Test Report

Pitclipse mutation tests

Check	Mutation Coverage	Test Strength	Mutation Types
Category A	40/40	40/40	• replaced double division with multiplication • replaced double multiplication with division • replaced return value with null • replaced integer addition with subtraction • negated conditional • removed call to log() method
HalsteadDifficultyCheck	9/9	9/9	
HalsteadEffortCheck	15/15	15/15	
HalsteadLengthCheck	5/5	5/5	
HalsteadVocabularyCheck	4/4	4/4	
HalsteadVolumeCheck	7/7	7/7	
Category B	37/37	37/37	• replaced return value with null • replaced boolean return with false • replaced integer addition with subtraction • removed call to log() method • negated conditional
CommentsCountCheck	6/6	6/6	
ExpressionCountCheck	5/5	5/5	
LineOfCommentCountCheck	11/11	11/11	
LoopStatementCountCheck	5/5	5/5	
OperandsCountCheck	5/5	5/5	
OperatorsCountCheck	5/5	5/5	

Fault models

Category A:

1. HalsteadLengthCheck:
   1.1 Count paring operators as two, such as (), {}, []
   1.2 Count operand/operator in a comment or a string literal
2. HalsteadVocabularyCheck:
   2.1 Miss-counting unique operands/operators (overcount or undercount)
   2.2 Counting compound operator such as “+=” as 2 separate operators
3. HalsteadVolumeCheck:
   3.1 Incorrect program length or vocabulary
4. HalsteadDifficultyCheck:

    4.1 Incorrect count of operators or operands
5. HalsteadEffortCheck:
    5.1 Incorrect volume or difficulty

Category B:

1. Number of comments
    1.1 Count multiple “//” on a same line as multiple comments
    1.2 Comment inside a string literal
    1.3 Not counting block comment
    1.4 Miss comments that are behind a piece of code on a line
2. Number of lines of comments
    2.1 An empty line between comment blocks may be incorrectly counted as a comment line.
    2.2 Count multiple block comments in one line as multiple lines
    2.3 Count multiple “//” on one line as multiple lines
3. Number of looping statements
    3.1 Miss counts nested loop
    3.2 Count key word in in a comment or string literal
4. Number of operators
    4.1 Compound operators like +=, -=, ++, -- may be counted as individual operators rather than compound ones.
    4.2 Post-Increment and Pre-Increment Operators might be counted as operands
    4.3 Count a pair of brackets as 2
5. Number of operands
    5.1 Constant values are not counted as operands
    5.2 Miss count array indices (a[5], a[6])
    5.3 Miss count operands inside an expression
6. Number of expressions
    6.1 Count expression inside comment or string
    6.2 Miss counts conditional expression inside in control flow statements (if, for, while)

Test Cases:

aCheckModels.java for Category A fault models

bCheckModels.java for Category B fault models

Class Testing

For the check class alone basic unit testing is feasible and it wouldn’t be too different form traditional testing. Integration class testing might be able to uncover more bugs (such as the requirement for getDefaultTokens(), getRequiredTokens(), and log() methods which seem not to have any impact in traditional Whitebox and Blackbox testing), but it is quite unpractical because we don’t really know how the classes in CheckStyle interact with each other. Testing method sequence is somewhat equivalent to Blackbox testing since it test the behavior of the class. State-Based testing for the check classes would not make a big impact since the state of the check class does not really matter in the interaction with other classes (even its count variables would not impact how other classes work).