Exercise 7: Financial Forecasting

Program :

import java.util.Map;

import java.util.HashMap;

public class FinancialForecast

{

    public static double forecast(double presentValue, double[] growthRates, int years) {

        if (years == 0) return presentValue;

        double prev = forecast(presentValue, growthRates, years - 1);

        return prev \* (1 + growthRates[years - 1]);

    }

    private static Map<Integer, Double> memo = new HashMap<>();

    public static double forecastMemo(double presentValue, double[] growthRates, int years) {

        if (years == 0) return presentValue;

        if (memo.containsKey(years)) return memo.get(years);

        double prev = forecastMemo(presentValue, growthRates, years - 1);

        double result = prev \* (1 + growthRates[years - 1]);

        memo.put(years, result);

        return result;

    }

    public static void main(String[] args) {

        double pv = 1000.0;

        double[] rates = {0.05, 0.04, 0.06, 0.05, 0.045};

        double simpleRec = forecast(pv, rates, rates.length);

        double optimized = forecastMemo(pv, rates, rates.length);

        System.out.printf("Simple recursive forecast: %.2f%n", simpleRec);

        System.out.printf("Memoized recursive forecast: %.2f%n", optimized);

    }

}