International Institute of Information Technology Bangalore

Mutation Testing on Address Book Management System in Java

-

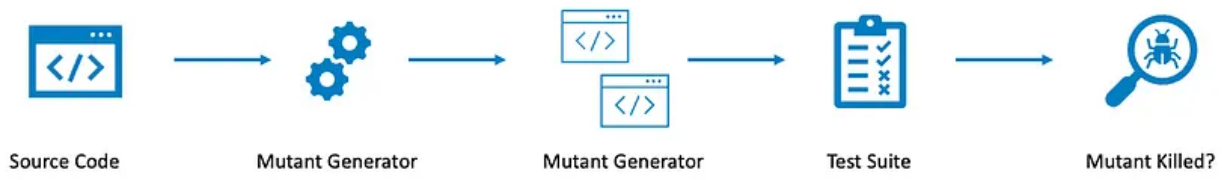
Team Members

Sanket Patil MT2023051
Anuja Gode MT2023039

GitHub Repository - MutationTesting

Introduction to Mutation Testing

A mutation is a small change in a program. Such small changes are intended to model low level defects that arise in the process of coding systems.
Mutation testing is a structural testing method aimed at assessing/improving the adequacy of test suites, and estimating the number of faults present in systems under test.

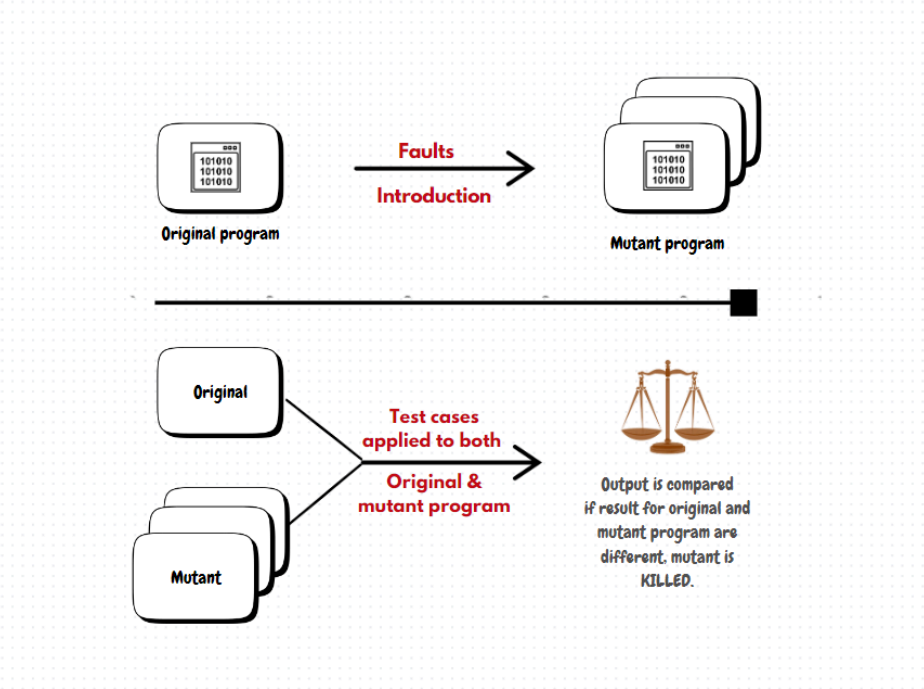


Source - https://medium.com/@nandan.abhi10/beyond-unit-testing-how-mutation-testing-helps-to-improve-code-quality-and-reliability-c8dc49737492

How mutation testing works:

Mutation testing basically involves introducing minor errors, called **mutants**, into the program code to check if the tests can detect these changes.

The ultimate goal is for robust tests to fail when they find these intentional changes. If the tests are unable to detect the mutants, it suggests they are ineffective at identifying problems.



Source - https://www.stackspot.com/en/blog/mutation-testing

The mutation testing process involves the following steps:

- Introducing faults: Changes are made to the program's source code to create mutants, such as replacing a plus sign with a minus sign in a mathematical expression.

- Test application: Test cases are applied to both the original code and the mutated program. If test results for the original code and the mutants differ, the mutant has been detected and eliminated, demonstrating the test's effectiveness.

