Student: Eduardo de Oliveira Castro
Blazer ID: edc
Date: 01/14/2015

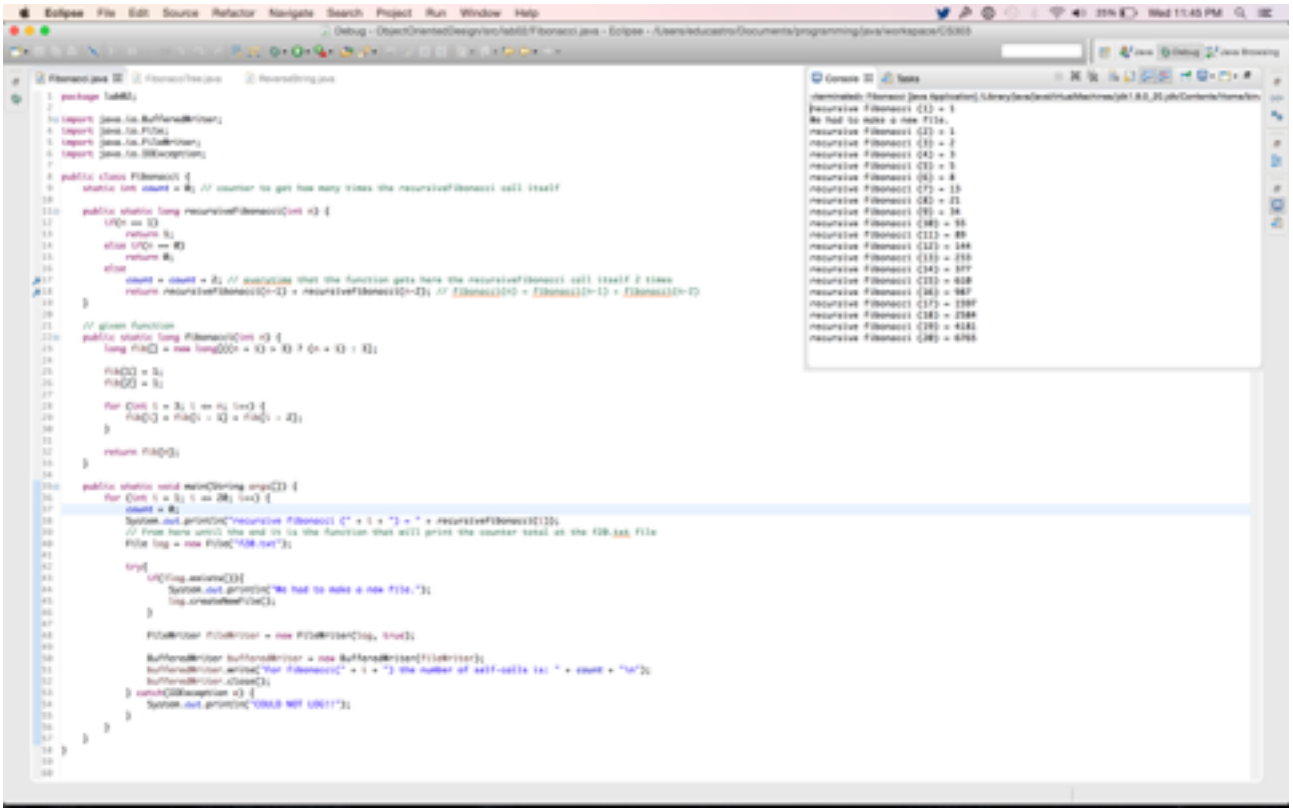
Report #02
Recursion

Problem)

It was demanded to create a recursive method for calculating the fibonacci sequence and to compare it with the procedure method. After this it was also demanded to print a tree showing the recursive procedure that this method makes. And for the last part it was demanded to create a recursive class that returns the inverse of a given string.

