myTaxiService

Code Inspection

A.Y. 2015/2016

Politecnico di Milano

Version 1.0

Cattaneo Michela Gaia, matr. 791685
Barlocco Mattia, matr. 792735

January 5th, 2015

Contents

1 Introduction **2**
1.1 Purpose 2
1.2 Document structure 2

2 Classes Assigned **3**

3 Functional Roles **4**
3.1 addCollectionRelationship 4
3.2 addJoinTableEntry 4
3.3 prepareUpdateFieldSpecial 4
3.4 setForeignKey 4

4 Issues **5**

5 Other Problems **7**

1 Introduction

1.1 Purpose

This document represents the Code Inspection document, whose main goal is to examine source code in order to discover bugs, verifying that coding conventions are respected and look for potential issues. This part is necessary to improve the quality of the code and to assure that there are no errors in it.
This document is addressed to the project development teams, who need to work on code without mistakes before the production phase, in order to avoid future possible problems.

1.2 Document structure

This document is divided in five main parts:

- **Introduction:** this section describes the document in general and its purpose.
- **Classes assigned:** this section specifies the name of the classes and methods assigned to the group.
- **Functional roles:** this section shows the role of each method assigned and their functions in the class.
- **Issues:** this section verifies that all the points of the code inspection checklist are respected in the classes assigned.
- **Other problems:** this section deals with other possible issues found in the code, referring to the lines generating a bug and how it can create problems in the program.

2 Classes Assigned

- **Package:** com.sun.jdo.spi.persistence.support.sqlstore
- **Class:** SQLStateManager
- **Methods:**
  - addCollectionRelationship
  - setForeignKey
  - addJoinTableEntry
  - prepareUpdateFieldSpecial

3 Functional Roles

3.1 addCollectionRelationship

This method sets the relationship for the objects added to a collection relationship. Given a list of object to add to the relationship and a list of newly registered state managers, this method transforms and adds all the elements of the addedList to the newlyRegisteredSMs list, by checking that the elements are not null and making them autopersistent. Then it checks that the element added has not been deleted and updates the relationship in the data store. It is used when it is necessary to apply the updates in the data store, in particular it processes the updates in the collections by adding the relationships between them.

Evidence: javadoc, function calls.

3.2 setForeignKey

This method is used to set the foreign key corresponding to the relationship field. The fieldDesc and inverseFieldDesc parameters are used for the iteration on the local or foreign fields. This method calls setForeignKey, a method with other parameters that actually sets the local field corresponding to a foreign key column to the new value for the relationship update. This method is called in the processForeignKey in order to set the foreign key on the added object. It is useful when an update of relationships in the data store is needed.

Evidence: javadoc, function calls.

3.3 addJoinTableEntry

3.4 prepareUpdateFieldSpecial

4 Issues

- Class:
  - line 75: There is no JavaDoc for the class.
  - line 151: /** I18N message handler. */
    private final static ResourceBundle messages = I18NHelper.loadBundle(SQLStateManager.class);
    Here the ResourceBundle is declared as a constant and it is not capitalized, as it is said in the naming conventions at point 7 of the checklist.
  - line 219: private void registerInstance(boolean throwDuplicateException, ArrayList newlyRegisteredSMs, LifeCycleState oldstate);
    There is no JavaDoc for this method.
  - line 270: private void unsetMaskBit(int index, int mask);
    There is no JavaDoc for this method.
  - line 283: private void clearMask(int mask);
    There is no JavaDoc for this method.
  - line 308: private void newFieldMasks();
    There is no JavaDoc for this method.
  - line 400: public synchronized void replaceObjectField(String fieldName, Object o);
    There is no JavaDoc for this method.
  - line 408: public synchronized void makeDirty(String fieldName);
    There is no JavaDoc for this method.
  - line 508: public void makePresent(String fieldName, Object value);
    There is no JavaDoc for this method.
  - line 560: private void makeAutoPersistent(Object pc);
    There is no JavaDoc for this method.
  - line 1108: private boolean compareUpdatedFields(SQLStateManager next);
    There is no JavaDoc for this method.
  - line 1542: private void reset(boolean retainValues, boolean wasNew, boolean keepState);
    There is no JavaDoc for this method.
  - line 1706: private void markKeyFieldsPresent();
    There is no JavaDoc for this method.
  - line 2154: private SQLStateManager copyPersistent();
    There is no JavaDoc for this method.
  - line 2240: private Object cloneObjectMaybe(Object source);
    There is no JavaDoc for this method.
  - line 4087: private void assertNotPK(int fieldNumber);
    There is no JavaDoc for this method.
  - line 4093: private void assertPKUpdate(FieldDesc f, Object value);
    There is no JavaDoc for this method.

- line 77-162: these lines does not respect the point 25.D and 25.E of the checklist, the public static or instance variables are not declared before the private ones.

- addCollectionRelationship:

  - line 3384: SQLStateManager addedSM = getAddedSM(addedObject, newlyRegisteredSMs);
    This line exceeds the 80 characters suggested at point 13 of the checklist and it is possible to reduce the number of characters in this line, by wrapping after the comma in the getAddedSM call, as said at point 15 of the checklist.
  - line 3409: updateRelationshipInDataStore(fieldDesc, addedSM, null, inverseFieldDesc, false);
    This line exceeds the 80 characters suggested at point 13 of the checklist and it is possible to reduce the number of characters in this line, by wrapping after the commas that separates the parameters of the method, as said at point 15 of the checklist.

- addJoinTableEntry:

  - line
  - line

- prepareUpdateFieldSpecial:

  - line
  - line

5 Other Problems