Steven Thaw

sjthaw@me.com

Abstract

A program which calculates the Nth value of the Fibonacci Sequence both recursively and iteratively, and provides an efficiency index for the method used.

fibonacci sequencer project 3

CMIS242 Intermediate Programming

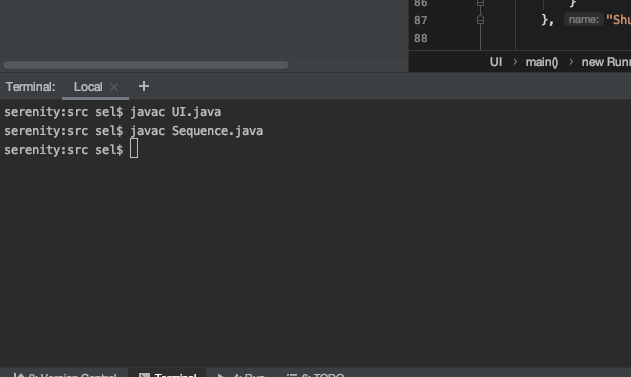


Figure Successful Compilation of Code

**Iterative Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected Output** | **Pass/Fail** |
| 1 | “Iterative” radio button selected, 2 | Enter N: 2  Result: 1  Efficiency: 1 | Pass |
| 2 | “Iterative” radio button selected, 20 | Enter N: 20  Result 6765  Efficiency: 19 | Pass |
| 3 | “Iterative” radio button selected, 14 | Enter N: 14  Result: 377  Efficiency: 13 | Pass |
| 4 | “Iterative” radio button selected, 8 | Enter N: 8  Result: 21  Efficiency: 7 | Pass |
| 5 | “Iterative” radio button selected, “Test” | A new dialog window pops up stating “Please enter an integer! | Pass |

