CS 731 : Software Testing

PROJECT REPORT

Topic : MUTATION TESTING

DRISHTI GUPTA | MT2023099
AYUSHI PRASAD | MT2023145

Introduction to Mutation Testing

What is Mutation Testing?

Mutation testing is a software testing technique used to evaluate the quality and effectiveness of test cases. It involves modifying the source code in small ways, known as "mutants," to simulate potential errors or bugs. The primary goal is to determine whether the existing test cases can detect these changes, thereby ensuring they are robust enough to catch real-world issues.

How Mutation Testing Works

1. Mutation Generation: Small changes, such as replacing an operator (`+` with `-`) or altering a condition (`==` with `!=`), are introduced into the code. These changes simulate typical coding mistakes.
2. Test Execution: The test suite is run against the mutated code.
3. Result Analysis:
    - If the test suite fails for a mutant, the mutant is considered "killed," indicating the tests are effective.
    - If the test suite passes, the mutant "survives," suggesting a gap in the tests.

Purpose of Mutation Testing

Traditional code coverage metrics like line or branch coverage indicate how much of the code is executed during tests but do not ensure the quality of the test cases. Mutation testing addresses this limitation by measuring how well the tests detect faults, making it an excellent indicator of test suite effectiveness.

Benefits of Mutation Testing

- Improved Test Coverage: Highlights weak or missing test cases, encouraging comprehensive coverage.

- Fault Detection: Simulates real-world bugs to ensure the system behaves correctly under potential error conditions.
- Confidence in Code Quality: Enhances confidence in the reliability of the software by ensuring robust testing practices.

Challenges of Mutation Testing

- Performance Overhead: Running tests against multiple mutants can be time-consuming.
- False Positives: Certain mutants may survive due to equivalent functionality, not because of inadequate tests.
- Complexity: Requires specialized tools and setup, especially for large projects.

Role of Mutation Testing in This Project

In this project, mutation testing is applied to the banking system to:

1. Evaluate the effectiveness of both unit and integration tests.
2. Identify untested edge cases or scenarios.
3. Ensure the reliability of critical banking operations, such as transactions and account management, by exposing potential weaknesses in the test suite.

By combining JUnit for unit testing, Mockito for mocking in integration tests, and PIT for mutation testing, the project demonstrates a comprehensive approach to testing and validating software quality.

In Mutation Testing in Python,we used Mutpy to perform the testing.Following are the operators in python:

- AOR
- LOR
- COR
- CBR
- CR
- VR
- SD
- LBR
- FCR
- ER
- UOI
- BCR

Unit Testing and Integration Testing

Unit Testing

Unit testing focuses on testing individual components or units of code, such as methods or classes, in isolation. The goal is to ensure that each unit performs as expected.

Characteristics of Unit Testing

- Isolated Testing: Tests only one class or method at a time, without dependencies on other parts of the system.
- Early Detection: Helps identify bugs early in the development lifecycle.
- Automated Frameworks: Frameworks like JUnit in Java enable developers to write automated tests for each unit efficiently.

Advantages of Unit Testing

1. Fault Localization: Errors are easier to identify because tests are isolated to specific methods or classes.
2. Improved Code Quality: Encourages developers to write modular and testable code.
3. Regression Safety: Helps catch errors introduced during code changes.

Integration Testing

Integration testing ensures that multiple components or units work together as expected. It focuses on testing the interaction between modules, APIs, or services in the system.

Characteristics of Integration Testing

- End-to-End Validation: Verifies workflows across multiple units or subsystems.
- Dependency Management: Includes handling real or mocked external dependencies (e.g., databases, services).
- Realistic Scenarios: Tests real-world use cases, such as fund transfers between accounts.

Advantages of Integration Testing

1. Identifies Interface Issues: Catches bugs that arise due to mismatched or poorly integrated components.
2. Ensures Cohesion: Validates the proper functioning of interconnected modules.

3. Increases Reliability: Reduces risk in system-wide failures.

TOOLS USED

PIT (Pitest):
PIT is a mutation testing tool for Java that evaluates the effectiveness of test cases by introducing small changes (mutations) in the code. It tests whether existing test cases can detect and fail on these mutations. If a mutation is detected by a test (killed), it signifies strong test coverage. If it survives, it indicates areas of weak testing. PIT integrates seamlessly with build tools like Maven and Gradle and supports frameworks like JUnit and TestNG, making it highly efficient for Java projects.

