Test Report

Run ID: N/A • Generated: 2026-01-12 19:11:02 • Duration: 50ms
Plugin: v0.1.0 (65f94132e69b39c0e7c5dee79c065492ae3fad6c) [dirty]
Repo: v0.1.0 (65f94132e69b39c0e7c5dee79c065492ae3fad6c) [dirty]
LLM: ollama / llama3.2 (complete context, 4 annotated)

100.0%
Total Coverage

4	4	0	0
TOTAL TESTS	PASSED	FAILED	SKIPPED

0	0	0
XFAILED	XPASSED	ERRORS

PASSED `tests/test_calculator.py::test_add_basic` 0ms 🛡️ 1

AI ASSESSMENT

Scenario: This test verifies that the add function returns the correct result for basic addition operations.

Why Needed: This test prevents incorrect results due to faulty implementation of the add function, which could lead to errors in further calculations or user trust issues.

Key Assertions:

- The test checks if the add function correctly handles two positive numbers (e.g., 1 + 2 = 3).
- It ensures that the add function returns a correct result for zero inputs (e.g., 0 + 2 = 2).
- This test also verifies that the add function behaves as expected with negative numbers (e.g., -1 + 2 = 1).
- The test checks if the add function raises an error when given invalid input types, such as non-numeric values.

COVERAGE

example_pkg/calculator.py	1 lines (ranges: 4)

PASSED `tests/test_calculator.py::test_add_negative` 0ms 1

AI ASSESSMENT

Scenario: This test verifies that the addition function correctly handles negative numbers.

Why Needed: This test prevents bugs where incorrect results are produced when dealing with negative inputs, which could lead to errors in calculations or unexpected behavior in downstream applications.

Key Assertions:

- The function should return -2 for the input (-1, -1).
- The function should handle one negative number correctly.
- The function should not raise an error when both numbers are negative.
- The function's result should be consistent with the expected output of a simple arithmetic operation (a + b).

COVERAGE

example_pkg/calculator.py	1 lines (ranges: 4)

PASSED `tests/test_calculator.py::test_divide_success` 0ms 1

AI ASSESSMENT

Scenario: This test verifies that the division function correctly returns the expected result when both inputs are non-zero numbers.

Why Needed: This test prevents bugs caused by incorrect handling of division by zero, ensuring the calculator module provides accurate results in all scenarios.

Key Assertions:

- The divide function raises a ValueError with a meaningful error message when attempting to divide by zero.
- The divide function correctly returns the expected result for positive and negative numbers.
- The divide function handles decimal results accurately, as demonstrated by the successful assertion of 10 divided by 2 equaling 5.

COVERAGE

example_pkg/calculator.py	2 lines (ranges: 8, 10)

PASSED tests/test_calculator.py::test_divide_zero 0ms 1

AI ASSESSMENT

Scenario: This test verifies that the calculator's divide function raises a ValueError when attempting to divide by zero.

Why Needed: This test prevents potential bugs where division by zero is not handled, leading to unexpected errors or crashes in the application.

Key Assertions:

- The test checks if the divide function correctly raises a ValueError
- It verifies that the error message 'Cannot divide by zero' is raised as expected
- The test ensures that the function behaves as intended when encountering division by zero

COVERAGE

example_pkg/calculator.py	2 lines (ranges: 8-9)

Source Coverage

FILE	STMTS	MISS	COVER	%	COVERED LINES	MISSED LINES
example_pkg/calculator.py	6	0	6	100.0%	2, 4, 6, 8-10	-