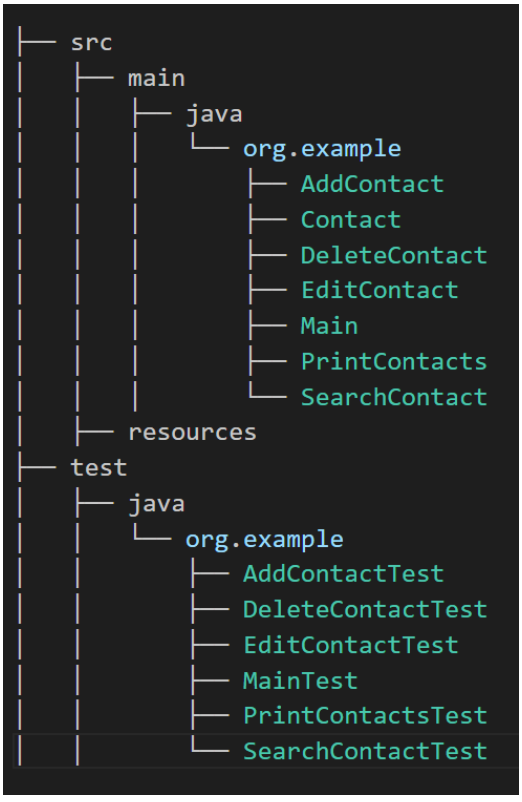
- Analysis of results: If the tests fail to detect the change and the results for the original code and the mutant are the same, the tests lack sensitivity to identify errors, indicating a need to improve the test cases.

Tools Used:

- **PITest -** PIT is a state of the art mutation testing system, providing gold standard test coverage for Java and the jvm. It's fast, scalable and integrates with modern test and build tooling.
  PIT is -
  - fast - can analyse in minutes what would take earlier systems days
  - easy to use - works with ant, maven, gradle and others
  - actively developed
  - actively supported

- **JUnit -** It is a popular testing framework for Java. It allows developers to write and run unit tests to check if individual parts (like methods or classes) of their code work as expected. In simple terms, JUnit helps in testing code to catch bugs early by automating the testing process.

Project Structure:



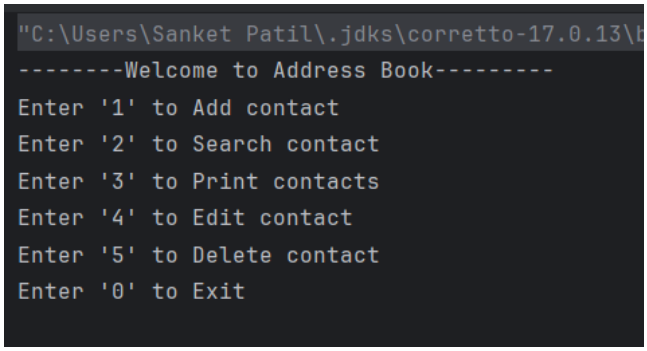
The project directory follows a standard Maven structure. The src/main/java folder contains the application's main Java classes, The test/java folder contains tests for the corresponding classes in the main directory.

Address Book Management System:

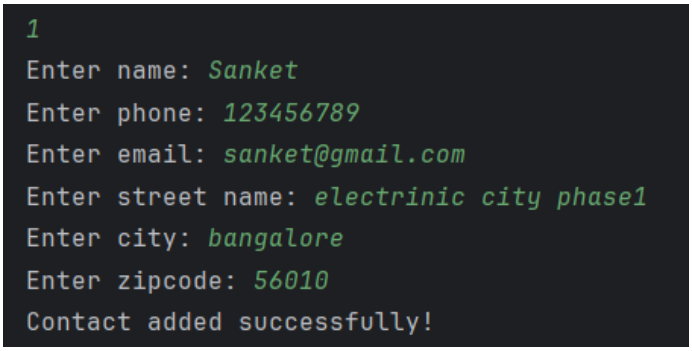
A simple Java-based console application to manage personal contacts. This application allows users to add, search, view, edit, and delete contact information, including details like name, phone number, email, street name, city and zip code.

