INSPECTION DOCUMENT

SOFTWARE ENGINEERING 2 PROJECT

Reference Professor: Elisabetta Di Nitto
POLITECNICO DI MILANO
Computer Science and Engineering
Project of Software Engineering 2

Andrea Binelli
858235

Sommario

1. Classes/methods assigned 4
1.1 Classes assigned 4
1.2 Methods assigned 4
1.2.1 EjbConversionHelper 4
1.2.2 getRealRelationRoleDescriptor 4
1.3 Functional role of assigned set of classes/methods: 4
2. Inspection 5
2.1 EjbConversionHelper 5
2.1.1 Naming Conventions 5
2.1.2 Indentions 5
2.1.3 Braces 5
2.1.4 File Organization 5
2.1.5 Wrapping Lines 5
2.1.6 Comments 6
2.1.7 Java Source Files 6
2.1.8 Package and Import Statements 6
2.1.9 Class and Interface Declarations 6
2.1.10 Initialization and Declarations 6
2.1.11 Method Calls 6
2.1.12 Arrays 6
2.1.13 Object Comparison 6
2.1.14 Output Format 6
2.1.15 Computation, Comparisons and Assignments 7
2.1.16 Exceptions 7
2.1.17 Flow of Control 7
2.1.18 Files 7
2.2 getRealRelationRoleDescriptor 7
2.2.1 Naming Conventions 7
2.2.2 Identations 7
2.2.3 Braces 7
2.2.4 File Organization 7
2.2.5 Wrapping Lines 7
2.2.6 Comments 7
2.2.7 Java Source Files 7
2.2.8 Package and Import Statements 7

2.2.9 Class and Interface Declarations 7
2.2.10 Initialization and Declarations 7
2.2.11 Method Calls 7
2.2.12 Arrays 7
2.2.13 Object Comparison 8
2.2.14 Output Format 8
2.2.15 Computation, Comparisons and Assignments 8
2.2.16 Exceptions 8
2.2.17 Flow of Control 8
2.2.18 Files 8
2.3 Other problems 8
3. References 8

1. Classes/methods assigned

1.1 Classes assigned

Project: Glassfish 4.1 application server

Name: EjbConversionHelper

Start Line: 71

Location:

appserver/persistence/cmp/support-ejb/src/main/java/com/sun/jdo/spi/persistence/support/ejb/ejbc/EjbConversionHelper.java

1.2 Methods assigned

1.2.1 EjbConversionHelper

Name: EjbConversionHelper(NameMapper nameMapper)

Start Line: 83

Location:

appserver/persistence/cmp/support-ejb/src/main/java/com/sun/jdo/spi/persistence/support/ejb/ejbc/EjbConversionHelper.java

1.2.2 getRealRelationRoleDescriptor

Name: getRealRelationRoleDescriptor(String ejbName , String cmrFieldName , boolean self)

Start Line: 390

Location:

appserver/persistence/cmp/support-ejb/src/main/java/com/sun/jdo/spi/persistence/support/ejb/ejbc/EjbConversionHelper.java

1.3 Functional role of assigned set of classes/methods:

According to the comments the class implements ConversionHelper interface by using data from IASEjbBundleDescriptor. According to the Javadoc ConversionHelper is: "[…] an interface for the MappingFile. The mapping file classes use an object that implements this interface to assist in the conversion from the sun-cmp-mapping file, into a TP dot-mapping file. The APIs to the deployment descriptors differ at deployment time and at development time. This interface provides a level of abstraction for the needed information."

One method is the constructor of the class, the other one is a private method that implements a function called by another private method.

2. Inspection

2.1 EjbConversionHelper

2.1.1 Naming Conventions

Point 1 not fulfilled:

The variables pFields (line 103), pIter (line 114), pKeys (line 120), pKeyMap (line 121) have a "p" that refers to different things:

- "p" for pFields stands for "persistent" so pFields can be changed in persFields (like the comment in line 102)
- "p" for pIter stands for "pseudo fields" so pIter can be changed in pseudoFieldsIter
- "p" for pKeys and pKeyMap stands for "primary" so pKeys can be changed in primaryKeys and pKeyMap can be changed in primaryKeyMap

The variables iter defined in line 87 is not very self-explicative. Should be like ejbsIter.

The variables fIter (line 105) and kIter (122) have only a letter for prefix ("f" stands for fields and "k" stands for keys). It's better to change them in fieldsIter and keysIter because they are still short names but more meaningful.

Point 5 not fulfilled:

Method iterator() (used in lines 87,105,114,122,133) should be a verb (like getIterator())

Method next() (used in lines 89, 107, 116, 124, 135) should be a verb (like getNext())

2.1.2 Indentions

No issues in this part of the checklist because the indentation is always correct (maybe done automatically by an ide) and isn't made by using the tabulation (but with the space character).

