Assignment 4

Marcos Benitez - MXB220120
Xzavier Matthew Vera Cruz - XAV230000
Sree Laasya Duddupudi - scd230191

1. EvoSuite

We will first look at EvoSuite

Automated Test Generation: One of the major strengths of Evosuite is automated test generation. It automatically creates unit tests for all methods without extra effort from a developer. This saves a lot of time and manual effort.

High Code Coverage: Another strength is high code coverage. Evosuite uses search-based algorithms to check as many execution paths as possible and this ensures maximum code coverage in testing. This helps reduce the number of hidden bugs as there is a better chance of them being found.

Regression Support: A really helpful feature is strong regression support. This means it can quickly regenerate and reuse tests after any changes in the code to detect new bugs.This ensures that bugs are caught quickly and that the test suite is the latest version which is especially helpful in large projects where modifications are frequent.

Automatic Assertions: Another great feature is assertions are automatically added to the test cases. These assertions help check whether the program works as expected and makes it more reliable.

Mutation Testing: Evosuite also uses mutation testing which checks how effective the tests are by adding in small bugs in the source code to check if the generated tests can find them.This shows how strong the tests are at finding real bugs.

2.Randoop

Now we can look over the strengths of Randoop

Simplicity: One big benefit of using Randoop is the simplicity of generation test cases. It just needs a classpath and the specific classes you want to test. The generated test cases are in standard JUnit files making it easier to do regression testing because you can save the output files for future testing.

Speed & Automation: Although Randoop is very easy to run, it is still very fast with creating thousands of test cases in seconds. Because of this, if you want to create new test cases after implementing major features to an application, you don’t have to worry about time when creating new test cases for the new features.

Code Coverage: With Randoop's random and feedback-based exploration, Randoop creates test cases that reach high code coverage. Meaning that it creates high quality test cases that cover many possibilities that a human may overlook or not even consider. Further making Randoop a good choice for regression testing.

3. ChatGPT

Strengths of ChatGPT-based Test Generation:

Readability: ChatGPT-based test generation has better readability and maintainability, because it can include comments about the code and ChatGPT can explain the code line by line for the coder.

Creativity: ChatGPT can create complex and creative tests because of its ability to create tests from prompts. This allows the tester to potentially ask ChatGPT to create complex and creative test cases, which includes scenarios like combining multiple methods or testing how methods interact with each other.

Accuracy: ChatGPT can be more accurate in certain cases. This is because ChatGPT can learn from the prompts and code, so that it can provide more contextual based tests. Also, the fact that the coverage can be determined by the tester based on their prompts. ChatGPT can still struggle with creating buggy tests, and requires human validation to ensure the quality and accuracy of the tests being generated.

4. ChatGPT Test Case Generation

We used ChatGpt to create the test cases by first providing ChatGPT with the code that needed to be tested. Then we asked it to provide tests for all the methods and edge cases. From there we revised the prompt, as well as reviewed the generated tests to cover missing cases and be more accurate.

5. Comparison

Approach:

- EvoSuite - It uses a search-based genetic algorithm to evolve tests for maximum coverage.

- Randoop - It uses randomized test generation based on sequences of method calls.
- ChatGPT - It uses natural language to create human-readable tests based on understanding of the code logic.

Generated Test Quality:

- EvoSuite - It generates precise tests which have plenty of assertions but they may be a bit complex.
- Randoop - It creates simple and fast tests but it often lacks assertions or logical validation.
- ChatGPT - It generates readable tests but may miss edge cases or rely on human refinement.

Coverage:

- EvoSuite- It generally has the highest statement and branch coverage due to a hybrid optimization approach.
- Randoop- It has moderate coverage depending on randomness and execution time.
- ChatGPT - It generally has lower coverage, because it depends on human-written prompts and understanding.

6. Tool Preference

Overall, Evosuite is the preferred tool because it has the highest amount of coverage and robustness. While Randoop and ChatGPT have their benefits, Evosuite is able to create highly precise tests with high amounts of coverage. While ChatGPT has arguably more readability and has the ability to gain more contextual background because of its ability to take in human written prompts, ChatGPT needs a lot of human assistance, as it is prone to making buggy tests. The need for human assistance defeats the purpose of test generation, which evosuite was designed and optimized for. All in all, Evosuite is the preferred approach for automatic test generation for its robustness, optimization, and unique features that differentiate it from Randoop and ChatGPT.

