# **DYNAMIC PROGRAMMING**

**Dynamic programming** (DP) is an algorithm of breaking a problem into smaller sub-problems

There are two ways to solve DP problems:

**Top-down** or **memoization** or **recursive approach**. The idea of this approach is to save previous result and re-use them if a sub problem repeats. This technique of storing the results of already solved subproblems is called memoization.

* Time complexity: O(N)
* Space complexity: O(1)

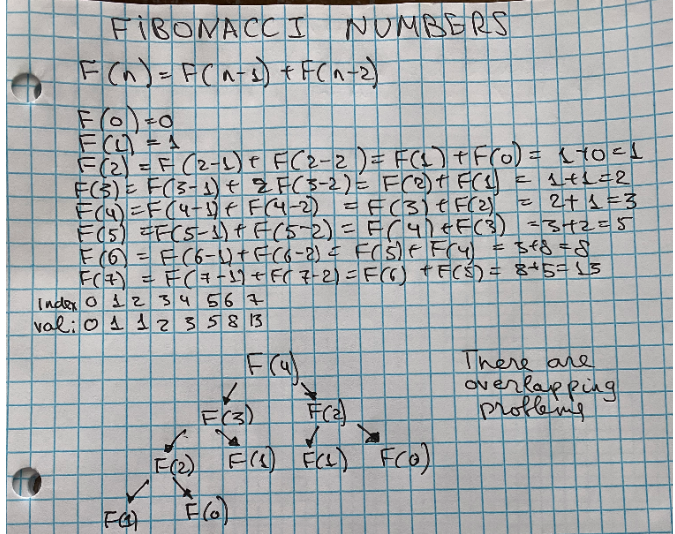
**Bottom-up** or **tabulation** or **iterative approach**. It is the opposite approach to memorization to avoid recursion. We don’t need to also save all previous results. For example, for Fibonacci numbers we need only two previous

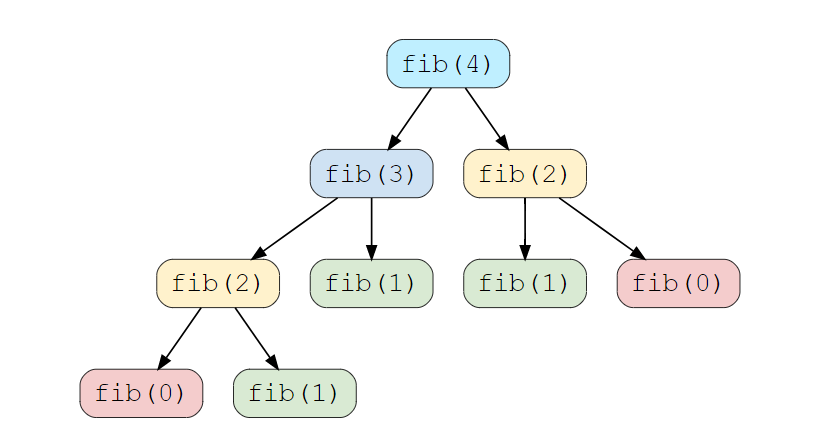
# **FIBONACCI PATTERN**

**Task. Calc Fibonacci numbers**.

It can be solved by three approaches

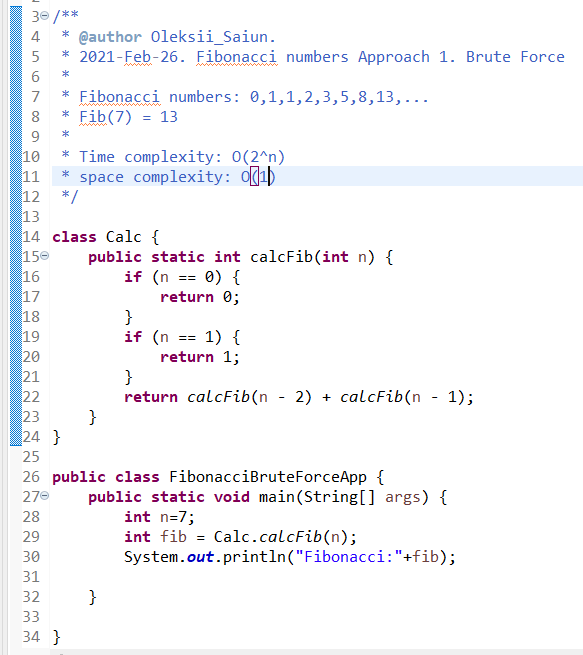
* Brute-force: Time complexity: O(2^N), Space complexity: O(1)
* Top-down (memoization): Time complexity: O(N), Space complexity: O(N)
* Top-down (memoization): Time complexity: O(N), Space complexity: O(N)



