Martin Olsson, Nicklas Svensson TestLabb DA489A

QUESTION 1: What type of code coverage is measured by the tool (statement, branch, path or condition)? Is it possible to choose more than one type?

branch coverage

QUESTION 2: What is the maximum code coverage you reach when executing your tests?

93,1% for line coverage
92,9% branch coverage
75% method coverage.

QUESTION 3: Which defects did you find?
Always big O. Additionally, there were missing consonants, G, D missing in rover.java class constants String.

QUESTION 4: How did you correct the defects?
Always adding small o and adding constants,

QUESTION 5: Do all test cases pass once the defects have been corrected?
Yes

QUESTION 6: What did you need to rewrite in order to use unit test?
We already had two methods in place. We needed to test them using JUnit with various inputs and BVA.

QUESTION 7: Which level of code coverage did you get when executing your tests?
100%

QUESTION 8: Did you find any bugs? Yes/No
Well decimal was buggy but could be solved thanks to the test cases.

QUESTION 9: What is the rationale for the Values you used in your BVA tests?
Testing according to BVA, which is based on equivalence partitioning theory. In BVA one is supposed to test the boundaries where issues tend to lurk. We tested for illegal values (-1) and zero values for different parameters.

QUESTION (TDD1):
How many state transitions (compare 0-switch, 1-switch, etc. mentioned in the lecture) are reasonable to test in this case? Well, it’s responsible to test around the boundaries. One below the boundary, one for normal value, and one above the boundary.
We choose only 6 test case for 0-swtich with the argument that 24 BVA test and 6 illegal tests has been used, thus a total of 36 is judged as adequate.

QUESTION (TDD2):

How much code coverage do you reach with your tests?
100% for line & bransch & statement coverage