MUTPY:
MUTPY is a mutation testing tool designed for Python projects. It mutates Python source code by altering operators, conditions, or functions and runs the test suite to verify its robustness. Similar to PIT, it categorizes mutations as killed or survived based on test performance. MUTPY works with popular Python testing frameworks like `unittest` and `pytest`, enabling developers to assess and improve test quality.

Comparison:
Both tools serve the same purpose of enhancing test quality but are designed for different languages. While PIT is highly optimized for Java projects and integrates with build tools, MUTPY provides similar capabilities for Python with standalone execution.

---

1. **BANKING MANAGEMENT SYSTEM (JAVA)**

   Lines of code

   Main.java : 83

   Bank.java : 40

   BankAccount.java : 68

Transaction.java : 36

BankingSystemTest.java : 157

IntegrationTest.java : 92

Total : 476 lines

REPORT

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
4	44%	53/120	23%	16/70	53%	16/30

Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
org.example	4	44%	53/120	23%	16/70	53%	16/30

Report generated by PIT 1.11.5

Enhanced functionality available at arcmutate.com

Pit Test Coverage Report

Package Summary

org.example

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
4	44%	53/120	23%	16/70	53%	16/30

Breakdown by Class

Name	Line Coverage		Mutation Coverage		Test Strength	
Bank.java	94%	17/18	22%	2/9	25%	2/8
BankAccount.java	96%	27/28	58%	11/19	61%	11/18
Main.java	0%	0/65	0%	0/38	0%	0/0
Transaction.java	100%	9/9	75%	3/4	75%	3/4

Report generated by PIT 1.11.5

Bank.java

Mutations

14 1. negated conditional → KILLED
16 1. removed call to java/io/PrintStream::println → SURVIVED
18 1. removed call to java/io/PrintStream::println → NO_COVERAGE
23 1. replaced return value with null for org/example/Bank::getAccount → KILLED
27 1. negated conditional → SURVIVED
28 1. removed call to java/io/PrintStream::println → SURVIVED
30 1. removed call to java/io/PrintStream::println → SURVIVED
35 1. removed call to java/io/PrintStream::println → SURVIVED
37 1. removed call to java/io/PrintStream::println → SURVIVED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankTests]/[method:testAddAccount()] (0 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankAccountTests]/[method:testWithdrawExceedingBalance()] (1 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:MainApplicationTests]/[method:testRemoveAndCheckAccount()] (16 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testDeposit()] (0 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankAccountTests]/[method:testWithdrawValidAmount()] (0 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankAccountTests]/[method:testDepositNegativeAmount()] (1 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankAccountTests]/[method:testDepositValidAmount()] (0 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testRemoveAccount()] (0 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testWithdrawMoreThanBalance()] (0 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testPrintAllAccounts()] (3 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:MainApplicationTests]/[method:testPrintingAccountDetails()] (20 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankTests]/[method:testPrintAllAccounts()] (0 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankAccountTests]/[method:testWithdrawNegativeAmount()] (1 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testAddAccount()] (0 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:MainApplicationTests]/[method:testAddingMultipleAccounts()] (2 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testWithdraw()] (1 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:TransactionTests]/[method:testTransactionCreation()] (2 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankTests]/[method:testRemoveAccountExisting()] (1 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankAccountTests]/[method:testTransactionHistory()] (0 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testPrintTransactions()] (1 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankTests]/[method:testRemoveAccountNonExisting()] (1 ms)

Transaction.java

Mutations

17	1. replaced return value with "" for org/example/Transaction::getType → KILLED
21	1. replaced double return with 0.0d for org/example/Transaction::getAmount → KILLED
25	1. replaced return value with null for org/example/Transaction::getTimestamp → KILLED
30	1. replaced return value with "" for org/example/Transaction::toString → SURVIVED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:MainApplicationTests]/[method:testPrintingAccountDetails()] (20 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:TransactionTests]/[method:testTransactionCreation()] (2 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankAccountTests]/[method:testTransactionHistory()] (0 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testPrintTransactions()] (1 ms)

