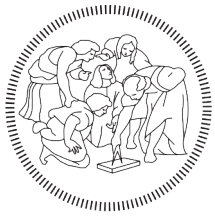

POLITECNICO
MILANO 1863

Code Inspection Document

Version: 1.0

Reda Aissaoui, Jinling Xing, Lidong Zhang

February 4, 2017

Content

1. Introduction

2. Assigned class

3. Functional role of class

4. Code inspection checklist

1. Naming Conventions
2. Indention
3. Braces
4. File Organization
5. Wrapping Lines
6. Comments
7. Java Source Files
8. Package and Import Statements
9. Class and Interface Declarations
10. Initialization and Declarations
11. Method Calls
12. Arrays
13. Object Comparison
14. Output Format
15. Computation, Comparisons and Assignments
16. Exceptions
17. Flow of Control
18. Files

5. Effort spent

1. Introduction

Code inspection is the most formal review type. It is led by the trained moderators. During inspection the documents are prepared and checked thoroughly by the reviewers before the meeting. It involves peers to examine the product. A separate preparation is carried out during which the product is examined and the defects are found.

The main goal of ode inspection is to have an efficient and readable code. This improves the maintainability of the code. It also increases the code-reuse.

2. Assigned Class

Requires taxcommon.class

TaxWare software

Why are they using BigDecimal. Currency calculations require precision to a specific degree, such as two digits after the decimal for most currencies. They also require a specific type of rounding behavior, such as always rounding up in the case of taxes.

Our group assigned class is '../apache-ofbiz-16.11.01/applications/order/src/main/java/org/apache/ofbiz/order
And our task is to perform the inspection and report on the quality status of selected code extracts using the checklist for Java code inspection reported. Also we are asked to deliver a document having the structure described in the code inspection assignments document.

3. Functional role of the Class

The class assigned to us is TaxWareUTL. The class, that is part of the following package **org.apache.ofbiz.order.thirdparty.taxware** . The class is part of an integration third party software, called TaxWare. This software is a solution for calculating the taxes.

The main goal of this integration is to be able to write TaxWare libraries. After exploring the other java files present with this one, we found class TaxwareServices that make use of TaxwareUTL. This class instantiates TaxwareUTL and sets a shipping address, a shipping amount and items. The process() function is then called in order to generate the file.

TaxWare software

Why are they using BigDecimal. Currency calculations require precision to a specific degree, such as two digits after the decimal for most currencies. They also require a specific type of rounding behavior, such as always rounding up in the case of taxes.

4. Code Inspection checklist

4.1 Naming Conventions

Row	Issue
77, 118, 119, 243, 244, 247, 248, 463	Missing braces for single If statment

Row	Issue
80, 85, 93, 101, 111, 113, 136, 138, 140, 146, 220, 277, 280, 301, 312, 323, 345, 356, 367, 453, 462	The blank line is useless according to the describle

Row	Code	Issue
84	Record rec = (Record) i.next();	The name of variable is not meaningful.
123	fos = new FileOutput-Stream(outFile);	The name of variable is not meaningful.
219	DataFile df = null;	The name of variable is not meaningful.
272,273	String headStr = retBuffer.toString().substring(0, 283);, String itemStr = retBuffer.toString().substring(284);	The name of variable is not meaningful.
288	Record rec = (Record) i.next();	The name of variable is not meaningful.
259	ModelField mf = (ModelField) model.fields.get(a);	The name of variable is not meaningful.

4.2 Indention

No errors found.

4.3 Braces

4.4 File Organization

4.5 Wrapping Lines

No errors found. ####4.6 Comments

There are few comments that are not meaningful for example line 74 '// make sure we have everything before processing'.

4.7 Java Source Files

4.8 Package and Import Statements

4.9 Class and Interface Declarations

4.10 Initialization and Declarations

4.11 Method Calls

4.12 Arrays

4.13 Object Comparison

```
263 if (shipToAddress == null)
264     throw new TaxwareException("Shipping address has not been set.");
```

4.14 Output Format

4.15 Computation, Comparisons and Assignments

4.16 Exceptions

```
124 try {
125     fos = new FileOutputStream(outFile);
126 } catch (FileNotFoundException e) {
127     e.printStackTrace();
128 }
```

```
131 try {
132     fos.close();
133 } catch (IOException e) {
134     e.printStackTrace();
135 }
```

The two exceptions quoted before do not take any action on the fact that there is a file I/O error. only the stack trace is printed.

4.17 Flow of Control

4.18 Files

5. Effort spent