Final Report – Mutation Testing: Task Management Application

# Objective

To evaluate the strength of the designed test suite by injecting faults (mutants) into the original code and determining whether the test cases can detect them. This helps ensure robustness and fault-tolerance in the software.

# Mutant Summary Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Location | Original Code | Mutated Code | Mutation Type | Killed? |
| 1 | markTaskAsComplete | task.setCompleted(true); | task.setCompleted(false); | Boolean Negation | ✅ Yes |
| 2 | deleteTask | tasks.remove(index); | //tasks.remove(index); | Statement Removal | ✅ Yes |
| 3 | addTask | tasks.add(task); | if (task != null) tasks.add(null); | Null Injection | ✅ Yes |
| 4 | getPriority | return priority; | return "Medium"; | Constant Replacement | ✅ Yes |
| 5 | editTask | tasks.set(index, updatedTask); | tasks.set(index, tasks.get(index)); | Assignment Skipped | ✅ Yes |
| 6 | main menu switch | case 5: | case 4: | Switch Case Swap | ✅ Yes |
| 7 | editTask condition | index >= 0 && index < tasks.size() | index > 0 && index < tasks.size() | Relational Operator Replacement | ✅ Yes |
| 8 | displayTasks | System.out.println(task); | System.out.println("Task"); | Output Mutation | ✅ Yes |
| 9 | toString Method | Full task info | Return only title | String Mutation | ✅ Yes |
| 10 | setCompleted | Calls setCompleted(true) | Skip method call | Method Call Skipped | ✅ Yes |

# Mutation Score

Formula:  
Mutation Score = (Killed Mutants / Total Mutants) × 100  
  
Total Mutants Introduced: 10  
Mutants Killed: 10  
Mutation Score: (10 / 10) × 100 = 100% ✅

# Simulation Example

To verify that the test suite detects mutants:  
- We manually changed `task.setCompleted(true)` to `task.setCompleted(false)` in `markTaskAsComplete()`.  
- Upon running the test `testMarkTaskAsComplete()`, the test failed, successfully killing the mutant.  
This confirmed the strength of the test case.

# Tools & Approach

- Manual mutation injection  
- JUnit 4 test suite implemented in BlueJ  
- Mutants tested one-by-one by modifying code and running test cases

# Conclusion

All 10 manually injected mutants were detected by the refined test suite, giving a 100% mutation score. This demonstrates that the test suite is comprehensive and effective in detecting logical faults, ensuring the reliability of the Task Management Application.