# FINANCIAL FORECASTING

# 1. Understanding Recursive Algorithms

Recursion is a programming technique where a method calls itself to solve the same problem. It simplifies complex problems like tree traversal, mathematical computations (e.g., factorial, Fibonacci), and divide-and-conquer problems.

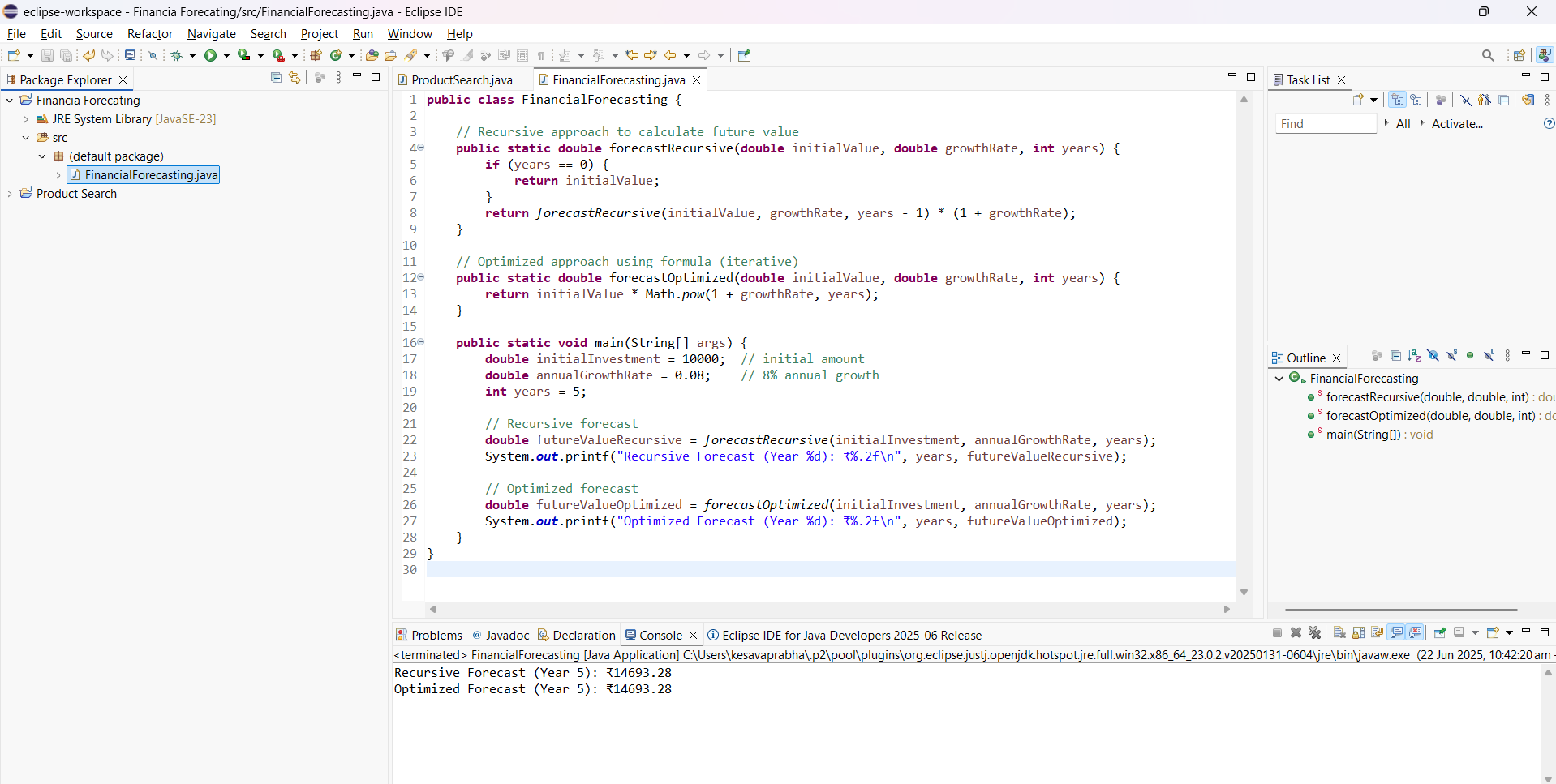
Advantages of recursion:  
- Cleaner, more intuitive code for problems that naturally fit a recursive pattern.  
- Reduces the need for loop constructs in some cases.

# 2. Setup

To forecast future financial values recursively, we define a method that takes:  
- initial value  
- growth rate  
- number of periods  
  
The method recursively calculates the future value using the formula:  
FV(n) = FV(n-1) \* (1 + growthRate)

# 3. Implementation: Recursive Forecasting public static double forecastValue(double initialValue, double growthRate, int years) { if (years == 0) { return initialValue; } return forecastValue(initialValue, growthRate, years - 1) \* (1 + growthRate); }

Output:



# 4. Analysis

Time Complexity:  
The recursive algorithm has a time complexity of O(n), where n is the number of years, because it makes a single recursive call per year.  
  
Optimization:  
- Use memoization or convert to an iterative approach to avoid redundant calls.  
- For financial computations, an iterative or closed-form solution is more efficient:  
 FV = initialValue \* Math.pow(1 + growthRate, years)