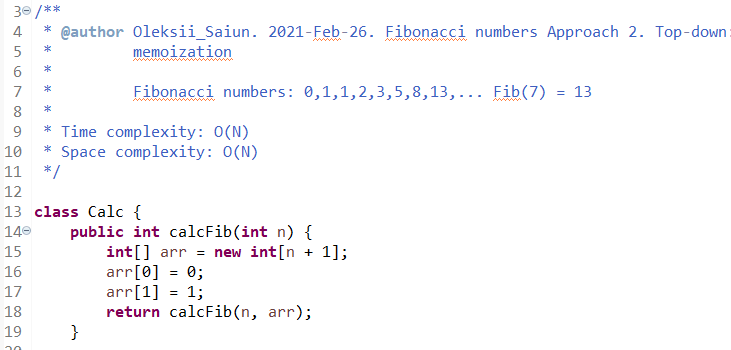


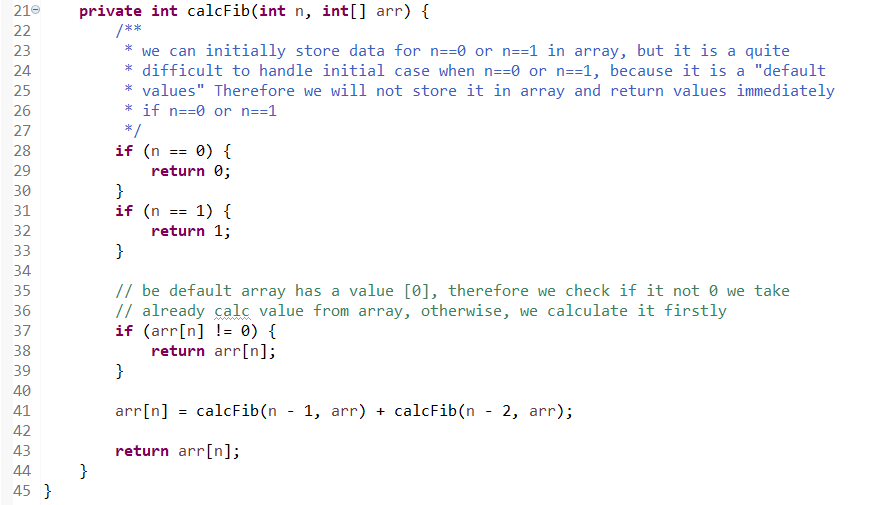
We can see there are overlapping problems. If we apply brute-force solution we will need to repeat calculations

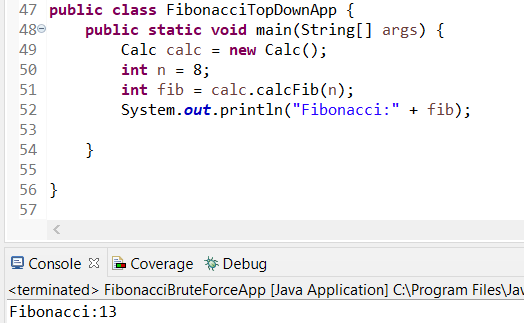
# **FIBONACCI NUMBERS. APPROACH 1. BRUTE-FORCE**



# **FIBONACCI NUMBERS. APPROACH 2. TOP-DOWN MEMOIZATION**

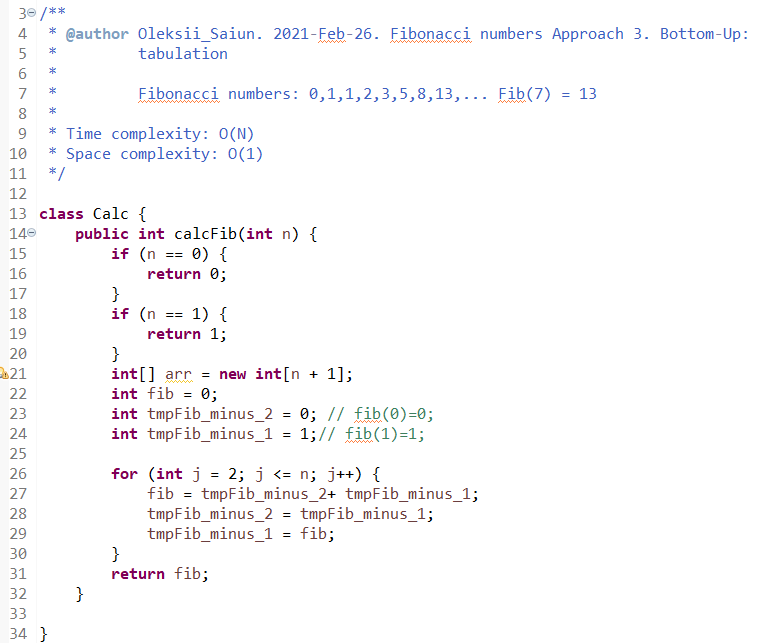


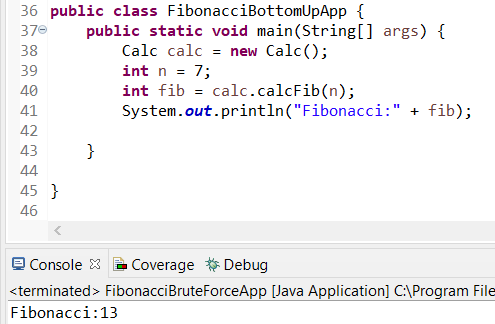




# **FIBONACCI NUMBERS. APPROACH 3. BOTTOM-UP TABULATION**

We can optimize the space used in our previous solution. We don’t need to store all the Fibonacci numbers up to ‘n’, as we only need two previous numbers to calculate the next Fibonacci number.