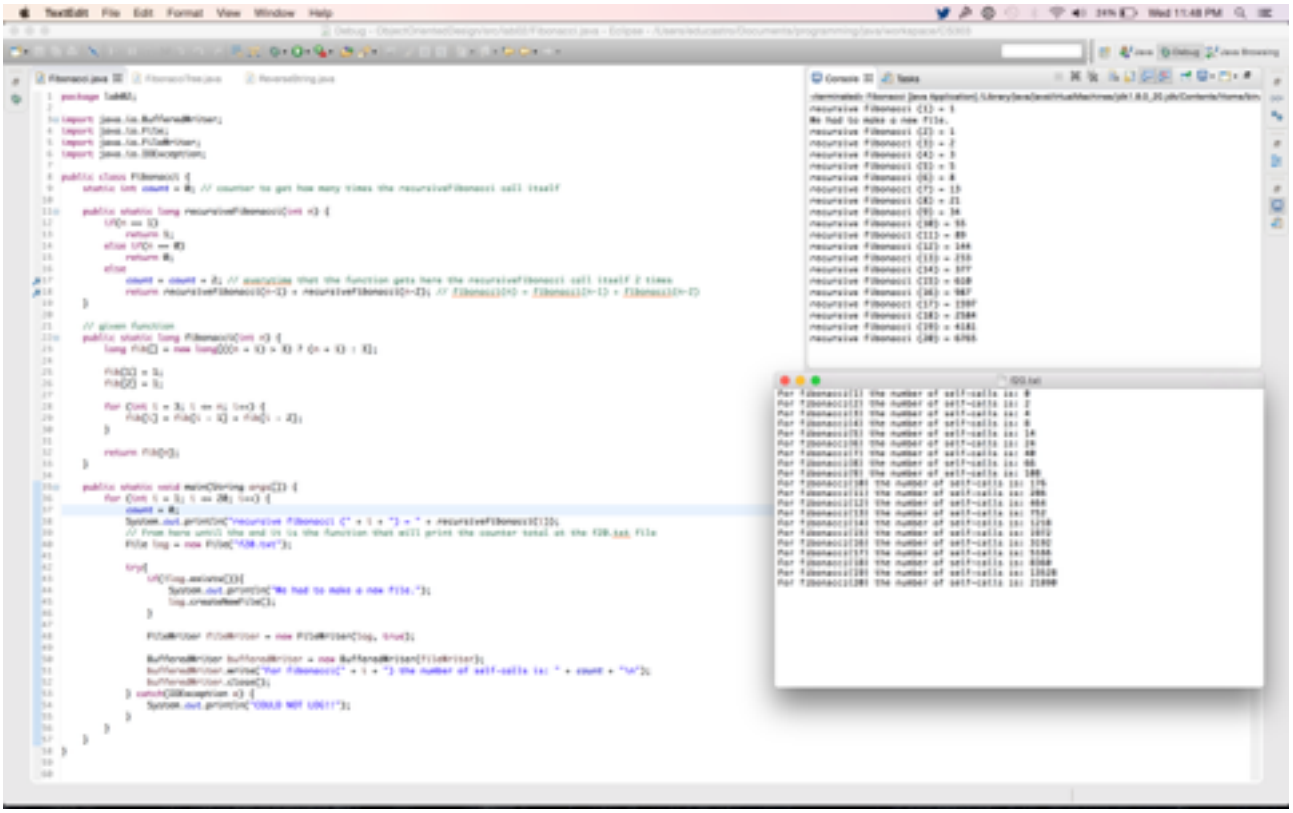
Process and Results)

For the first part I just created the recursive method with base cases n == 0 and n == 1. This was the function and the output.



