning comments, package/import statements, class/interface decla- rations which include class variable/attributes declarations, construc- tors, and methods)”

Issues found on lines:

- 290: useless blank line

13

The number of characters of the following lines exceed 80 characters but not 120.

- 18: 81 characters
- 86: 111 characters
- 89: 112 characters
- 92: 111 characters
- 113: 96 characters
- 120: 119 characters
- 137: 87 characters
- 139: 89 characters
- 141: 91 characters
- 155: 104 characters
- 164: 99 characters
- 165: 96 characters
- 168: 83 characters
- 170: 106 characters
- 183: 119 characters
- 186: 89 characters
- 191: 117 characters
- 208: 108 characters
- 210: 107 characters
- 218: 108 characters

- 227: 83 characters
- 236: 87 characters
- 237: 82 characters
- 244: 103 characters
- 245: 118 characters
- 267: 110 characters
- 281: 109 characters
- 290: 89 characters
- 292: 84 characters
- 298: 95 characters
- 303: 82 characters
- 328: 95 characters
- 332: 84 characters
- 334: 117 characters
- 335: 87 characters
- 343: 90 characters
- 347: 86 characters
- 354: 92 characters
- 355: 91 characters
- 358: 89 characters
- 359: 83 characters
- 362: 81 characters
- 391: 103 characters
- 393: 107 characters
- 401: 108 characters
- 404: 120 characters
- 429: 81 characters
- 433: 106 characters

- 452: 96 characters
- 454: 98 characters
- 456: 91 characters
- 457: 109 characters
- 458: 85 characters
- 459: 86 characters
- 469: 85 characters
- 470: 89 characters
- 475: 86 characters
- 479: 88 characters
- 481: 81 characters
- 487: 85 characters
- 488: 92 characters
- 494: 107 characters
- 496: 90 characters
- 502: 87 characters
- 503: 87 characters
- 505: 85 characters
- 506: 102 characters
- 508: 92 characters
- 509: 108 characters
- 511: 86 characters
- 512: 96 characters
- 514: 96 characters
- 515: 111 characters
- 523: 119 characters
- 536: 106 characters
- 537: 105 characters

- 538: 112 characters
- 552: 92 characters
- 556: 107 characters
- 557: 87 characters
- 559: 108 characters
- 561: 85 characters
- 568: 99 characters
- 570: 97 characters
- 585: 112 characters
- 606: 97 characters
- 613: 100 characters

14

The number of characters of the following lines exceed 120:

- 70: 126 characters
- 79: 130 characters
- 95: 150 characters
- 107: 124 characters
- 110: 148 characters
- 132: 173 characters
- 153: 136 characters
- 177: 147 characters
- 213: 186 characters
- 221: 129 characters
- 226: 126 characters
- 270: 160 characters
- 280: 123 characters
- 293: 138 characters
- 308: 126 characters

- 311: 128 characters
- 320: 168 characters
- 327: 147 characters
- 329: 182 characters
- 331: 143 characters
- 348: 191 characters
- 380: 195 characters
- 396: 177 characters
- 415: 123 characters
- 427: 146 characters
- 445: 123 characters
- 484: 265 characters
- 489: 147 characters
- 492: 217 characters
- 527: 126 characters
- 544: 137 characters
- 549: 127 characters
- 554: 203 characters
- 575: 130 characters
- 581: 149 characters
- 612: 137 characters
- 620: 155 characters

15

"Line break occurs after a comma or an operator."
Issues found on lines:

- 458, 459, 460: line breaks before the operator "+" instead of after it

18

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

Issue found: generally speaking the comments are not complete. Some methods are poorly commented and others present no comments at all.

19

"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."

Issues found: as stated before, there are incomplete comments and moreover there's no date on a comment explaining why and when it could be removed.

23

"Check that the javadoc is complete (i.e., it covers all classes and files part of the set of classes assigned to you)."

Issue found: Javadoc is not complete. It doesn't cover all methods and it doesn't explain the role of the class.

In particular no javadoc has been found for the methods:

- setShoppingListRecurrence
- createListReorders
- splitShipmentMethodString
- makeListFromOrder
- autoDeleteAutoSaveShoppingList

27

"Check that the code is free of duplicates, long methods, big classes, breaking encapsulation, as well as if coupling and cohesion are ade- quate."

Issues found: potential issue in the method "MakeListFromOrder" that's 162 lines long.

33

"Declarations appear at the beginning of blocks (A block is any code surrounded by curly braces '{' and '}'). The exception is a variable can be declared in a for loop. "

Issue: variables are not declared at the beginning of the block.

Issues found at lines:

- 99, 100

- 105
- 114
- 164, 165
- 168
- 213
- 266
- 323 - 338, there are a lot of declarations nested into if cycles
- 443
- 466
- 477
- 487, 488, 489
- 585
- 598
- 601, 602

42

"Check that error messages are comprehensive and provide guidance as to how to correct the problem."

Issues found on lines:

- 109, "Debug.logError(e, module)" is not a precise way to describe an occured exception
- 147, same thing as above
- 179, same thing as above
- 193, same thing as above
- 319, same thing as above
- 338, same thing as above
- 345, same thing as above
- 376, same thing as above
- 397, same thing as above
- 447, same thing as above

- 529, same thing as above
- 583, same thing as above
- 608, same thing as above
- 616, same thing as above
- 622, same thing as above

50

"Check throw-catch expressions, and check that the error condition is actually legitimate."

Issues found on lines:

- 330 - 339, the caught exception is not clearly defined and potentially is not correclty handled

53

"Check that the appropriate action are taken for each catch block."

Issues found on lines:

- 347, why not doing that inside the catch block?
- 373, same thing as above

5 Other Problems

- 131, 132: a variable is declared on line 131 and initialized to null . On line 132 the same variable is initialized again with a different value.
- 482: blank line is used at the beginning of an if block. This is not consistent with the general code style of the whole class.
- 483-485: this block should be more indentedò
- Often variables are initialised as null and then they are assigned to a specific value. This initialisation to null is redundant and it has to be removed.

6 Effort Spent

Giovanni Agugini:

- 28/01/2017 : 3h
- 01/02/2017 : 3h
- 03/02/2017: 2h
- 05/02/2017: 2h

Matteo Foglio:

- 28/01/2017 : 3h
- 01/02/2017 : 3h
- 03/02/2017: 2h
- 05/02/2017: 2h

Tommaso Massari:

- 28/01/2017 : 3h
- 01/02/2017 : 3h
- 03/02/2017: 2h
- 05/02/2017: 2h