

1. Why did Fibonacci fail the `testThrowsIllegalArgumentException` test? What did you have to do to fix it?

The error message is “Threw `IllegalArgumentException` for 0 but 0 is nonnegative”. This means that the function did not accept 0 as a valid argument when it should. The function should accept all nonnegative values, which includes 0.

To fix this, I first looked for where `IllegalArgumentException` is thrown. It is only thrown when `n`, the single argument for the function, is ≤ 0 . This means all negative values plus zero throws the error. Only all negative values should throw the error. So I changed the \leq to a $<$, and this removed the error.

2. Why did Fibonacci fail the `testBaseCase` test? What (if anything) did you have to do to fix it?

The statement that tested for this error is `assertEquals("getFibTerm(0)", 1, fib.getFibTerm(0))`.

This asserts that when 0 is the argument for a function call, the function should return 1. I did not have to do anything to fix this. It was no longer an error after doing the change for question 1, as `IllegalArgumentException` was thrown before the return statement.

3. Why did Fibonacci fail the `testInductiveCase` test? What (if anything) did you have to do to fix it?

The final return value for at least one function call is incorrect. The test cases correctly match Fibonacci numbers with their position in the Fibonacci sequence, so at least one function call (where the argument represents the position of the number in the sequence) did not return the correct Fibonacci number. This had to do with the correctness of the Fibonacci function's implementation. I saw that the “+” and “-” signs were mixed up in the statement containing the recurrence relation.

Original: `return getFibTerm(n + 1) - getFibTerm(n - 2)`

Correction: `return getFibTerm(n - 1) + getFibTerm(n - 2)`

Also, 1 is returned when $n \leq 2$, when it should be $n \leq 1$.

After implementing these 2 corrections, the function passed this test, and thus all errors were fixed.