**Exercise 7: Financial Forecasting**

Forecaster.java

package com.example.forecasting;

public class Forecaster {

    public static double calculateFutureValue(double principal, double rate, int years) {

        // Base case

        if (years == 0) {

            return principal;

        }

        // Recursive step

        double updatedPrincipal = principal \* (1 + rate / 100);

        return calculateFutureValue(updatedPrincipal, rate, years - 1);

    }

}

TestForecaster.java

package com.example.forecasting;

import java.util.Scanner;

public class TestForecaster {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

       System.out.print("Enter initial investment amount (₹): ");

        double principal = scanner.nextDouble();

        System.out.print("Enter annual growth rate (%): ");

        double rate = scanner.nextDouble();

        System.out.print("Enter number of years: ");

        int years = scanner.nextInt();

        // Calling recursive function

        double futureValue = Forecaster.calculateFutureValue(principal, rate, years);

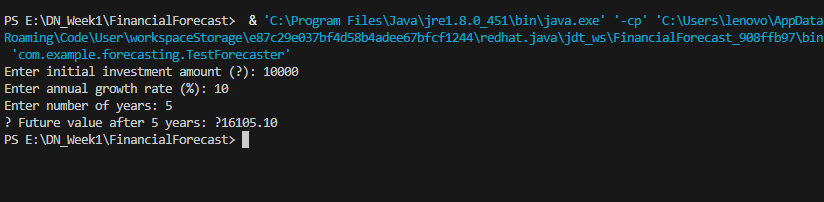
        System.out.printf("Future value after %d years: ₹%.2f\n", years, futureValue);

        scanner.close();

    }

}

OUTPUT:



ANALYSIS:

|  |  |  |  |
| --- | --- | --- | --- |
| Method | TimeComplexity | Space Complexity | Notes |
| Recursive | O(n) | O(n) | Simple and readable but stack-heavy |
| Iterative | O(n) | O(1) | Recommended for large input sizes |