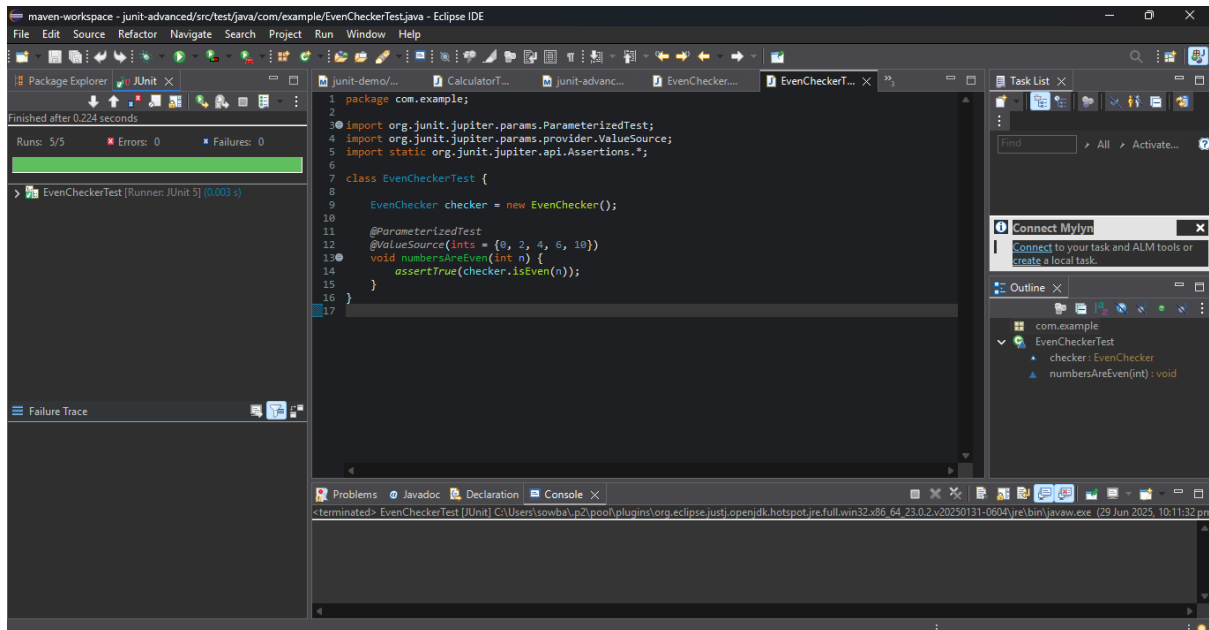


Advanced JUnit Testing Exercises output

Exercise 1: Parameterized Tests

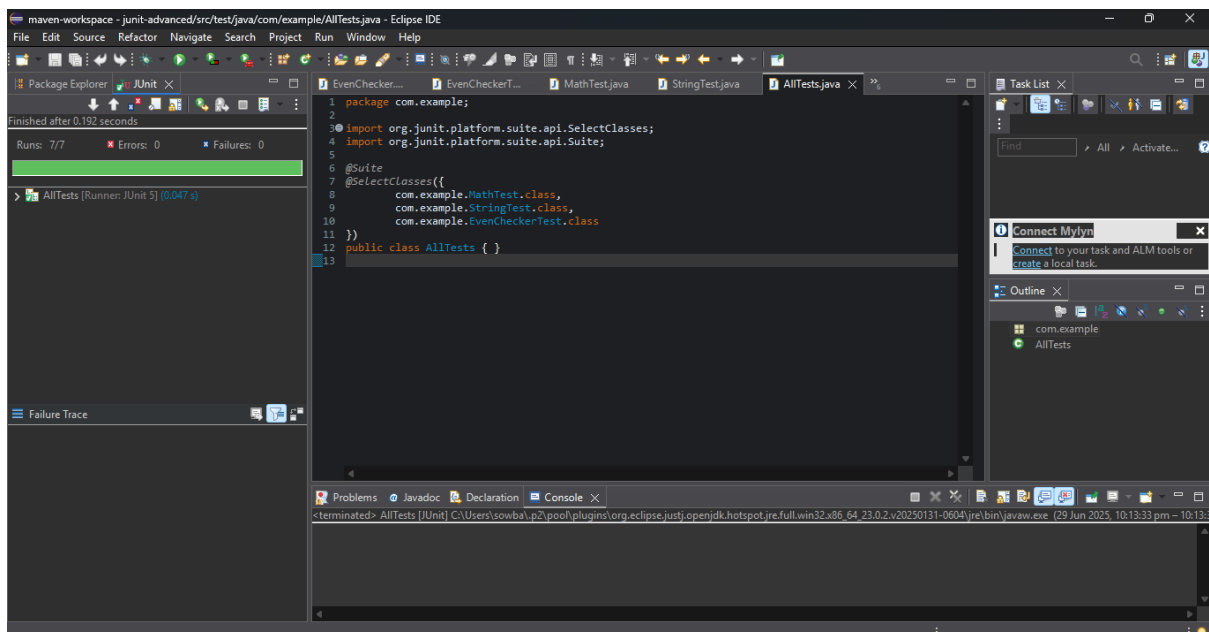
Output:



