Dylan Gyori
Check test case report
11/17/2024

CommentLineCountCheck Test Cases: all passed with 100% branch and line coverage.

- **isCommentNodesRequiredTest:** tests if comment nodes required method returns true.
- **getTokensTest:** tests if getDefaultTokens and getAcceptableTokens methods return an array with one COMMENT_CONTENT token and getRequiredTokens returns an empty array.
- **beginTreeTest:** tests if beginTree method sets comment line count to 0.
- **finishTreeTest:** tests if frinishTreeTest passes correct message to AbstractCheck's log method.
- **visitTokenTest:** tests if visitToken method correctly counts number of lines a multiline comment and single line comment take up.

NumCommentsCheck Test Cases: all passed with 100% branch and line coverage.

- **isCommentNodesRequiredTest:** tests if comment nodes required method returns true.
- **getTokensTest:** tests if getDefaultTokens and getAcceptableTokens methods return an array with one COMMENT_CONTENT token and getRequiredTokens returns an empty array.
- **beginTreeTest:** tests if beginTree method sets comment count to 0.
- **finishTreeTest:** tests if frinishTreeTest passes correct message to AbstractCheck's log method.
- **visitTokenTest:** tests if visitToken method increments count on token visit.

ExpresssionCountCheck Test Cases: all passed with 100% branch and line coverage.

- **getTokensTest:** tests if getDefaultTokens and getAcceptableTokens methods return an array with one EXPR token and getRequiredTokens returns an empty array.
- **beginTreeTest:** tests if beginTree method sets count to 0.
- **finishTreeTest:** tests if frinishTreeTest passes correct message to AbstractCheck's log method.
- **visitTokenTest:** tests if visitToken method increments count on token visit.

LoopCountCheck Test Cases: all passed with 100% branch and line coverage.

- **getTokensTest:** tests if getDefaultTokens and getAcceptableTokens methods return an array of LITERAL_FOR, DO_WHILE, and LITERAL_WHILE tokens and getRequiredTokens returns an empty array.
- **beginTreeTest:** tests if beginTree method sets count to 0.
- **finishTreeTest:** tests if frinishTreeTest passes correct message to AbstractCheck's log method.
- **visitTokenTest:** tests if visitToken method increments count on token visit.

NumOperandsCheck Test Cases: all passed with 100% branch and line coverage.

- **getTokensTest:** tests if getDefaultTokens and getAcceptableTokens methods return an array of operand tokens and getRequiredTokens returns an empty array.
- **beginTreeTest:** tests if beginTree method sets count to 0.

- **finishTreeTest:** tests if frinishTreeTest passes correct message to AbstractCheck's log method.
- **visitTokenTest:** tests if visitToken method increments count on token visit.

NumOperatorsCheck Test Cases: all passed with 100% branch and line coverage.

- **getTokensTest:** tests if getDefaultTokens and getAcceptableTokens methods return an array of operator tokens and getRequiredTokens returns an empty array.
- **beginTreeTest:** tests if beginTree method sets count to 0.
- **finishTreeTest:** tests if frinishTreeTest passes correct message to AbstractCheck's log method.
- **visitTokenTest:** tests if visitToken method increments count on token visit.

HalsteadLengthCheck Test Cases: all passed with 100% branch and line coverage.

- **getTokensTest:** tests if getDefaultTokens and getAcceptableTokens methods return an array of operator and operand tokens and getRequiredTokens returns an empty array.
- **beginTreeTest:** tests if beginTree method sets count to 0.
- **finishTreeTest:** tests if frinishTreeTest passes correct message to AbstractCheck's log method.
- **visitTokenTest:** tests if visitToken method increments count on token visit.

HalsteadVocabularyCheck Test Cases: all passed with 100% branch and line coverage.

- **getTokensTest:** tests if getDefaultTokens and getAcceptableTokens methods return an array of operator and operand tokens and getRequiredTokens returns an empty array.
- **beginTreeTest:** tests if beginTree method sets count to 0.
- **finishTreeTest:** tests if frinishTreeTest passes correct message to AbstractCheck's log method.
- **visitTokenTest:** tests if visitToken method increments count only when unique tokens are visited.

HalsteadVolumeCheck Test Cases: all passed with 100% branch and line coverage.

- **getTokensTest:** tests if getDefaultTokens and getAcceptableTokens methods return an array of operator and operand tokens and getRequiredTokens returns an empty array.
- **beginTreeTest:** tests if beginTree method sets halstead vocabulary and length to 0.
- **finishTreeTest:** tests if frinishTreeTest passes correct message to AbstractCheck's log method when vocabulary is equal to zero or one.
- **visitTokenTest:** tests if visitToken method increments vocabulary once but length multiple times on visit of duplicate tokens.

HalsteadDifficultyCheck Test Cases: all passed with 100% branch and line coverage.

- **getTokensTest:** tests if getDefaultTokens and getAcceptableTokens methods return an array of operator and operand tokens and getRequiredTokens returns an empty array.
- **beginTreeTest:** tests if beginTree method sets unique operators, unique operands, and total operands to 0.

- **finishTreeTest:** tests if frinishTreeTest passes correct message to AbstractCheck's log method when no or one operand is counted.
- **visitTokenTest:** tests if visitToken method increments unique operators once when duplicate operators are visited, then it checks if unique operands are incremented once but total operands are incremented multiple times when duplicate operands are visited.

HalsteadEffortCheck Test Cases: all passed with 100% branch and line coverage.

- **getTokensTest:** tests if getDefaultTokens and getAcceptableTokens methods return an array of operator and operand tokens and getRequiredTokens returns an empty array.
- **beginTreeTest:** tests if beginTree method sets unique operators, unique operands, total operators, and total operands to 0.
- **finishTreeTest:** tests if frinishTreeTest passes correct message to AbstractCheck's log method when no or one operand is counted.
- **visitTokenTest:** tests if visitToken method increments unique operators once but total operators multiple times when duplicate operators are visited, then it checks if unique operands are incremented once but total operands are incremented multiple times when duplicate operands are visited.