BankAccount.java

Mutations

22	1. replaced return value with "" for org/example/BankAccount::getAccountNumber → KILLED
26	1. replaced return value with "" for org/example/BankAccount::getAccountHolderName → KILLED
30	1. replaced double return with 0.0d for org/example/BankAccount::getBalance → KILLED
34	1. replaced return value with "" for org/example/BankAccount::getAccountType → NO_COVERAGE
38	1. negated conditional → KILLED 2. changed conditional boundary → SURVIVED
39	1. Replaced double addition with subtraction → KILLED
42	1. removed call to java/io/PrintStream::println → SURVIVED
47	1. changed conditional boundary → SURVIVED 2. changed conditional boundary → SURVIVED 3. negated conditional → KILLED 4. negated conditional → KILLED
48	1. Replaced double subtraction with addition → KILLED
50	1. replaced boolean return with false for org/example/BankAccount::withdraw → KILLED
52	1. removed call to java/io/PrintStream::println → SURVIVED
53	1. replaced boolean return with true for org/example/BankAccount::withdraw → KILLED
58	1. removed call to java/io/PrintStream::println → SURVIVED
60	1. removed call to java/io/PrintStream::println → SURVIVED
66	1. replaced return value with Collections.emptyList for org/example/BankAccount::getTransactions → KILLED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankAccountTests]/[method:testWithdrawExceedingBalance()] (1 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankTests]/[method:testAddAccount()] (0 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:MainApplicationTests]/[method:testRemoveAndCheckAccount()] (16 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testDeposit()] (0 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankAccountTests]/[method:testWithdrawValidAmount()] (0 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankAccountTests]/[method:testDepositNegativeAmount()] (1 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankAccountTests]/[method:testDepositValidAmount()] (0 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testRemoveAccount()] (0 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testWithdrawMoreThanBalance()] (0 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testPrintAllAccounts()] (3 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:MainApplicationTests]/[method:testPrintingAccountDetails()] (20 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankAccountTests]/[method:testWithdrawNegativeAmount()] (1 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankTests]/[method:testPrintAllAccounts()] (0 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testAddAccount()] (0 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testWithdraw()] (1 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:MainApplicationTests]/[method:testAddingMultipleAccounts()] (2 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:TransactionTests]/[method:testTransactionCreation()] (2 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankTests]/[method:testRemoveAccountExisting()] (1 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankAccountTests]/[method:testTransactionHistory()] (0 ms)
- IntegrationTest.[engine:junit-jupiter]/[class:IntegrationTest]/[method:testPrintTransactions()] (1 ms)
- BankingSystemTest.[engine:junit-jupiter]/[class:BankingSystemTest]/[nested-class:BankTests]/[method:testRemoveAccountNonExisting()] (1 ms)

