**Exercise 7: Financial Forecasting**

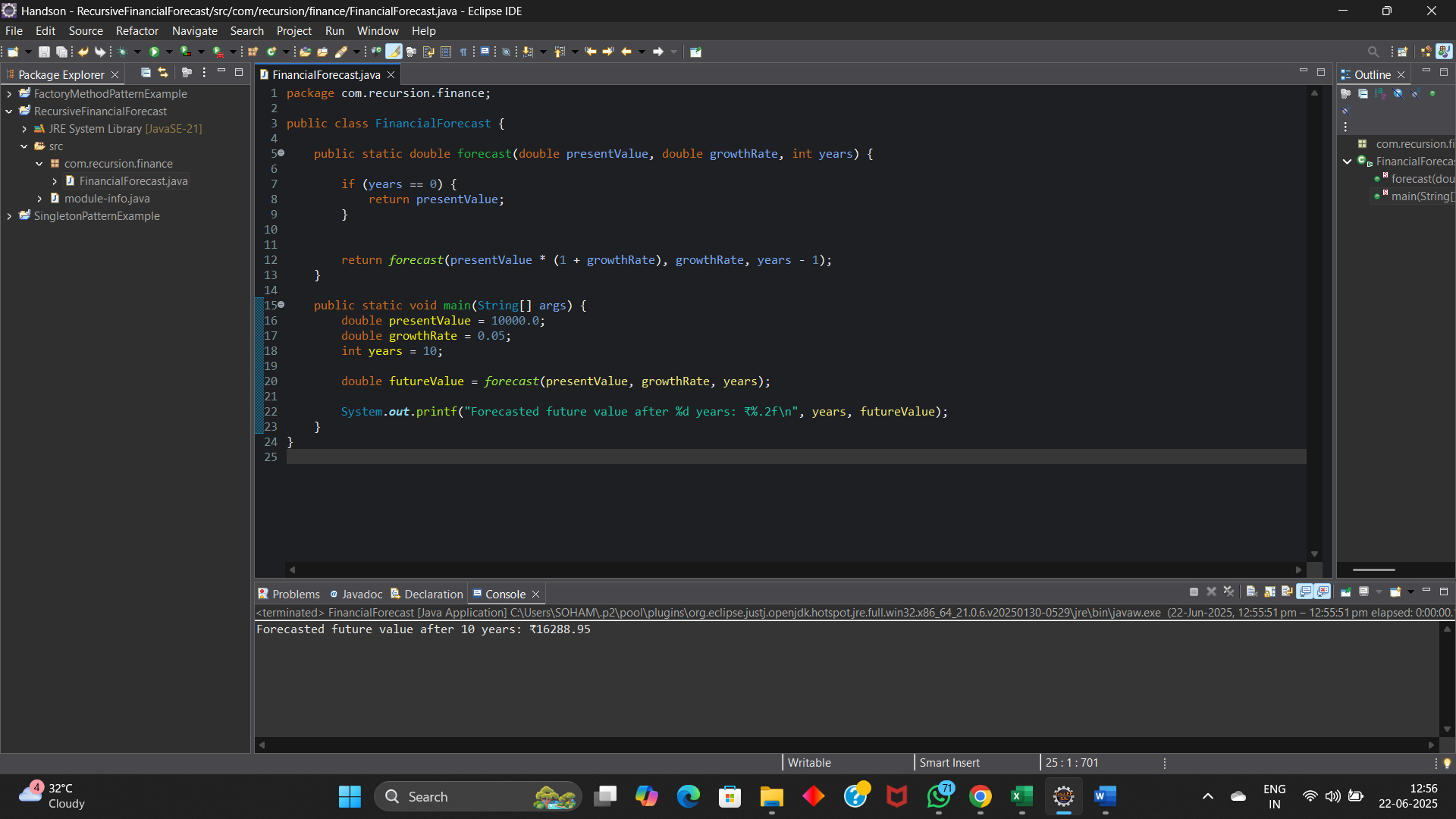
**Scenario:**

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

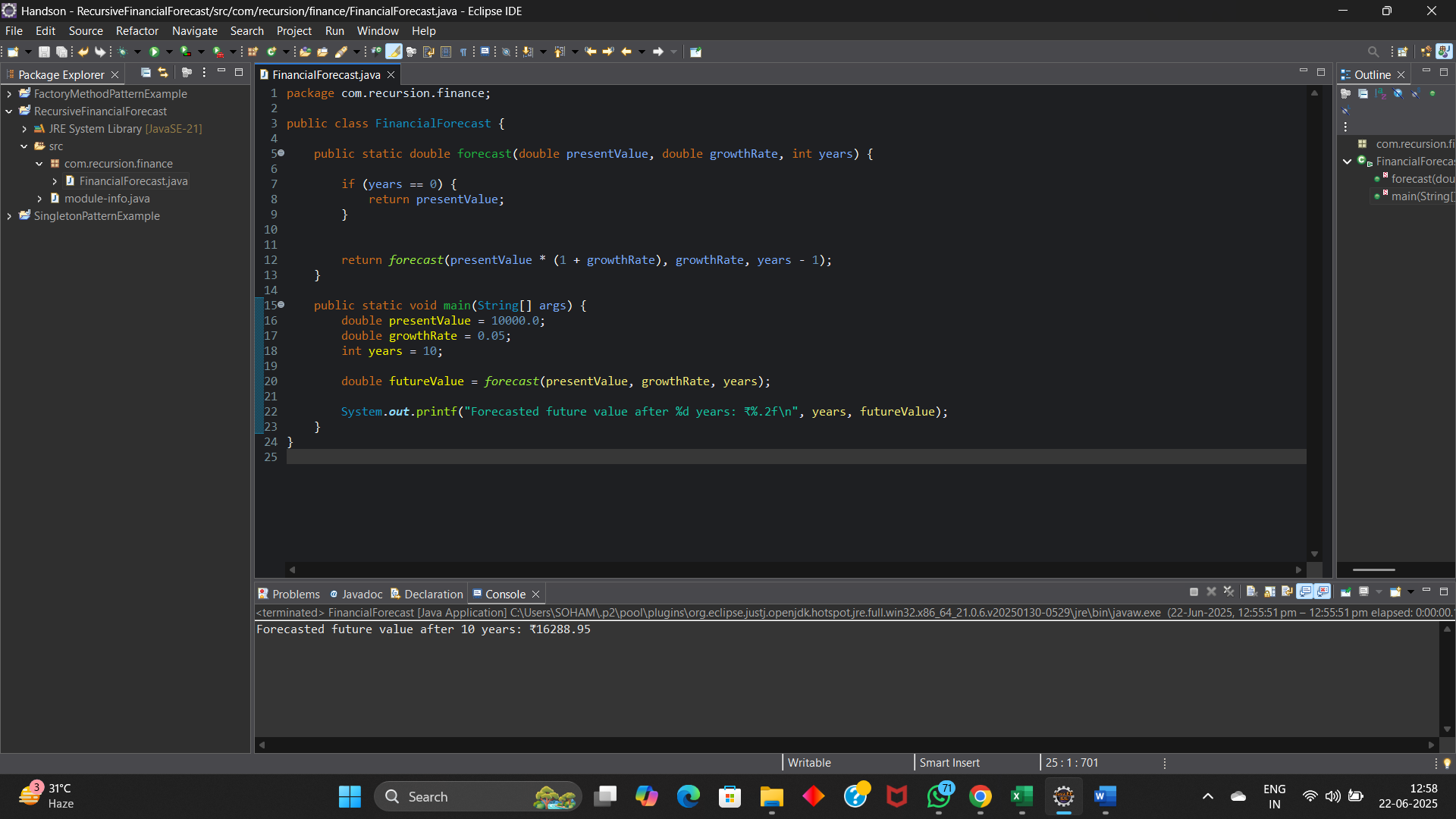
**Steps:**

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:**



**OUTPUT:**



**RECURSION:**

Recursion is a method where a function calls itself to solve smaller parts of the same problem. It continues until a base case is reached, which stops the recursion.

Recursion simplifies problems that have a repetitive or nested structure, like calculating factorials, solving Fibonacci series, or tree traversal. Instead of writing long loops, recursive code is shorter and easier to understand for such problems.

**TIME COMPLEXITY:**

O(n),as each recursive call reduces years by 1.

**How to optimize the recursive solution to avoid excessive computation?**

Recursive solutions can become slow if they repeat the same calculations many times. To optimize them, we can use techniques like memoization, which stores already calculated results and reuses them instead of recalculating.

Another way is to use iteration (loops) instead of recursion, which is faster and uses less memory. This helps avoid issues like stack overflow and makes the program more efficient.