

Assignment 2

White box testing

Model-Based Testing
DIT848/GU and TDA260/Chalmers

March, 2012

1 Introduction

Code coverage is used as a metric to measure quality of tests. The purpose of this assignment is to become familiar with code coverage analysis, which is one of the most common white box testing methods. Using this method you can find parts of the code that are not covered by the test cases, and thus guide you to come with new test cases to increase coverage.

2 Submitting your work

If you want to have feedback on your assignment, check with Hamid Ebadi (hamide@student.chalmers.se) on how (and when) to submit. If you want to submit, please attach a .zip or .tar.gz archive, containing your source code and .txt, .pdf or .doc file describing your answers. Please name your file with the number of assignment and your (last) name as in the following example: `ebadi_assignment02.zip`.

3 Structure

The file `calculator2.zip` contains the following files:

- `src/Calculator2.java`: Interface and implementation for a modification of module implementing the functionality of a simple calculator. This version fixes problems with the first version and also can evaluate expressions that contain parentheses, e.g. `55-(2*(3+7))`.
- `src/Calculator2Test.java`: Example showing three sample JUnit test cases for the calculator class.

4 What to do

The file `calculator2.zip` contains a modification of `calculator.zip` distributed last week. Run the existing test suite using EclEmma. Create more test cases to increase code coverage as much as possible. Does your test suite have full statement and branch coverage? If not, add test cases to get full coverage, or motivate that it is not possible.