**COMP 3505 - Software Testing**

**Assignment Report #2 – Specification-Based Test Generation**

|  |  |
| --- | --- |
| Group #: | 4 |
| Student Names: | Kevin Ruiz |
| Charlemagne Palisoc |
| Ryan Wong |

1. Which test case design techniques have you used for designing your test cases, show one example from your test cases for each technique that you have used (If you have not used one of the techniques in your tests, you do not have to create an example).

* Decision Table

Combine Method

R1 = VALID RANGE

R2 = VALID RANGE

R3 = NULL

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| Range 1 | R1 | R1 | R2 | R3 | R3 |
| Range 2 | R2 | R3 | R3 | R1 | R3 |
| Actions |  |  |  |  |  |
| Return combined range value | X |  |  |  |  |
| Return null |  |  |  |  | X |
| Return the same valid range |  | X | X | X |  |

* BVA

Constrain Method - given the valid range [-1,10], perform boundary value analysis

|  |  |  |
| --- | --- | --- |
| Case# | Specified Value | Expected Output |
| 1 | -15 | -1 |
| 2 | -1 | -1 |
| 3 | 1 | 1 |
| 4 | 6.5 | 6.5 |
| 5 | 9 | 9 |
| 6 | 10 | 10 |
| 7 | 11 | 10 |

Weak normal ECP

* Testing for the expandToInclude method
* cases for a value to expand to equal to the upper and lower boundary (cases 2 & 3) were designed but not implemented as they were determined to be in the same expected partition for a value that is contained in the range.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Range input** | **Value input** | **Expected Output** | **Test data** |
| 1 | Any real number lower boundary (x) to any real number upper boundary (y) | Any real number (k) between (x) to (y) | A range that includes the value (k), and all of (x) to (y) | X = -10.0  Y = 10.0  K = 5.0 |
| 2 | Any real number lower boundary (x) to any real number upper boundary (y) | k = x | A range that includes the value (k), and all of (x) to (y) | X = -10.0  Y = 10.0  K = -10.0 |
| 3 | Any real number lower boundary(x) to any real number upper boundary (y) | k = y | A range that includes the value (k), and all of (x) to (y) | X = -10.0  Y = 10.0  K = 10.0 |
| 4 | Any real number lower boundary (x) to any real number upper boundary (y) | k < x | A range that includes the value (k), (k) to (y) | X = -10.0  Y = 10.0  K = -11.0 |
| 5 | Any real number lower boundary (x) to any real number upper boundary (y) | k > y | A range that includes the value (k), (x) to (k) | X = -10.0  Y = 10.0  K = 11.0 |
| 6 | Null | Any real number (k) | A range that includes the value (k), and boundaries of (x) to (y) | X,Y = null  K = 5.0 |

1. If you were to develop test cases for org.jfree.data.DataUtilities, how can this dependency affect test cases that you develop for DataUtilities class. Discuss your answer (at this point, you do not need to develop test cases for this class).

* The DataUtilities class is dependent on some other classes for its testing/use. The methods for calculateColumnTotals and calculateRowTotal both require an instance of the Values2D object for use. The method getCumulativePercentages requires an instance of the KeyedValues obj. This dependence interferes with the isolation of testing methods and creates planning and executing issues for their testing.
* If these dependent classes behave incorrectly, the resulting test cases that we develop for DataUtilities may result in failures. But these failures are not necessarily caused by an incorrect implementation of the DataUtilities functions but as a result of issues in the dependencies. This can have some effect on the order of our test cases, some test cases will have to be executed when a certain case/parent case has met a certain condition.
* Testing of the DataUtilities methods will need to be done after the testing of the interface Values2D and KeyedValues interface methods.

1. Difficulties encountered, challenges overcome, and lessons learned from performing the assignment.

* I think there is an innate feeling of creating and coding code in eclipse as soon as we finished reading the assignment document. It was challenging to approach it in a more documented way, and applying the testing techniques we learned from the class. This way instead of blindly creating test cases that will result in a pass jUnit execution, it was important that we read and understood the specification from the documentations, this way we can start writing code that will test the expected behaviour of the function. We learned that testing is not about getting your test cases to pass, but instead, it was about making sure that the methods were functioning correctly, and returning the expected outputs.
* At the beginning of writing the test cases, there were some thoughts of using the other class methods to confirm if the functionality was returning the correct output. For example with combined ranged, combining two ranges [-1,10] and [11,20] should return [-1,20]. To verify this in the beginning I tried to output the returned object into the console and receive a totally different range. It provided the thought that we could not trust the resulting output as well since this function might have been implemented incorrectly as well therefore it is reporting different range. The lesson learned here is that in situations where we do not have access to the source code, the reliable way that we can test is to always consider the specifications.
* We also found that if any of the functionalities aren’t working or behaving as they are expected, it could result in some more of the functionalities to show the defects as well. Hence when a functionA is dependant on another functionB that has some defects, this functionA will also inherit the issue and cause a test to fail. Case in point a test case that fails may not necessarily be because of the function being tested but can be a result of another deeper issue.
* It is also necessary that we understood the documentation properly as it can cause issues when testing functionality. This can result in confusion in whether or not the test case failed because of the test code or if it failed because the method/functionality has some issues.