---

2. Gaussian Elimination Method (Python)

Lines of Code

gaussian_elimination.py: 73

test_gaussian_elimination.py: 56

REPORT

```
   - [#  40] ROR jacobi_iteration: [0.00622 s] killed by test_non_strictly_diagonally_dominant (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  41] ROR jacobi_iteration: [0.00861 s] survived
[*] Mutation score [1.78697 s]: 92.7%
   - all: 41
   - killed: 38 (92.7%)
   - survived: 3 (7.3%)
   - incompetent: 0 (0.0%)
   - timeout: 0 (0.0%)
drishti@drishti-HP-Laptop-14s-cr2xxx:~/testing$ mut.py --target=gaussian_elimination.py --unit-test=test_gaussian_elimination.py
[*] Start mutation process:
   - targets: gaussian_elimination.py
   - tests: test_gaussian_elimination.py
[*] 7 tests passed:
   - test_gaussian_elimination [0.00181 s]
[*] Start mutants generation and execution:
   - [#   1] AOR gaussian_elimination: [0.01397 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#   2] AOR gaussian_elimination: [0.00526 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#   3] AOR gaussian_elimination: [0.00741 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#   4] AOR gaussian_elimination: [0.00566 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#   5] AOR gaussian_elimination: [0.00670 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#   6] AOR gaussian_elimination: [0.00414 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#   7] AOR gaussian_elimination: [0.00594 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#   8] AOR gaussian_elimination: [0.00597 s] killed by test_gaussian_elimination_negative (test_gaussian_elimination.TestGaussianElimination)
   - [#   9] AOR gaussian_elimination: [0.00753 s] killed by test_gaussian_elimination_negative (test_gaussian_elimination.TestGaussianElimination)
   - [#  10] AOR gaussian_elimination: gaussian_elimination:58: RuntimeWarning: invalid value encountered in divide
gaussian_elimination:58: RuntimeWarning: divide by zero encountered in divide
[0.00626 s] killed by test_gaussian_elimination_negative (test_gaussian_elimination.TestGaussianElimination)
   - [#  11] AOR gaussian_elimination: gaussian_elimination:58: RuntimeWarning: invalid value encountered in floor_divide
[0.00742 s] killed by test_gaussian_elimination_negative (test_gaussian_elimination.TestGaussianElimination)
   - [#  12] AOR gaussian_elimination: gaussian_elimination:58: RuntimeWarning: divide by zero encountered in power
[0.00543 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#  13] ASR gaussian_elimination: [0.00614 s] killed by test_gaussian_elimination_negative (test_gaussian_elimination.TestGaussianElimination)
   - [#  14] COI gaussian_elimination: [0.00467 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#  15] COI gaussian_elimination: [0.00496 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#  16] COI gaussian_elimination: [0.00675 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#  17] COI gaussian_elimination: [0.00558 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#  18] COI gaussian_elimination: [0.00473 s] survived
   - [#  19] LCR gaussian_elimination: [0.00464 s] survived
   - [#  20] ROR gaussian_elimination: [0.00471 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#  21] ROR gaussian_elimination: [0.00530 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#  22] ROR gaussian_elimination: [0.00681 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#  23] ROR gaussian_elimination: [0.00539 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#  24] ROR gaussian_elimination: [0.00585 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#  25] ROR gaussian_elimination: [0.00526 s] survived
   - [#  26] SIR gaussian_elimination: [0.00570 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#  27] SIR gaussian_elimination: [0.00493 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#  28] SIR gaussian_elimination: [0.00517 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
   - [#  29] SIR gaussian_elimination: [0.00586 s] killed by test_gaussian_elimination_2x2 (test_gaussian_elimination.TestGaussianElimination)
[*] Mutation score [0.64717 s]: 89.7%
   - all: 29
   - killed: 26 (89.7%)
   - survived: 3 (10.3%)
   - incompetent: 0 (0.0%)
   - timeout: 0 (0.0%)
```

3. Jacobi Iteration Method (Python)

Lines of Code

jacobi_iteration.py: 205

test_jacobi_iteration.py: 68

REPORT

```
[*] Start mutation process:
   - targets: jacobi_iteration.py
   - tests: test_jacobi_iteration.py
[*] 8 tests passed:
   - test_jacobi_iteration [0.07339 s]
[*] Start mutants generation and execution:
   - [#   1] AOD jacobi_iteration: [0.07594 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#   2] AOD jacobi_iteration: [0.01573 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#   3] AOD jacobi_iteration: [0.01349 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#   4] AOD jacobi_iteration: [0.01496 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#   5] AOR jacobi_iteration: [0.00738 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#   6] AOR jacobi_iteration: [0.00378 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#   7] AOR jacobi_iteration: [0.00394 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#   8] AOR jacobi_iteration: [0.00471 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#   9] AOR jacobi_iteration: [0.00426 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  10] AOR jacobi_iteration: [0.01009 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  11] AOR jacobi_iteration: [0.00557 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  12] AOR jacobi_iteration: [0.00552 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  13] AOR jacobi_iteration: [0.00666 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  14] AOR jacobi_iteration: [0.00613 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  15] AOR jacobi_iteration: [0.00559 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  16] AOR jacobi_iteration: jacobi_iteration:161: RuntimeWarning: invalid value encountered in power
[0.00605 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  17] AOR jacobi_iteration: [0.01120 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  18] AOR jacobi_iteration: [0.00841 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  19] AOR jacobi_iteration: [0.00616 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  20] AOR jacobi_iteration: [0.00954 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  21] ASR jacobi_iteration: [0.01272 s] killed by test_non_strictly_diagonally_dominant (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  22] BCR jacobi_iteration: [0.00700 s] survived
   - [#  23] COI jacobi_iteration: [0.00377 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  24] COI jacobi_iteration: [0.00390 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  25] COI jacobi_iteration: [0.00518 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  26] COI jacobi_iteration: [0.00463 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  27] COI jacobi_iteration: [0.00559 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  28] COI jacobi_iteration: [0.00383 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  29] COI jacobi_iteration: [0.00368 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  30] COI jacobi_iteration: [0.01751 s] survived
   - [#  31] LOD jacobi_iteration: [0.00364 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  32] ROR jacobi_iteration: [0.00357 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  33] ROR jacobi_iteration: [0.00485 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  34] ROR jacobi_iteration: [0.00608 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  35] ROR jacobi_iteration: [0.00420 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  36] ROR jacobi_iteration: [0.00535 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  37] ROR jacobi_iteration: [0.00767 s] killed by test_zero_iterations (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  38] ROR jacobi_iteration: [0.00397 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  39] ROR jacobi_iteration: [0.00488 s] killed by test_basic_case (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  40] ROR jacobi_iteration: [0.00622 s] killed by test_non_strictly_diagonally_dominant (test_jacobi_iteration.TestJacobiIterationMethod)
   - [#  41] ROR jacobi_iteration: [0.00861 s] survived
[*] Mutation score [1.78697 s]: 92.7%
   - all: 41
   - killed: 38 (92.7%)
   - survived: 3 (7.3%)
   - incompetent: 0 (0.0%)
   - timeout: 0 (0.0%)
```

