****

2015-2016

**Code Inspection**

Morenzetti Sergio 791558.

Marcucci Antonio Pio 854473.

Released date: 05/01/2016

***Table of content :***

*1. Classes that were assigned to the group;*

*2. Functional role of assigned set of classes;*

*3. List of issues found by applying the checklist;*

*4. Any other problem you have highlighted.*

***1. Classes that were assigned to the group***

The class that was assigned to our group is RegisteredResource.java .

Its location is : appserver/transaction/jts/src/main/java/com/sun/jts/CosTransactions/RegisteredResources.java.

The method assigned is :

Name: commitOnePhase( )

Start Line:1470

***2. Functional role of assigned set of classes***

The class cluster assigned to our group is commitOnePhase void method. This method has the task to manage the commit phase of a resource object involved in a transaction, that is doing some updates on a resource object or managing eventually exception raised.

It works only if the resource object involved is just one and if it is a registered resource, otherwise the method ends.

Moreover there is a logger which records everything that happens during this process.

The method has a finite number of possible retries in order to perform the commit so that once exceeded this number, the process ends and throws a fatal error.

Before performing the commit, the method checks if the object that is referred to the resource is local or remote.

The resource object involved is marked with a status to know its state during the commit phase.

If the commit is done correctly and the resource object reference is proxy (local),than the resource object is updated and released. Otherwise a rollback exception is raised.

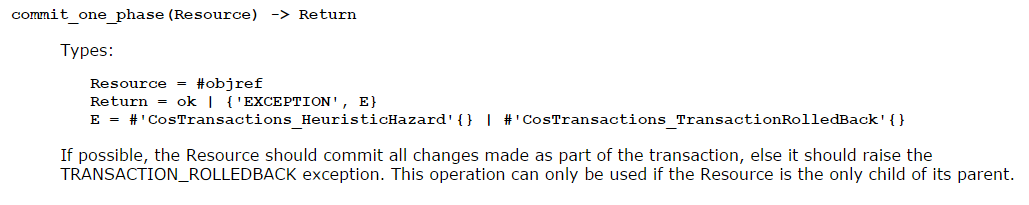
Several exceptions can be raised during the method performance, such as incorrect resource for this transaction, incorrect object reference of the resource, heuristic exception or rolled back exception. They are managed in order to avoid the lost of data integrity; for example in the case of rolled back exception, a massage is thrown and the method finishes and the resource object is not updated.

Finally we can say that this method is related to the management of the commit process of a resource involved in a transaction.

We have to say that the management ins not properly “total” because in some cases the exceptions are not properly managed but are just thrown to the caller method.

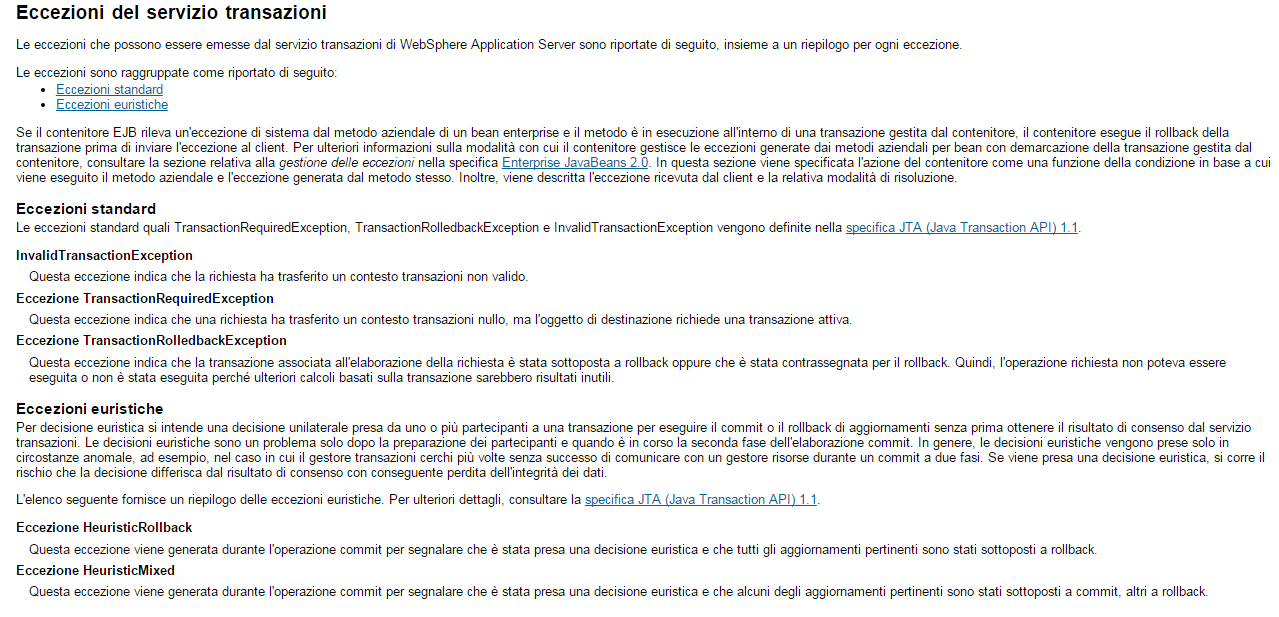
In order to better understand the role of the assigned method we have firstly read the code with relative comments, and then we have looked for some documentation on different web sites.