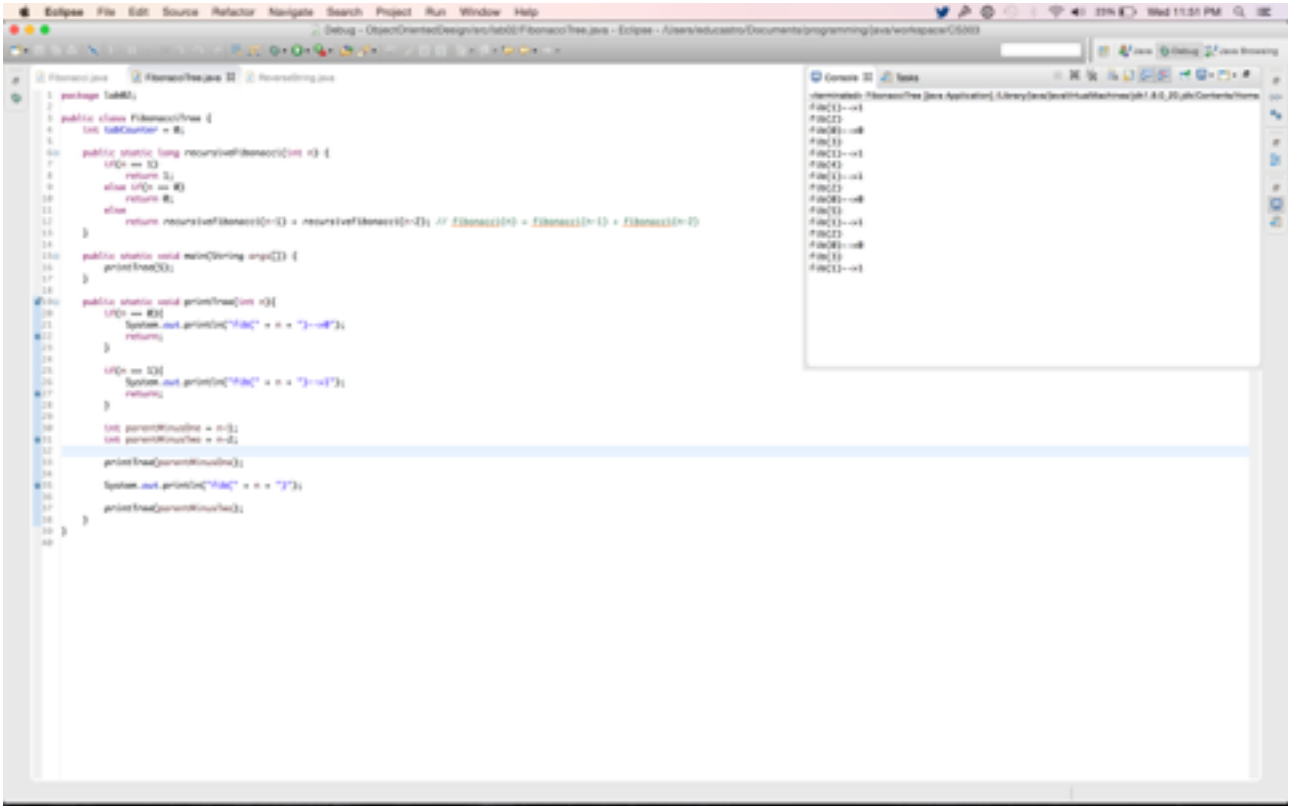
Recursive method and output

For the second part it was demanded to check how many times the recursive method called itself. Each time that the procedure goes to the last else it calls itself 2 times, so I just added 2 to the counter variable. This is the txt file output.