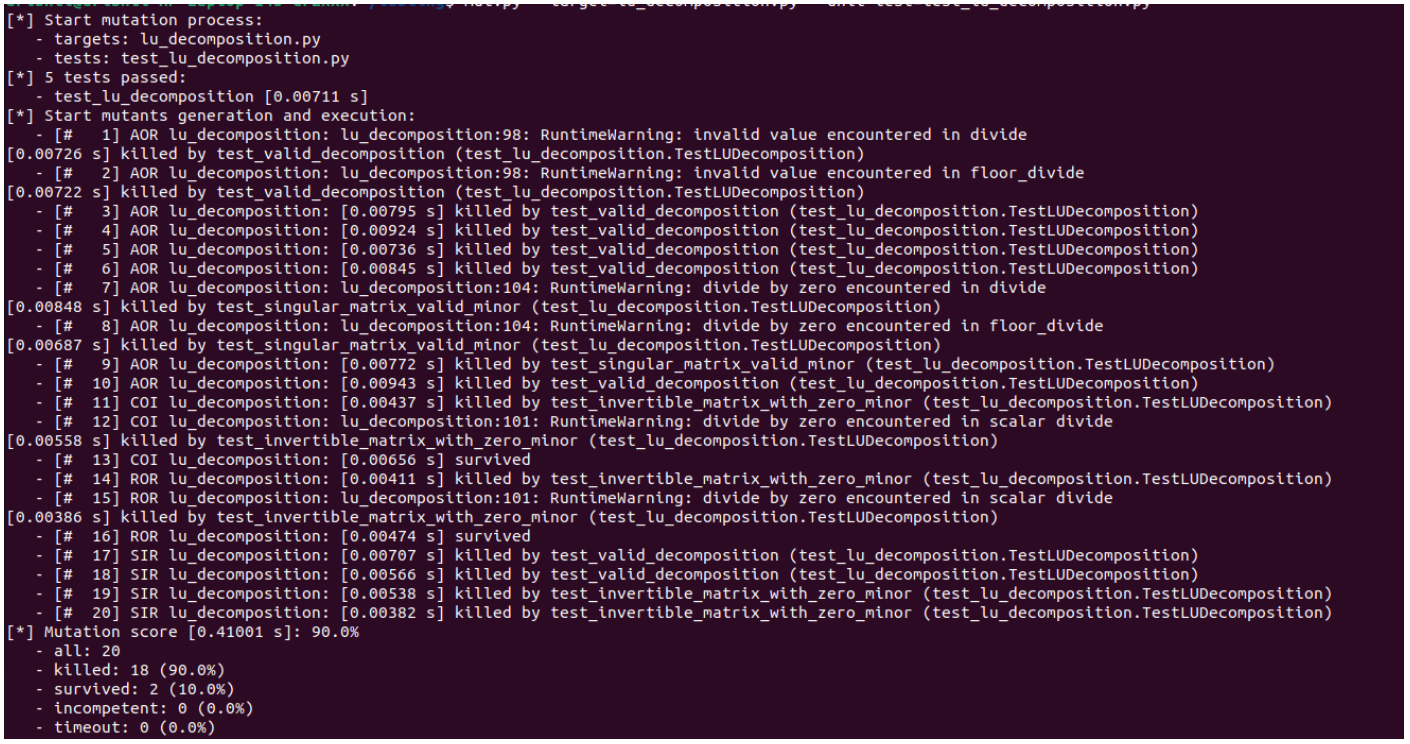
4. LU Decomposition (Python)