E.g. in order to understand what the method commit\_one\_phase() does we found some information on erlang.org web site, as follow(s) :



In our case no parameters are requested due to the fact that this method is called directly by the resource.

E.g. in order to understand the meaning of some exceptions we have found information on ibm web site as follow(s) :



Most parts of the code are understandable thanks to good comments wrote by the developers, which are placed in a smart position.

No java documentation related to this method was found.

***3. List of issues found by applying the checklist***

Those points of the checklist not present in the table below represent "errors" that we have not found in our portion of code.

We have only indicated class/code fragments that do not fulfill checklist points, explaining also the reasons.

|  |  |
| --- | --- |
| **Code line** | **Issue description (checklist point)** |
| 1601 | The variable e should be better specified, since it is not temporary and is used many times in that part of the code (2) |
| 1480 | rollback\_occurred should be rollbackOccurred because is not a constant variable but is an attribute. This modification should be applied also to its recurrences (6) |
| 1481 | outstanding\_resources attribute should be outstandingResources (6) |
| 1482 | retry\_limit should be retry\_limit (6) |
| 1483 | no\_of\_attempts should be noOfAttempts (6) |
| 1693 | The indention is not correctly done (8) |
| 1732 | The indention is not correctly done (8) |
| 1513 | Is used a tab instead of 4 spaces (9) |
| 1515 | Is used a tab instead of 4 spaces (9) |
| 1516 | Is used a tab instead of 4 spaces (9) |
| 1517-1519 | Is used a tab instead of 4 spaces (9) |
| 1562-1564 | Is used a tab instead of 4 spaces (9) |
| 1569 | Is used a tab instead of 4 spaces (9) |
| 1571-1573 | Is used a tab instead of 4 spaces (9) |
| 1601-1611 | Is used a tab instead of 4 spaces (9) |
| 1613-1619 (no 1616) | Is used a tab instead of 4 spaces (9) |
| 1656 | Is used a tab instead of 4 spaces (9) |
| 1658-1663 | Is used a tab instead of 4 spaces (9) |
| 1641 | Is used a tab instead of 4 spaces (9) |
| 1643-1648 | Is used a tab instead of 4 spaces (9) |
| 1561 | Since consistent bracing style must be used, either the preferred “Allman” style (first brace goes underneath the opening block) or the “Kernighan and Ritchie” style (first brace is on the same line of the instruction that opens the new block), not both. We assumed to be used the second one. So this line should be adequately corrected (10) |
| 1570 | Since consistent bracing style must be used, the brace should be put on the same line of the instruction that opens the new block (10) |
| 1506-1507 | The if statement that have only one statement to execute have to be surrounded by curly braces (11) |
| 1711-1713 | The if statement that have only one statement to execute have to be surrounded by curly braces (11) |
| From 1610 to 1617 | The if and else statements that have only one statement to execute have to be surrounded by curly braces (11) |
| 1515 | A new statement must be aligned with the beginning of the expression at the same level as the previous line (17) |
| 1519 | A new statement must be aligned with the beginning of the expression at the same level as the previous line (17) |
| From 1601 to 1622 | A new statement must be aligned with the beginning of the expression at the same level as the previous line (17) |
| 1644 | A new statement must be aligned with the beginning of the expression at the same level as the previous line (17) |
| 1648 | A new statement must be aligned with the beginning of the expression at the same level as the previous line (17) |
| 1659 | A new statement must be aligned with the beginning of the expression at the same level as the previous line (17) |
| 1663 | A new statement must be aligned with the beginning of the expression at the same level as the previous line (17) |
| 1669 | A new statement must be aligned with the beginning of the expression at the same level as the previous line (17) |
| From 1694 to 1699 | A new statement must be aligned with the beginning of the expression at the same level as the previous line (17) |
| 1485 | Just one missing word; better to write “First, get the commit retry count” instead of “First, get the retry count” (18) |
| From 1487 to 1504 | This part of code is commented because it has been later replaced by a smaller set of instruction; initially this part of code is reasonable to better understand the changes, but later we thought that it is not so necessary and it should be deleted (19) |
| From 1745 to 1753 | This part of code is commented because it has been later replaced by a smaller set of instruction; initially this part of code is reasonable to better understand the changes, but later we thought that it is not so necessary and it should be deleted (19) |
| From 1513 to 1519 and from 1641 to 1648 | In this two sections there is duplication of code, indeed there is the same logger action management and exception throwing. They differ only for a parameter (27) |
| From 1641 to 1648 and from 1656 to 1663 | In this two sections there is duplication of code, indeed there is the same logger action management and exception throwing (27) |
| 1525 | No checking on the size of the resourceObject array. It is better to check it in order to be sure that an exception will not be raised (37) |
| 1531 | No checking on the size of the resourceObject array. It is better to check it in order to be sure that an exception will not be raised (37) |
| 1644 | The var msg is missing info about the type of exception. It is better to add info about the NotPrepared exception to understand the reasons for raising it (42) |
| 1674 | It is needed an appropriate action in this catch block, because if the “sleep” call generate an exception it is better to know that the problem is generated from this part of the code (53) |

***4. Any other problem you have highlighted***

The part of the code that we think is not correctly written, is the one related to the exception handling of the commit\_one\_phase() method.

The problem is the usage of the “instance of” statement: we have learnt that this statement is a bad practice in a case like that, indeed is preferable to substitute all the conditional cases stated in the catch block and rewrite the code adding a catch block for each possible exception.