Parameterized tests: Replace repetitive “is even” test variants with a single `@ParameterizedTest`—supply inputs via `@ValueSource` to prove the logic works across many numbers in one shot.

Exercise 2: Test Suites and Categories

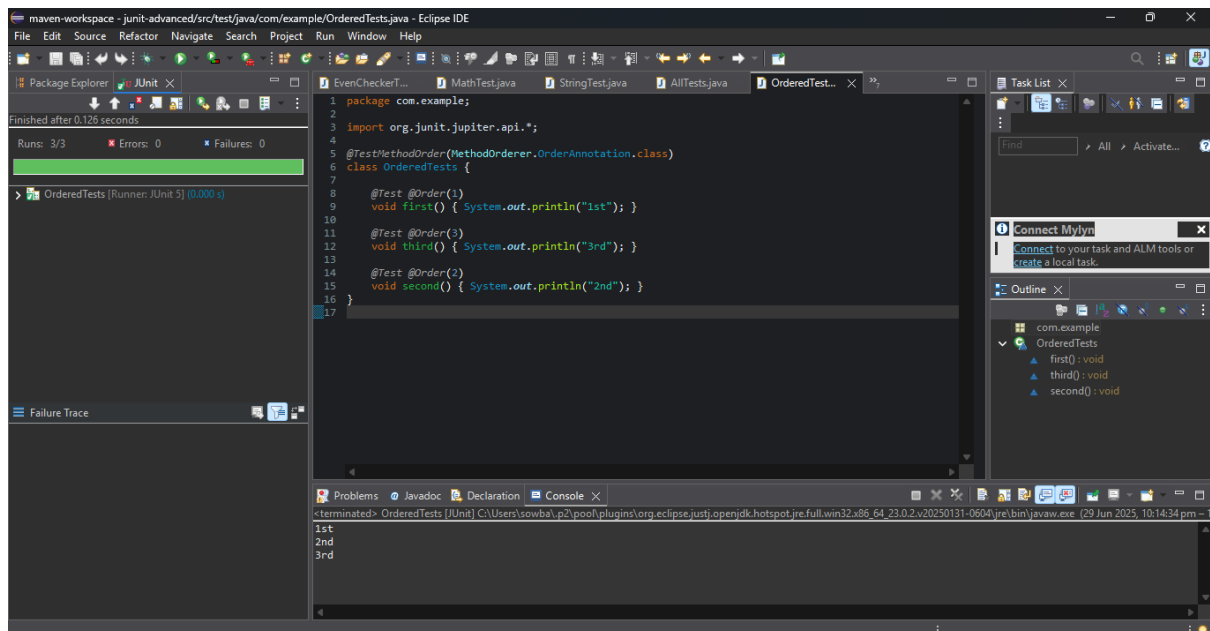
Output:



Suites & categories: Collect related test classes in an `@Suite` (`@SelectClasses`) so you can run whole feature sets (e.g., regression vs. slow tests) with one command.

