School of Industrial and Information Engineering

Department of Computer Science and Engineering

Software Engineering II

PowerEnJoy – Code Inspection Document

Presented by:

Bachar Senno

Gianluigi Iobizzi

Onell Saleeby

Under the supervision of:

Prof. Luca Mottola

Prof. Elisabetta Di Nitto

V1.0, 05-02-2017

Contents

Assigned Classes

- OrderManagerEvents.java
    - Path:../apache-ofbiz-16.11.01/applications/order/src/main/java/org/apache/ofbiz/order/OrderManagerEvents.java
    - Reviewed by Gianluigi Iobizzi and Onell Saleeby
- KeywordSearchUtil.java
    - Path:../apache-ofbiz-16.11.01/framework/common/src/main/java/org/apache/ofbiz/common/KeywordSearchUtil.java
    - Reviewed by Bachar Senno

Functional Role of Assigned Classes

OrderManagerEvents.java

The class offers several static tools useful to manage the payment status of commercial orders.

KeywordSearchUtil.java

This class contains methods pertaining to keyword searching. Just by examining the name of the class and the naming of the contained methods (i.e. processForKeywords, fixupKeywordSet, expandKeywordForSearch etc.) this can be easily deduced. This is also mentioned in the documentation available here: https://ci.apache.org/projects/ofbiz/site/javadocs/index.html?help-doc.html

Technical Issues

OrderManagerEvents.java

Into class "OrderManagerEvents"

- Lines **52-54** (before "OrderManagerEvents class statement"): The class documentation comment (Javadoc) doesn't contain any detail about the role and the goal of it (**issues 23 / 25a**);

Into class "OrderManagerEvents" – method "processOfflinePayments"

- Line **60** ("//FIXME : … ") : A temporary comment before "processOfflinePayments" method's statement, inserted to signal the probably uselessness of it and suggesting its deletion, has been left into the code; (**issue 19**)
- Line **71** (inner variable "paymentPrefs" definition): the variable's names is not enough quite-explanatory and raises doubts about the suffix's meaning (maybe preferences?) (**issue 1**);
- Line **73** (before *try … catch* statement): no blank lines have been left between the control block and the previous declarations (**issue 12**);
- Lines **74 -75** (assignments to "paymentPrefs" and "placingCustomer" variables): the right part of the assignments has got too many nested calls to be resolved that might reduce code's clearness, moreover each line's length exceeds the suggested limit (80 ASCII characters per line) (**issues 44/ 13**);
- Line **76** (*catch* statement): the bracing conventions are not followed cause the command shouldn't be placed within the same line of a block's final bracket but in a new one (**issue 10**);
- Line **81** (*if* statement): no blank lines have been left between the control block and the previous block (**issue 12**);
- Line **82** (enhanced for-loop over List "paymentPrefs"): the name of variable "ppref" is not enough quite-explanatory and raises doubts about the its meaning (maybe preference?) (**issue 1**);
- Line **84** ("TODO: …"): A Temporary comment has been left to suggest the opportunity of moving the service into a dedicated method (**issue 19**);
- Line **91** (*try … catch* statement): no blank lines have been left between the control block and the previous declarations (**issue 12**);

- Lines **92-93** (assignment to "results" variable): the right part of the assignment has got too many nested calls to be resolved that might reduce code's clearness (**issues 44**);
- Line **94** (*catch* statement): the bracing conventions are not followed cause the command shouldn't be placed within the same line of a block's final bracket but in a new one (**issue 10**);
- Line **110** (*catch* statement): the bracing conventions are not followed cause the command shouldn't be placed within the same line of a block's final bracket but in a new one (**issue 10**);

Into class "OrderManagerEvents" – method "receiveOfflinePayment"

- Line **135** (*try … catch* statement): no blank lines have been left between the control block and the previous declarations (**issue 12**);
- Line **136** (assignment to "orderHeader" variable): the right part of the assignment has got too many nested calls to be resolved that might reduce code's clearness (**issue 44**);
- Line **137** (*catch* statement): the bracing conventions are not followed cause the command shouldn't be placed within the same line of a block's final bracket but in a new one (**issue 10**);
- Line **143** (declaration of "grandTotal" variable): The declaration does not appear at the begging of a block (**issue 33**);
- Line **144** (*if* statement): no blank lines have been left between the control block and the previous block (**issue 12**);
- Line **149** (declaration of "paymentMethodTypes" variable): The declaration does not appear at the beginning of a block (**issue 33**);
- Line **152** (assignment to "paymentMethodTypes" variable): the right part of the assignment has got too many nested calls to be resolved that might reduce code's clearness, moreover the line's length exceeds the suggested limit (80 ASCII characters per line) (**issues 44/ 13**);
- Line **153** (*catch* statement): the bracing conventions are not followed cause the command shouldn't be placed within the same line of a block's final bracket but in a new one (**issue 10**);
- Line **165** (declaration of "paymentMethods" variable): The declaration does not appear at the start of a block (**issue 33**);