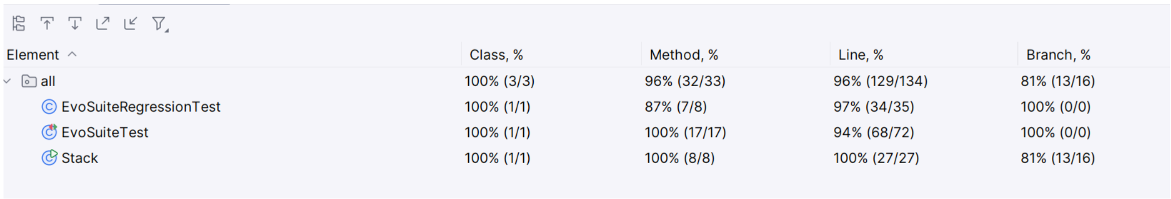
7. Improvement Ideas

Evosuite and Randoop: A feature that would create explanations for each test, as this would increase their main flaws of readability and maintainability. As well as, maybe a feature that allows the coder to provide some sort of contextual background to increase the quality of the tests being made.
ChatGPT: ChatGPT needs just an overall improvement on making less buggy tests, as well as a more efficient way to apply these tests within code. The biggest problem for ChatGPT was the buggy nature, as well as the constant need for human validation to ensure the quality of the tests.

8. Coverage Table

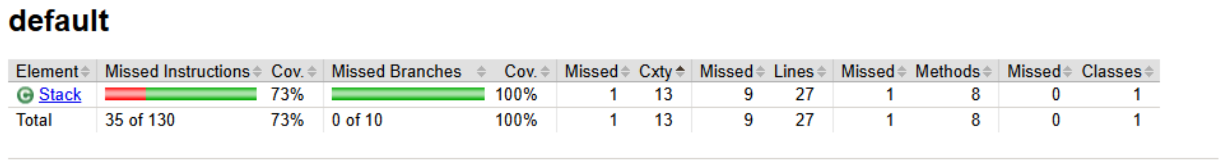
	Class	Methods	Line	Branch
EvoSuite	100%	100%	94%	100%
Randoop	73%	88%	67%	100%
ChatGPT	100%	88%	73%	80%

EvoSuite



Element	Class, %	Method, %	Line, %	Branch, %
all	100% (3/3)	96% (32/33)	96% (129/134)	81% (13/16)
EvoSuiteRegressionTest	100% (1/1)	87% (7/8)	97% (34/35)	100% (0/0)
EvoSuiteTest	100% (1/1)	100% (17/17)	94% (68/72)	100% (0/0)
Stack	100% (1/1)	100% (8/8)	100% (27/27)	81% (13/16)

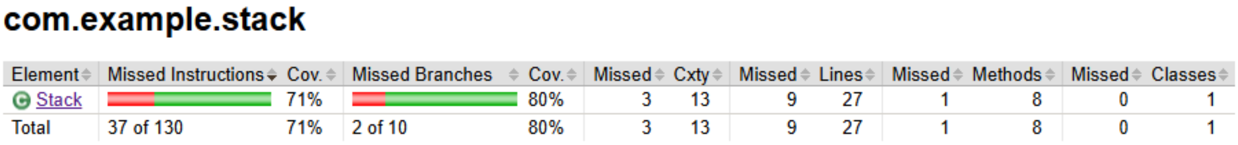
Randoop



default

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Stack	🟥🟩	73%	🟩	100%	1	13	9	27	1	8	0	1
Total	35 of 130	73%	0 of 10	100%	1	13	9	27	1	8	0	1

ChatGPT



com.example.stack

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Stack	🟥🟩	71%	🟥🟩	80%	3	13	9	27	1	8	0	1
Total	37 of 130	71%	2 of 10	80%	3	13	9	27	1	8	0	1