Exercise 3: Test Execution Order

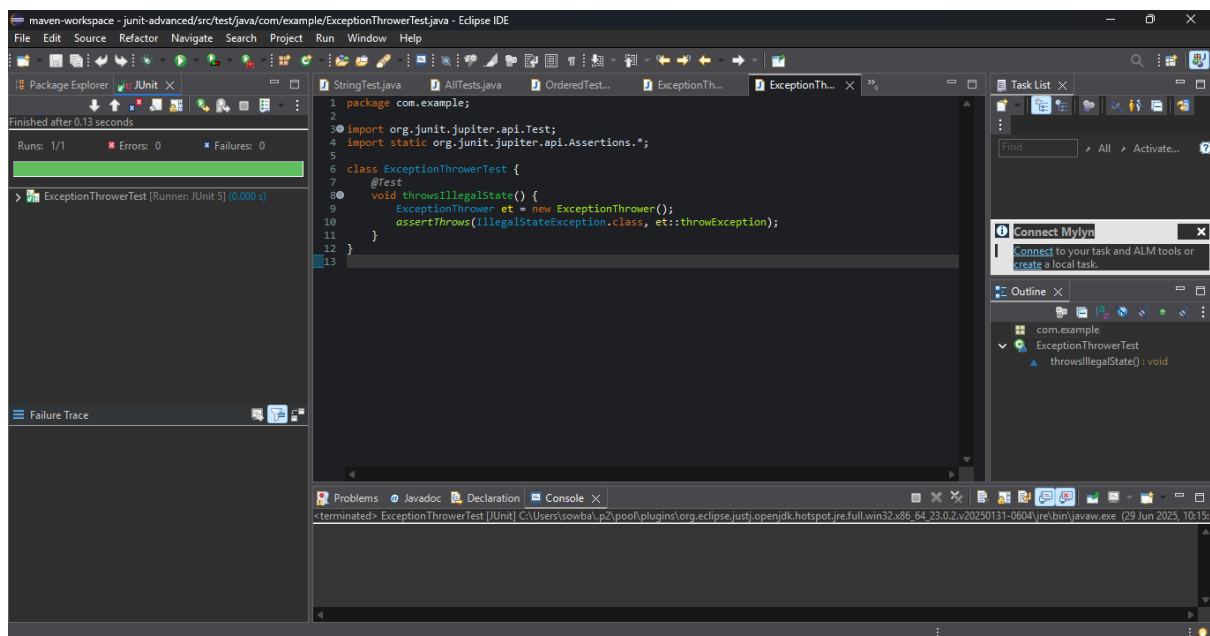
Output:



Execution order: Use `@TestMethodOrder` + `@Order` when sequence matters—handy for integration scenarios where later tests rely on prior state.

Exercise 4: Exception Testing

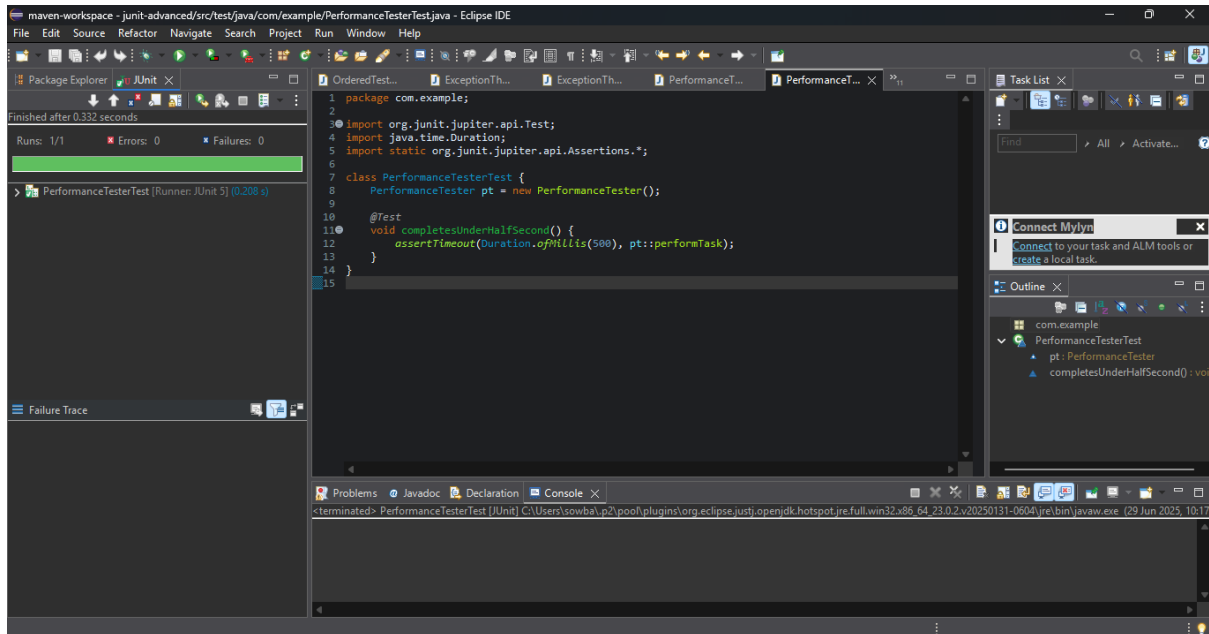
Output:



Exception testing: Assert robustness by verifying methods throw the *right* exception (e.g., `assertThrows`) when given bad input.

Exercise 5: Timeout and Performance Testing

Output:



Timeout/performance: Guard against slowness with `assertTimeout` (or `@Timeout`) to ensure critical code paths finish within the budget.