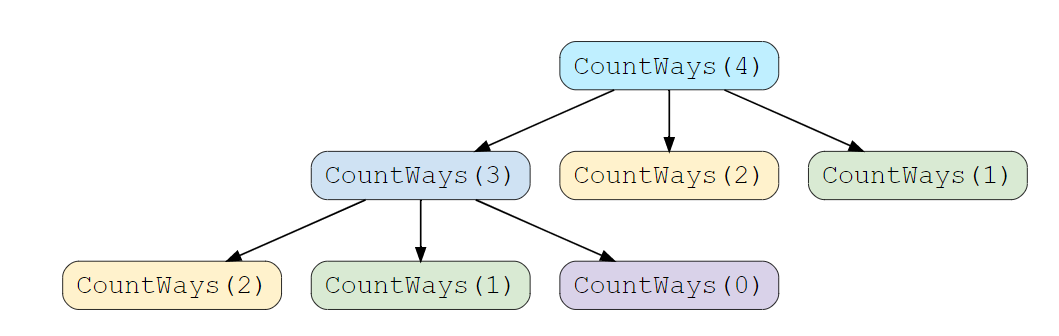


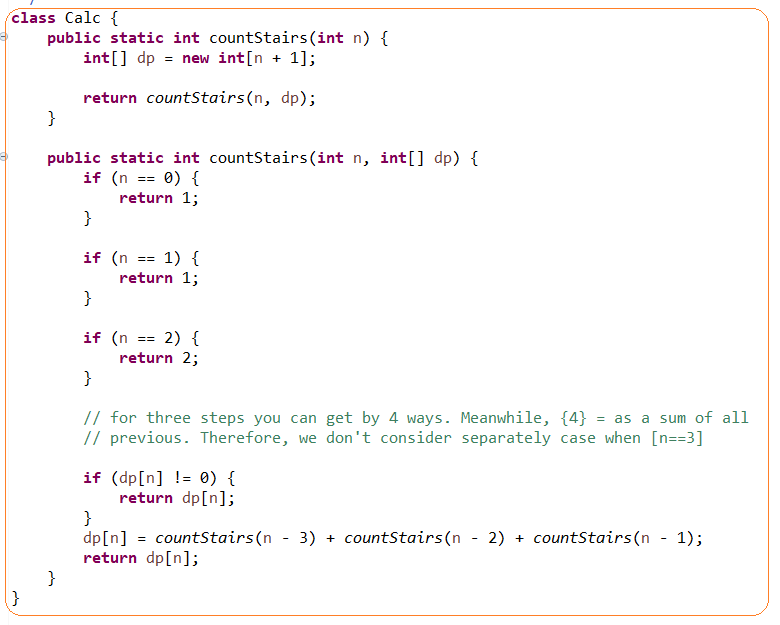
# **STAIRCASE. APPROACH 1. TOP-DOWN MEMOIZATION**

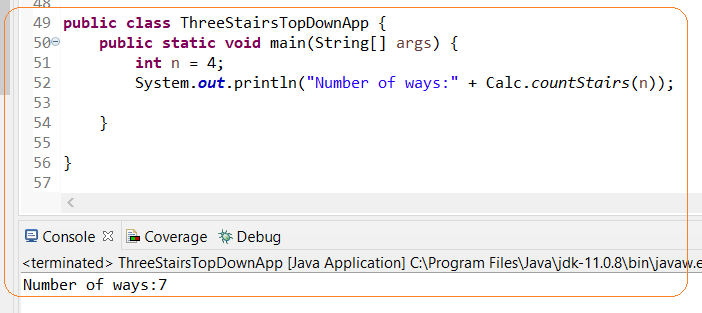
A Person can jump to {1} or {2} or {3} stairs. Find the number of possible ways to reach {n} stairs

* **0 stair**: can reach by 1 step (it is like a default case) {0,0}
* **1 stair**: can reach by 1 step {0,1}
* **2 stair**: can reach by 2 step {0,1,1}, {0,2}
* **3 stair**: can reach by 3 step {0,1,1,1}, {0,2,1}, {0,1,2}, {0,3}

0 stairs: can reach by 1 step (it is like a default case)







# **STAIRCASE. APPROACH 2. BOTTOM-UP TABULATION**