Features

1. Add Contact
    - Add a new contact with the following details:
        - Name (unique and non-empty)
        - Phone number (digits only, minimum 7 characters)
        - Email address (valid format)
        - Address details: Street name, city, and zip code(5 digits)
    - Prevents duplicate names.
2. Search Contact
    - Search for a contact by name.
    - Optionally refine search using street name, city, or zipcode.
3. View Contacts
    - Display all saved contacts.
4. Edit Contact
    - Modify existing contact details.
5. Delete Contact
    - Remove a contact by name.
6. Error Handling
    - Provides clear error messages for invalid or missing inputs.



Console Menu for Address Book Application



Contact Added in the System

Printed all contacts in the system

PIT Mutation Testing Plugin

```
<build>
        <plugins>
            <plugin>
                <groupId>org.pitest</groupId>
                <artifactId>pitest-maven</artifactId>
                <version>1.14.0</version>
                <executions>
                    <execution>
                        <id>mutation-test</id>
                        <phase>verify</phase>
                        <goals>
                            <goal>mutationCoverage</goal>
                        </goals>
                    </execution>
                </executions>
                <configuration>
                    <targetClasses>
                        <param>org.example.*</param>
                    </targetClasses>
                    <targetTests>
                        <param>org.example.*Test</param>
                    </targetTests>
                    <mutators>
                        <!-- mutators to generate mutations -->
                        <mutator>CONDITIONALS_BOUNDARY</mutator>
                        <mutator>EMPTY_RETURNS</mutator>
                        <mutator>FALSE_RETURNS</mutator>
                        <mutator>INCREMENTS</mutator>
                        <mutator>INVERT_NEGS</mutator>
                        <mutator>MATH</mutator>
                        <mutator>NEGATE_CONDITIONALS</mutator>
                        <mutator>PRIMITIVE_RETURNS</mutator>
                        <mutator>TRUE_RETURNS</mutator>
                        <mutator>VOID_METHOD_CALLS</mutator>
                        <mutator>NON_VOID_METHOD_CALLS</mutator>
                        <mutator>EXPERIMENTAL_ARGUMENT_PROPAGATION</mutator>
                        <mutator>EXPERIMENTAL_NAKED_RECEIVER</mutator>
                        <mutator>REMOVE_CONDITIONALS</mutator>
                        <mutator>NULL_RETURNS</mutator>
                    </mutators>
                </configuration>
            </plugin>
        </plugins>
    </build>
```

Dependencies

```xml
<dependencies>
        <!-- JUnit 5 dependency -->
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter</artifactId>
            <version>5.10.0</version>
            <scope>test</scope>
        </dependency>
        <!-- PIT JUnit 5 Plugin -->
        <dependency>
            <groupId>org.pitest</groupId>
            <artifactId>pitest-junit5-plugin</artifactId>
            <version>1.2.0</version>
        </dependency>
        <dependency>
            <groupId>org.mockito</groupId>
            <artifactId>mockito-core</artifactId>
            <version>3.8.0</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
```

Mutation Operators Used :

- Unit-Level Operators:
    1. CONDITIONALS_BOUNDARY -

       Modifies boundary conditions (e.g., < to <=)
    2. EMPTY_RETURNS -

       Modifies return statements to return empty/default values
    3. FALSE_RETURNS - Changes return values to false
    4. INCREMENTS - Modifies increment operators
    5. INVERT_NEGS - Inverts negative numbers
    6. MATH - Changes mathematical operators
    7. NEGATE_CONDITIONALS - Negates conditional statements
    8. PRIMITIVE_RETURNS - Modifies primitive return values
    9. TRUE_RETURNS - Changes return values to true
    10. VOID_METHOD_CALLS - Removes void method calls
    11. NULL_RETURNS - Changes return values to null

- Integration-Level Operators:
    1. NON_VOID_METHOD_CALLS -

       Affects method interactions and return value handling
    2. EXPERIMENTAL_ARGUMENT_PROPAGATION -

       Affects method parameter passing and interactions
    3. EXPERIMENTAL_NAKED_RECEIVER -

       Impacts object interactions and method calls
    4. REMOVE_CONDITIONALS -

       Can affect program flow and component interactions

JUnit Test Results Summary:

```
mvn test
```

```
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 45, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  4.291 s
[INFO] Finished at: 2024-11-25T18:42:36+05:30
[INFO] ------------------------------------------------------------------------
```