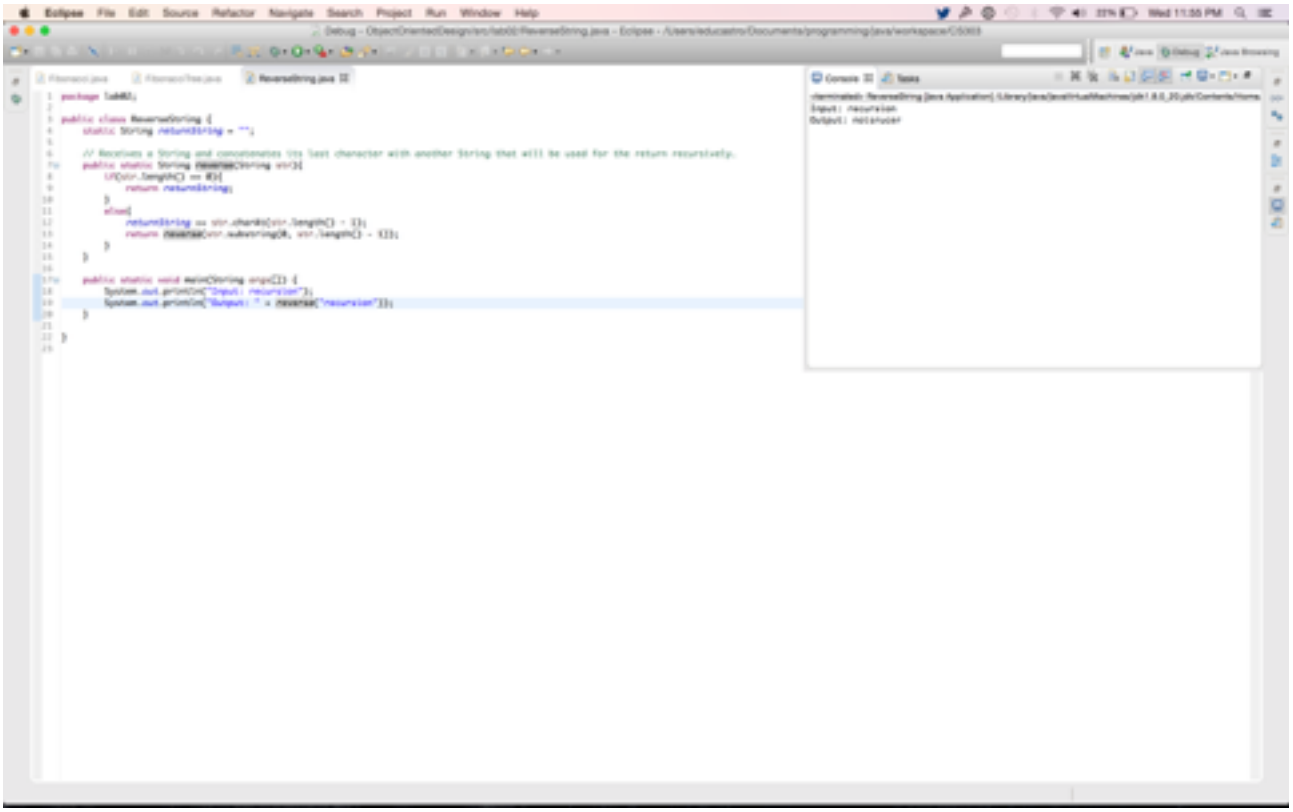


File output

The last part from problem 1 was the most difficult, I've tried a lot until figure it out how to print the tree. The bases cases was n == 0 and n == 1, so we printed the base values(0 and 1, respectively) and break the procedure. Then we split the work in two parts, the first and the second parents(n-1 and n-2, respectively). Something related to the divide-to-conquer approach. After finishing the first parent recursively I print it and then start the second procedure, with n-2. I didn't figure out how to print with the tab spaces correctly. This is the picture with the code and the output.



The second problem was pretty easy, I just had to create an external variable, an empty string, that in each call receives the last char from the input string and then I call it again with a new substring that finishes before the last char. When the length of the string is 0(after a lot of recursive calls the substring method wont have any character because all of them will be at the external string) then I return this external string, this is the base case. Here is a picture of the code and the result.



Output for the string problem recursion

Conclusion)

Recursion is a powerful way to solve problems in a simple, elegant, easy and fast way. I should try more because I had a lot of difficult until figure out how to do the tree, all the other exercises were pretty easy.