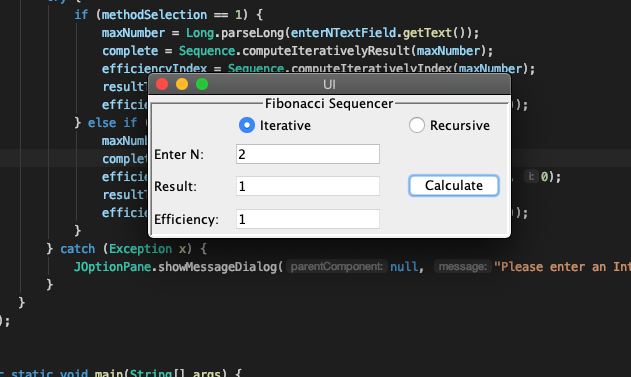


Figure Test Case 1

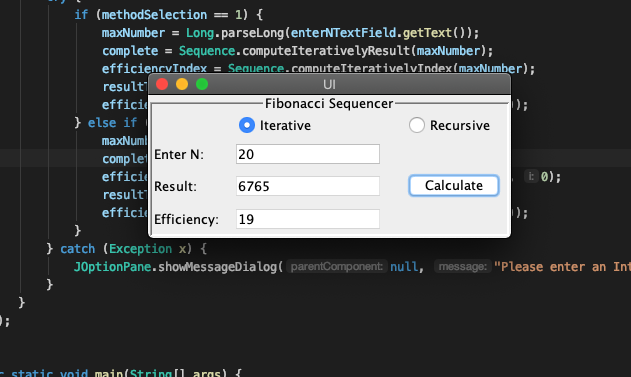


Figure Test Case 2

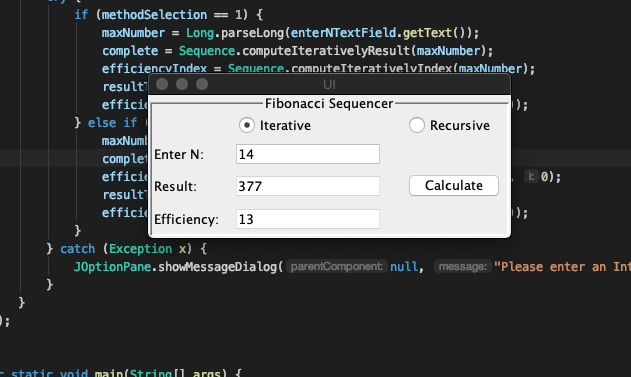


Figure Test Case 3

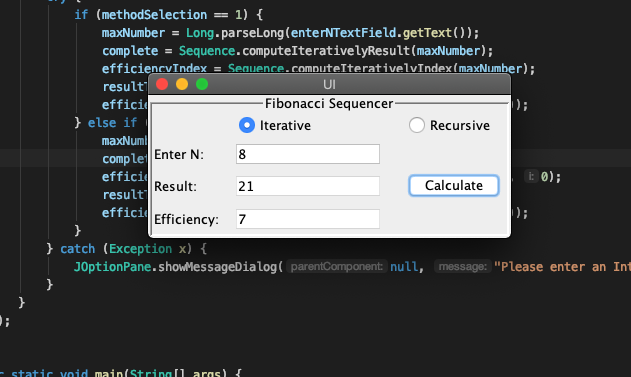


Figure Test Case 4

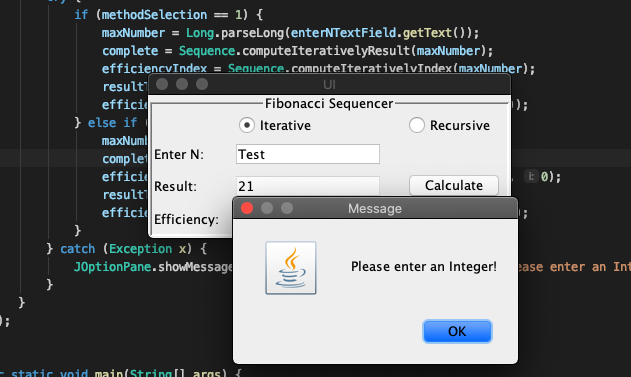


Figure Test Case 5

**Recursive Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected Output** | **Pass/Fail** |
| 6 | “Recursive” radio button selected, 2 | Enter N: 2  Result: 1  Efficiency: 2 | Pass |
| 7 | “Recursive” radio button selected, 20 | Enter N: 20  Result: 6765  Efficiency: 153000 | Pass |
| 8 | “Recursive” radio button selected, 14 | Enter N: 14  Result: 377  Efficiency: 5878 | Pass |
| 9 | “Recursive” radio button selected, 8 | Enter N: 8  Result: 21  Efficiency: 180 | Pass |
| 10 | “Recursive” radio button selected, “Test” | A new dialog window pops up stating “Please enter an integer! | Pass |

