**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 object used that hasn’t any references | 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 method are in the correct order. | - |
| **35** | * commit\_one\_phase() should be commitOnePhase() * .getProxyChecker appear that no exists. The methods callable 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** |  |  |
| **38** |  |  |
| **39** |  |  |
| **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 parenthesis in bold can be omitted  **(**i == nRes – 1**)** && lastXAResCommit && **(**laoResource == null**)** && | 469 |
| **47** | There is no arithmetic division in the method | - |
| **48** | All the integer arithmetic operation don’t cause unexpected truncation/rounding due to avoid only addition and difference. | - |
| **49** | All the Boolean comparison are correct | - |
| **50** | The error condition in all try-catch expression is correctly managed. | - |
| **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. | - |
| **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. | - |
| **58** |
| **59** |
| **60** |