Lines of Code

lu_decomposition.py: 112

test_lu_decomposition.py: 59

REPORT



```
[*] Start mutation process:
   - targets: lu_decomposition.py
   - tests: test_lu_decomposition.py
[*] 5 tests passed:
   - test_lu_decomposition [0.00711 s]
[*] Start mutants generation and execution:
   - [#   1] AOR lu_decomposition: lu_decomposition:98: RuntimeWarning: invalid value encountered in divide
[0.00726 s] killed by test_valid_decomposition (test_lu_decomposition.TestLUDecomposition)
   - [#   2] AOR lu_decomposition: lu_decomposition:98: RuntimeWarning: invalid value encountered in floor_divide
[0.00722 s] killed by test_valid_decomposition (test_lu_decomposition.TestLUDecomposition)
   - [#   3] AOR lu_decomposition: [0.00795 s] killed by test_valid_decomposition (test_lu_decomposition.TestLUDecomposition)
   - [#   4] AOR lu_decomposition: [0.00924 s] killed by test_valid_decomposition (test_lu_decomposition.TestLUDecomposition)
   - [#   5] AOR lu_decomposition: [0.00736 s] killed by test_valid_decomposition (test_lu_decomposition.TestLUDecomposition)
   - [#   6] AOR lu_decomposition: [0.00845 s] killed by test_valid_decomposition (test_lu_decomposition.TestLUDecomposition)
   - [#   7] AOR lu_decomposition: lu_decomposition:104: RuntimeWarning: divide by zero encountered in divide
[0.00848 s] killed by test_singular_matrix_valid_minor (test_lu_decomposition.TestLUDecomposition)
   - [#   8] AOR lu_decomposition: lu_decomposition:104: RuntimeWarning: divide by zero encountered in floor_divide
[0.00687 s] killed by test_singular_matrix_valid_minor (test_lu_decomposition.TestLUDecomposition)
   - [#   9] AOR lu_decomposition: [0.00772 s] killed by test_singular_matrix_valid_minor (test_lu_decomposition.TestLUDecomposition)
   - [#  10] AOR lu_decomposition: [0.00943 s] killed by test_valid_decomposition (test_lu_decomposition.TestLUDecomposition)
   - [#  11] COI lu_decomposition: [0.00437 s] killed by test_invertible_matrix_with_zero_minor (test_lu_decomposition.TestLUDecomposition)
   - [#  12] COI lu_decomposition: lu_decomposition:101: RuntimeWarning: divide by zero encountered in scalar divide
[0.00558 s] killed by test_invertible_matrix_with_zero_minor (test_lu_decomposition.TestLUDecomposition)
   - [#  13] COI lu_decomposition: [0.00656 s] survived
   - [#  14] ROR lu_decomposition: [0.00411 s] killed by test_invertible_matrix_with_zero_minor (test_lu_decomposition.TestLUDecomposition)
   - [#  15] ROR lu_decomposition: lu_decomposition:101: RuntimeWarning: divide by zero encountered in scalar divide
[0.00386 s] killed by test_invertible_matrix_with_zero_minor (test_lu_decomposition.TestLUDecomposition)
   - [#  16] ROR lu_decomposition: [0.00474 s] survived
   - [#  17] SIR lu_decomposition: [0.00707 s] killed by test_valid_decomposition (test_lu_decomposition.TestLUDecomposition)
   - [#  18] SIR lu_decomposition: [0.00566 s] killed by test_valid_decomposition (test_lu_decomposition.TestLUDecomposition)
   - [#  19] SIR lu_decomposition: [0.00538 s] killed by test_invertible_matrix_with_zero_minor (test_lu_decomposition.TestLUDecomposition)
   - [#  20] SIR lu_decomposition: [0.00382 s] killed by test_invertible_matrix_with_zero_minor (test_lu_decomposition.TestLUDecomposition)
[*] Mutation score [0.41001 s]: 90.0%
   - all: 20
   - killed: 18 (90.0%)
   - survived: 2 (10.0%)
   - incompetent: 0 (0.0%)
   - timeout: 0 (0.0%)
```

CONTRIBUTION OF TEAM MEMBERS

Ayushi Prasad : Java Testing

Drishti Gupta : Python Testing
