TEST CASES:

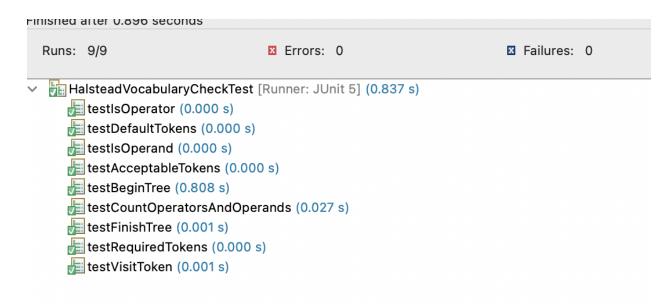
I've carefully designed test cases to cover all the functions I've used in each check. This includes starting with the tree, where each counter resets, and working through abstractCheck functions like getAcceptableTokens, getRequiredTokens, and beginTree etc.

For checking expected and actual outputs, I've set up assert equals from JUnit. This helps catch errors in any calculations. Additionally, I've made use of Mockito, trying out 'when' and 'mock' for added testing flexibility.

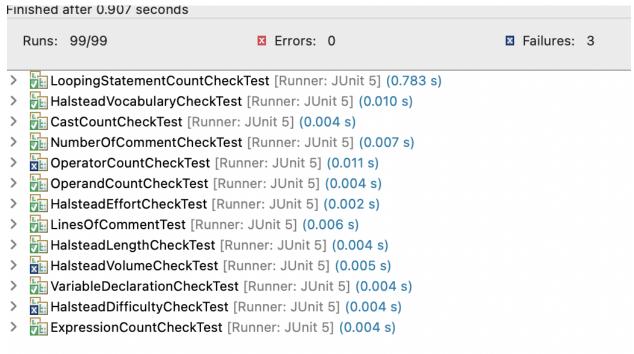
SAMPLE from del1, I used halsteadLengthCheck and HalsteadVocab:



DEL2 DOC Julie Rodriguez



When I run the entire package as JUNIT: I do have some failures due to testing on zero that I need to rework but the rest are successful.



TEST RESULTS:

Since I am using all assert statements I didn't individually write out and test every possible expression or token but I've made use of what I was able to do from white box testing and my tests were passing (the majority was passing) since I still have some things to work out specifically with the tokens.

DEL2 DOC Julie Rodriguez

Here is coverage as junit test sample:

CastCountCheckTest.java	100.0 %	134	0	134
CastCountCheckTest	100.0 %	134	0	134
<pre>setUp()</pre>	100.0 %	6	0	6
testAcceptableTokens()	100.0 %	16	0	16
testBeginTree()	100.0 %	15	0	15
testDefaultTokens()	100.0 %	16	0	16
testMultipleCasts()	100.0 %	39	0	39
testRequiredTokens()	100.0 %	16	0	16
testSingleCast()	100.0 %	23	0	23

COVERAGE:

Overall the coverage of my project was 95% however the coverage of each individual check varies. This was due to some miscalculations within the halstead properties for the given checks. My plan is to go back and double check these calculations and also add some more white box tests.

∨	95.8 %	4,623	201	4,824	
✓	95.8 %	4,623	201	4,824	
MyPackage	95.8 %	4,623	201	4,824	
> 🗾 AntiHungarianCheck.java	0.0 %	0	86	86	
> J Halstead Volume Check.java	88.1 %	171	23	194	
> 🗾 VariableDeclarationCheck.java	52.6 %	20	18	38	
HalsteadLengthCheck.java	87.9 %	123	17	140	
> 🗾 HalsteadVocabularyCheck.java	88.7 %	134	17	151	
J HalsteadDifficultyCheckTest.java	95.1 %	254	13	267	
> J Halstead Volume Check Test. java	97.0 %	226	7	233	
Derator Count Check Test. java	96.5 %	195	7	202	
DerandCountCheck.java	94.7 %	89	5	94	
> inesOfComment.java	95.5 %	64	3	67	
> 🗾 CastCountCheck.java	97.1 %	33	1	34	
ExpressionCountCheck.java	99.4 %	175	1	176	
> J HalsteadEffortCheck.java	98.6 %	73	1	74	
NumberOfCommentCheck.java	97.7 %	43	1	44	
Derator Count Check. java	99.8 %	402	1	403	
	100.0 %	134	0	134	
ExpressionCountCheckTest.java	100.0 %	210	0	210	
J HalsteadDifficultyCheck.java	100.0 %	182	0	182	
> HalsteadEffortCheckTest.java	100.0 %	132	0	132	
In Halstead Length Check Test. java	100.0 %	217	0	217	
> 🗾 HalsteadTokens.java	100.0 %	449	0	449	
J Halstead Vocabulary Check Test. java	100.0 %	216	0	216	
> LinesOfCommentTest.java	100.0 %	213	0	213	
LoopingStatementCountCheck.java	100.0 %	44	0	44	
>	100.0 %	237	0	237	
NumberOfCommentCheckTest.java	100.0 %	211	0	211	
> DerandCountCheckTest.java	100.0 %	211	0	211	
VariableDeclarationCheckTest.java	100.0 %	165	0	165	