- Line **166** (*if* statement): no blank lines have been left between the control block and the previous block (**issue 12**);
- Line **167** (assignment to “paymentMethods” variable): the right part of the assignment has got too many nested calls to be resolved that might reduce code’s clearness (**issue 44**);
- Line **168** (*catch* statement): the bracing conventions are not followed cause the command shouldn’t be placed within the same line of a block’s final bracket but in a new one (**issue 10**);
- Line **182** check if paymentMethod is null before iterating
- Line **212** possible NullPoint when calling results.get
- Line **270** possible NullPoint when calling results.get
- Line **237**(“TODO: …”): A Temporary comment has been left to suggest the opportunity of moving the service into a dedicated method (**issue 19**);
- Line **223** (declaration of “paymentMethodTypes” variable): The declaration does not appear at the begging of a block (**issue 33**);
- Line **317**(“TODO: …”): A Temporary comment has been left to suggest the opportunity of moving the service into a dedicated method (**issue 19**);
- -Line **270** possible NullPoint when calling results.get
- Line **307** (declaration of “grandTotal” variable): The declaration does not appear at the begging of a block (**issue 33**);
- Line **195** (*if* statement): no blank lines have been left between the control block and the previous block (**issue 12**);
- Line **227** (*if* statement): no blank lines have been left between the control block and the previous block (**issue 12**);
- Line **235** (*if* statement): no blank lines have been left between the control block and the previous block (**issue 12**);
- Line **291** (*if* statement): no blank lines have been left between the control block and the previous block (**issue 12**);

- Line **307** (*if* statement): no blank lines have been left between the control block and the previous block (**issue 12**);
- Line **310** (*if* statement): no blank lines have been left between the control block and the previous block (**issue 12**);
- Line **318** (*try … catch* statement): no blank lines have been left between the control block and the previous declarations (**issue 12**);
- Line **175** (*try … catch* statement): no blank lines have been left between the control block and the previous declarations (**issue 12**);
- Line **189**(*try … catch* statement): no blank lines have been left between the control block and the previous declarations (**issue 12**);
- Line **198** (*try … catch* statement): no blank lines have been left between the control block and the previous declarations (**issue 12**);
- Line **229** (*try … catch* statement): no blank lines have been left between the control block and the previous declarations (**issue 12**);
- Line **259** (*try … catch* statement): no blank lines have been left between the control block and the previous declarations (**issue 12**);
- Line **282** (*try … catch* statement): no blank lines have been left between the control block and the previous declarations (**issue 12**);
- Line **191** (*catch* statement): the bracing conventions are not followed cause the command shouldn't be placed within the same line of a block's final bracket but in a new one (**issue 10**);

Overall considerations

- All over the class, the comments aren't used enough to explain the code and where used they are too "skinny" (**issue 18**);
- **No logical bugs detected**

KeywordSearchUtil.java

At the start of the class, the package to which the class belongs is specified. Then, the needed libraries are imported, and one public class pertaining to the java file is defined. This is all done correctly according to the required general structure of the document. Before the declaration of the public class, a brief commented line describes the function of the class.

The class itself contains at the start the definition of the needed variables. During the declaration of these variables, the following code takes place:

```
private static Set<String> thesaurusRelsToInclude = new HashSet<String>();
private static Set<String> thesaurusRelsForReplace = new HashSet<String>();

static {
    thesaurusRelsToInclude.add("KWTR_UF");
    thesaurusRelsToInclude.add("KWTR_USE");
    thesaurusRelsToInclude.add("KWTR_CS");
    thesaurusRelsToInclude.add("KWTR_NT");
    thesaurusRelsToInclude.add("KWTR_BT");
    thesaurusRelsToInclude.add("KWTR_RT");

    thesaurusRelsForReplace.add("KWTR_USE");
    thesaurusRelsForReplace.add("KWTR_CS");
}
```

While this isn't outright incorrect, this chunk of code could've been replaced by a for-loop that does the same job but that is neater, as follows:

```
private String[] relsToAdd = {"KWTR_UF", "KWTR_USE", "KWTR_CS", "KWTR_NT", "KWTR_BT", "KWTR_RT"};
private String[] relsForReplace = {"KWTR_USE", "KWTR_CS"};

for (String rel : relsToAdd) {
    static {
        thesaurusRelsToInclude(rel);
    }
}

for (String rel : relsForReplace) {
    static {
        thesaurusRelsForReplace(rel);
    }
}
```

The advantage with code relies in the fact that in case there comes a time where more elements need to be added to the list, this can be done simply by adding the needed strings in the respective arrays without having to add new statements for each of the added strings.

Next, 12 public methods are defined and implemented. These methods are the main purpose of this class.

When making value comparisons, the .equals method is called (and not the "==" comparator) which matches the required criteria related to this topic. An example of this would be the `return "true".equals(removeStemsStr);` on line 75 which compares the value of `removeStemsStr` to the string "true" and returns the Boolean value of the comparison.

On line 112, at the start of the method, there's a test to make sure that the keywordSet isn't null. This is fine, however, if the keywordSet is indeed null, the function returns without any kind of message. It might be better to just add a debug message so that when the code is running it's easier to follow the flow. Something like `Debug.logInfo("keywordSet is null in fixupKeywordSet");` does the job.

The code is overall clean, indentations are well placed (4 spaces, no tabs), except for one indentation mistake at the end of the `makeKeywordSet` method, where the closing brace should be moved one indentation forward. The parts that need explanation are commented, although it might help to also add some comments at the start of each method for documentation purposes, just briefly describing the purpose, inputs, and outputs of the method.

The number of characters on some of the lines is way too big, reaching a maximum of 262 characters (when defining the `processForKeywords` method) on the same line. This could've been avoided by adding line breaks on the comma separators.