**Exercise 7: Financial Forecasting**

**Scenario:**

You are developing a financial forecasting tool that predicts future values based on past data.

**Steps:**

1. **Understand Recursive Algorithms:**
   * Explain the concept of recursion and how it can simplify certain problems.
2. **Setup:**
   * Create a method to calculate the future value using a recursive approach.
3. **Implementation:**
   * Implement a recursive algorithm to predict future values based on past growth rates.
4. **Analysis:**
   * Discuss the time complexity of your recursive algorithm.
   * Explain how to optimize the recursive solution to avoid excessive computation.

**Solution:**

**1. Understand Recursive Algorithms**

**What is Recursion?**

Recursion is a programming technique where a function calls itself to solve smaller subproblems of a bigger problem.

**Why use recursion?**

* Simplifies complex problems like factorials, Fibonacci, tree traversals, etc.
* Helps break down problems naturally into smaller parts.

***Base Case* stops the recursion; *Recursive Case* keeps it going.**

**2. Setup: Forecasting Future Value**

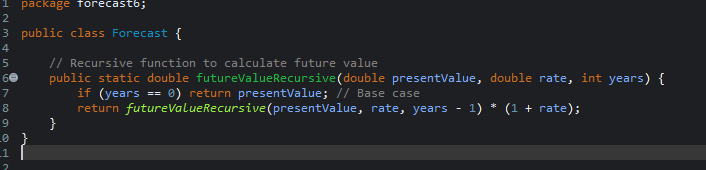
Let’s assume:

* futureValue(n) = presentValue × (1 + rate)^n
* We calculate it recursively:

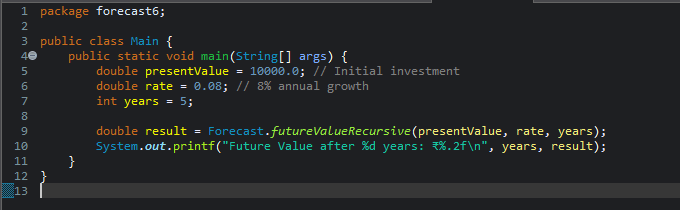
**futureValue(n) = futureValue(n - 1) \* (1 + rate)**

**3. Implementation in Java**

**Recursive Method for Financial Forecast**

****

**Main Method to Test Forecasting**

****

**Output:**

****

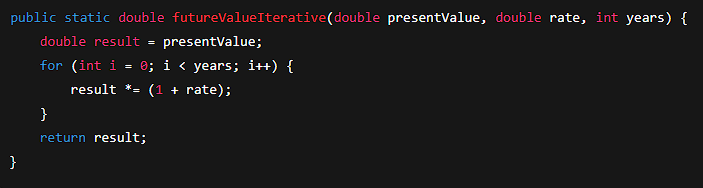
**4. Analysis**

**Time Complexity**

* Each call reduces years by 1 → O(n) where n is the number of years.
* Space complexity: O(n) due to the call stack.

**How to Optimize Recursion**

**Option 1: Use Iteration Instead**

****

Time Complexity: O(n)

Space Complexity: O(1) (no stack calls)