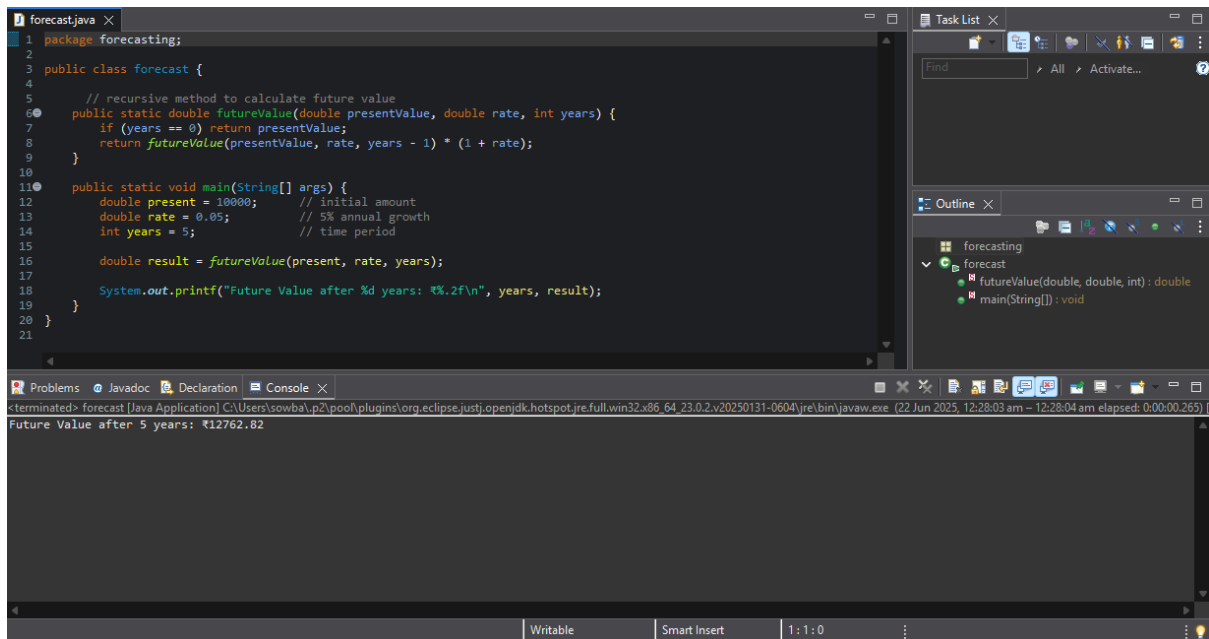


Exercise 7: Financial Forecasting

OUTPUT:



```
1 package forecasting;
2
3 public class forecast {
4
5     // recursive method to calculate future value
6     public static double futureValue(double presentValue, double rate, int years) {
7         if (years == 0) return presentValue;
8         return futureValue(presentValue, rate, years - 1) * (1 + rate);
9     }
10
11     public static void main(String[] args) {
12         double present = 10000; // initial amount
13         double rate = 0.05; // 5% annual growth
14         int years = 5; // time period
15
16         double result = futureValue(present, rate, years);
17
18         System.out.printf("Future Value after %d years: ₹%.2f\n", years, result);
19     }
20 }
21
```

Task List

Find

Outline

- forecasting
 - forecast
 - futureValue(double, double, int) : double
 - main(String[]) : void

Problems Javadoc Declaration Console

<terminated> forecast [Java Application] C:\Users\sowba\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.23.0.2.v20250131-0604\jre\bin\javaw.exe (22 Jun 2025, 12:28:03 am - 12:28:04 am elapsed: 0:00:00.265)

Future Value after 5 years: ₹12762.82

Writable Smart Insert 1:1:0

In the Financial Forecasting tool, recursion simplifies future value prediction based on past data. However, recursive methods can be inefficient ($O(2^n)$ in some cases), so optimization techniques like memoization or converting to iteration are essential for better performance.