Run the PIT Command -

```
mvn org.pitest:pitest-maven:mutationCoverage
```

```
================================================================================
- Statistics
================================================================================
>> Line Coverage (for mutated classes only): 203/215 (94%)
>> Generated 352 mutations Killed 305 (87%)
>> Mutations with no coverage 2. Test strength 87%
>> Ran 830 tests (2.36 tests per mutation)
Enhanced functionality available at https://www.arcmutate.com/
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  45.952 s
[INFO] Finished at: 2024-11-25T19:04:20+05:30
[INFO] ------------------------------------------------------------------------
PS C:\Users\Sanket Patil\Documents\IIITB\SEM3\ST\Project>
```

After the execution, the mutation report will be generated in the target/pit-reports directory.
Open the index.html file in the browser to view the detailed PIT mutation coverage report.

Pit Test Coverage Report

Package Summary

org.example

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
7	94%	203/215	87%	305/352	87%	305/350

Breakdown by Class

Name	Line Coverage		Mutation Coverage		Test Strength	
AddContact.java	98%	39/40	91%	68/75	91%	68/75
Contact.java	94%	30/32	100%	7/7	100%	7/7
DeleteContact.java	97%	29/30	73%	55/75	73%	55/75
EditContact.java	94%	45/48	94%	85/90	96%	85/89
Main.java	90%	26/29	95%	18/19	100%	18/18
PrintContacts.java	89%	8/9	100%	16/16	100%	16/16
SearchContact.java	96%	26/27	80%	56/70	80%	56/70

Report generated by PIT 1.14.0

SearchContact.java

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- EXPERIMENTAL_ARGUMENT_PROPAGATION
- EXPERIMENTAL_NAKED_RECEIVER
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NON_VOID_METHOD_CALLS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- REMOVE_CONDITIONALS_EQUAL_ELSE
- REMOVE_CONDITIONALS_EQUAL_IF
- REMOVE_CONDITIONALS_ORDER_ELSE
- REMOVE_CONDITIONALS_ORDER_IF
- TRUE_RETURNS
- VOID_METHOD_CALLS

Main.java

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- EXPERIMENTAL_ARGUMENT_PROPAGATION
- EXPERIMENTAL_NAKED_RECEIVER
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NON_VOID_METHOD_CALLS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- REMOVE_CONDITIONALS_EQUAL_ELSE
- REMOVE_CONDITIONALS_EQUAL_IF
- REMOVE_CONDITIONALS_ORDER_ELSE
- REMOVE_CONDITIONALS_ORDER_IF
- TRUE_RETURNS
- VOID_METHOD_CALLS

Mutations -

30	2. removed call to java/lang/String::trim → KILLED 3. removed call to java/util/Scanner::nextLine → KILLED
32	1. removed call to java/io/PrintStream::print → SURVIVED
33	1. removed call to java/lang/String::trim → KILLED 2. removed call to java/util/Scanner::nextLine → KILLED 3. replaced call to java/lang/String::trim with receiver → SURVIVED
36	1. removed call to java/util/List::stream → KILLED
37	1. replaced call to java/util/stream/Stream::filter with receiver → KILLED 2. removed call to org/example/Contact::getName → KILLED 3. removed conditional - replaced equality check with false → KILLED 4. removed conditional - replaced equality check with true → KILLED 5. negated conditional → KILLED 6. removed call to java/lang/String::equalsIgnoreCase → KILLED 7. removed call to java/util/stream/Stream::filter → KILLED
38	1. negated conditional → KILLED 2. negated conditional → KILLED 3. removed call to java/lang/String::equalsIgnoreCase → KILLED 4. removed conditional - replaced equality check with true → KILLED 5. removed call to org/example/Contact::getStreetName → KILLED 6. removed conditional - replaced equality check with false → KILLED 7. removed conditional - replaced equality check with true → SURVIVED 8. removed call to java/lang/String::isEmpty → KILLED 9. removed conditional - replaced equality check with false → SURVIVED

SearchContact.java

References -

1. https://www.inf.ed.ac.uk/teaching/courses/st/2011-12/Resource-folder/09_mutation.pdf
2. https://www.stackspot.com/en/blog/mutation-testing
3. https://pitest.org/quickstart/mutators/
4. https://www.youtube.com/watch?v=DSv2vpvD-ds&t=2195s
5. https://www.youtube.com/watch?v=hVKDEXKLN2c&t=2713s
6. https://softengbook.org/articles/mutation-testing