2.1.3 Braces

No issues in this part of the checklist because the braces are always correct with the "Kernighan and Ritchie" style (maybe done automatically by an ide).

Also, all the statements that open a block with only one instruction have the curly braces. Example in line 115, 150, 165.

2.1.4 File Organization

Blank lines are used wisely in order to separate related lines of code and in some cases there is a comment before the blocks of code.

Comments with lines like //------Comment------ are not used in this method.

2.1.5 Wrapping Lines

The only wrapping line is used in the long statement at the line 91 and it is used correctly (the line breaks after the operator "=").

There is no higher-level breaks.

Statement at the line 94 is correctly aligned with the beginning of the statement at the line 91.

2.1.6 Comments

Point 18 not fulfilled:

The comment at the line 95 is not correct because is in the middle of a block of code. It should be after the statement in the line 94 because that instruction is related with the following one.

The rest of the code seems to be wisely commented.

2.1.7 Java Source Files

No issues in this part of the checklist.

2.1.8 Package and Import Statements

No issues in this part of the checklist.

2.1.9 Class and Interface Declarations

Point 26 not fulfilled:

The class that contains this constructor method doesn't group methods by functionality but it list before the public methods and then the private ones.

Point 27 not fulfilled:

The constructor inspected is a long method. It can be divided in other methods in order to increase the legibility and the cohesion.

2.1.10 Initialization and Declarations

Point 8 not fulfilled:

The class that contains this constructor have two variables without the scope (generateFields, ensureValidation at lines 80 and 81).

Point 33 not fulfilled:

Declarations don't always appear at the beginning of blocks. The main block of the constructor there are the variables iter (line 87), rels (line 132) and relIter (133) that are declared after other instructions.

In the "if" block, starting from the line 90, the variables pers (line 99), pFiels (103), fieldMap (line 104), fIter (line 105), pseudoFields (line 113), pIter (line 114), pKeys (line 120), pKeyMap (line 121) and kIter (line 122) are not declared in at the beginning of the block.

In the "while" block, starting from the line 134, the variables sourceCMRField (line 149), sinkEjbName (line 155), sinkRels (line 156) and sinkCMRField (line 164).

2.1.11 Method Calls

No issues in this part of the checklist.

2.1.12 Arrays

No issues in this part of the checklist. Array lists and collections are always initialized when declared and there is no "out of index" issue.

2.1.13 Object Comparison

Point 40 not fulfilled:

Objects sourceRels (line 142), sourceCMRField (line 150), sinkRels (line 157) and sinkCMRField (line 165) are compared with "==" and not with "equals".

2.1.14 Output Format

No issues in this part of the checklist.

2.1.15 Computation, Comparisons and Assignments
No issues in this part of the checklist.

2.1.16 Exceptions
According to the javadoc, the methods called by this constructor don't throws exceptions. This constructor doesn't manage any type of exception.

2.1.17 Flow of Control
No issues in this part of the checklist.

2.1.18 Files
No issues in this part of the checklist.

2.2 getRealRelationRoleDescriptor
2.2.1 Naming Conventions
No issues in this part of the checklist.

2.2.2 Identations
No issues in this part of the checklist because the indentation is always correct (maybe done automatically by an ide) and isn't made by using the tabulation (but with the space character).

2.2.3 Braces
No issues in this part of the checklist because the braces are always correct with the "Kernighan and Ritchie" style (maybe done automatically by an ide).

Also, all the statements that open a block with only one instruction have the curly braces.

2.2.4 File Organization
No issues in this part of the checklist.

2.2.5 Wrapping Lines
No issues in this part of the checklist.

2.2.6 Comments
Point 18 not fulfilled:

There is no comments in this method. It may be useful for understanding better the code.

2.2.7 Java Source Files
No issues in this part of the checklist.

2.2.8 Package and Import Statements
No issues in this part of the checklist.

2.2.9 Class and Interface Declarations
No issues in this part of the checklist.

2.2.10 Initialization and Declarations
No issues in this part of the checklist.

2.2.11 Method Calls
No issues in this part of the checklist.

2.2.12 Arrays
No issues in this part of the checklist.

2.2.13 Object Comparison

No issues in this part of the checklist.

2.2.14 Output Format

No issues in this part of the checklist.

2.2.15 Computation, Comparisons and Assignments

No issues in this part of the checklist.

2.2.16 Exceptions

No issues in this part of the checklist.

2.2.17 Flow of Control

No issues in this part of the checklist.

2.2.18 Files

No issues in this part of the checklist.

2.3 Other problems

No other problem have been encountered.

3. References

The tools I used to create the inspection document are:

- Microsoft Office Word 2013: to redact and format this document;
- Notepad++: to see the java code of the class;
- Virtual Box + Ubuntu 15.10 + Subversion: to retrieve the code of Glassfish.

For redacting and writing this document I spent about 15 hours.