**Code inspection**

|  |  |  |
| --- | --- | --- |
| **Ass.** | **Description** | **line** |
| **30** | The object currResource is created without calling the relative constructor  The object checkProxy is created without calling the relative constructor | 466  505 |
| **31** | At line 592 there is an used object without any reference. | 592 |
| **32** | All the variables are initialized when declared. | 452  453  454  465  466  505  506  549  550 |
| **33** | All the declarations appear at the beginning of a block. |
| **34** | All the parameters used when calling a method are in the correct order. | - |
| **35** | * commit\_one\_phase() should be commitOnePhase(). * .getProxyChecker seems that it doesn’t exist. The callable methods on Configuration are:   getConfiguration()  getInstance(…)  setConfiguration(…)   * .isProxy(currResource) doesn’t correspond to any method in the class. | 479  505  506 |
| **36** | The .set method called on an ArrayList replace the specified element by the index with the relative given by the parameter but returns the element previously at the specified position. | 480  490  558  624  645  655 |
| **37** | There are no off-by-one errors in array indexing. | - |
| **38** | For the “resourceObjects” array there is only a getter with an index that returns the element at the specified position in this list. The for cycle exits when one of this condition is not verified:  **i < nRes** && result != Vote.VoteRollback;  Everytime nRes was incremented, an item has been added to the array and that’s why it isn’t possible to go out of bounds.  The same thing is valid for the “resourceStates” array too. | - |
| **39** | All the arrays in our method have been correctly initialized and each time an item was added, the right way to add it to the dynamic array had been invoked. | - |
| **40** | All the comparison are done using ‘==’ instead of .equalsTo  laoResource == null  result == Vote.VoteCommit  currResult == null  currResult == Vote.VoteCommit  logSection == null  result == Vote.VoteReadOnly  currResult == Vote.VoteRollback  result == Vote.VoteCommit | 469  470  520  599  602  608  629  635 |
| **41** | About **Output Format:** in our piece of code we don’t have any portion of code that is going to display something to the user that have to be comprehensive or guides to how to correct the problem. | - |
| **42** | - |
| **43** | - |
| **44** |  |  |
| **45** | The order of evaluation and computation is correct | - |
| **46** | The highlighted parentheses in bold can be omitted because the precedence of the arithmetic and logic operators is preserved anyway.  **(**i == nRes – 1**)** && lastXAResCommit && **(**laoResource == null**)** && | 469 |
| **47** | There are no arithmetic divisions in the method | - |
| **48** | All the integer arithmetic operations don’t cause unexpected truncation/rounding due to having only addition and difference. | - |
| **49** | All the Boolean comparisons are correct | - |
| **50** | The error condition in every try-catch expression is correctly managed. Essentially if a try branch in which there’s a communication about commit fails, in the catch branch the program will communicate the intention of a rollback action. | - |
| **51** | .get(i) returns the element at the specified position in this list. The program force this element to be a Resource type  (Resource) resourceObjects.get(i); | 466 |
| **52** | There is no throw-catch expression that ensure a no [IndexOutOfBoundsException](https://docs.oracle.com/javase/8/docs/api/java/lang/IndexOutOfBoundsException.html) because the code already guarantee this case. But it’s always better to have managed this exception in order to manage any error that can happen if, for example, something went wrong before. | - |
| **53** | The appropriation action is taken whenever a try fails. | - |
| **54** | About **Flow of Control:** in our piece of code we don’t have **switch statement** and loops. | - |
| **55** |
| **56** |
| **57** | About **Files:** in our method there is no reference to external file or any other operations that concern them. | - |
| **58** |
| **59** |
| **60** |