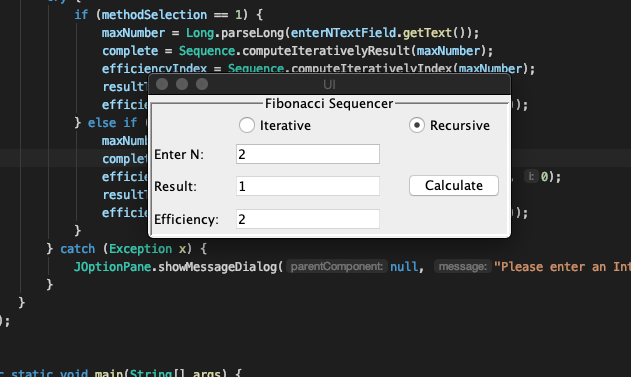


Figure Test Case 6

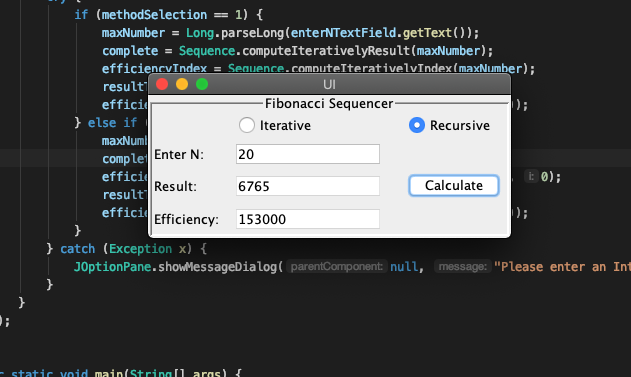


Figure Test Case 7

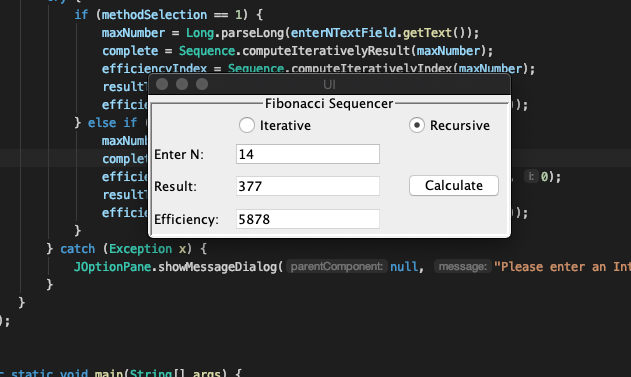


Figure Test Case 8

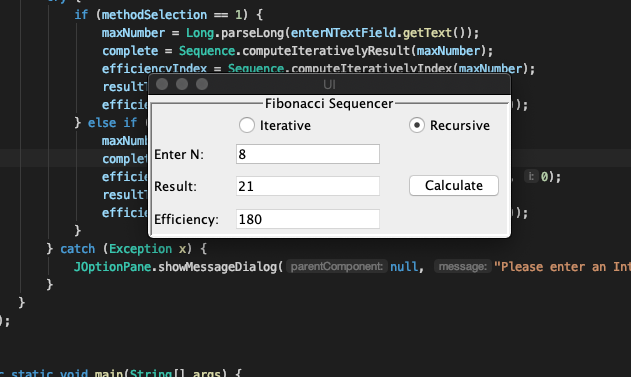


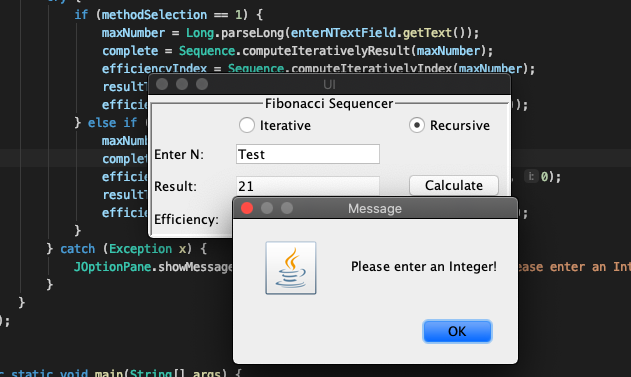
Figure Test Case 9

Figure Test Case 10



As can be seen from the chart above, iterative methods for solving problems are much more efficient than recursive methods. It took the iterative method a fraction of the cycles it took the recursive method to compute the value of N. Every time the recursive method is called on itself, it creates an entirely new copy of the method and all variables onto the stack.

Because I was curious, I decided to attempt to use the recursive method to discover the 50th number in the Fibonacci Sequence. Using an i9 processor with 32 gigs of RAM, after one minute the recursive method had not yet completed calculating and the program was locked up. The CPU fan went from quiet to sounding like a rocket ship in my living room, and the program had to be force closed. At that time, the CPU fan was back to normal operation within 4 seconds. On the other hand, the iterative method completed almost instantly, with no discernable effect on the CPU or CPU thermals.