import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class FinancialForecast {

private List<Double> revenueHistory;

public FinancialForecast() {

revenueHistory = new ArrayList<>();

}

public void addMonthlyRevenue(double revenue) {

revenueHistory.add(revenue);

}

public double forecastNextMonthMovingAverage(int months) {

if (revenueHistory.size() < months) {

throw new IllegalArgumentException("Not enough data for the given moving average window.");

}

double sum = 0.0;

int startIndex = revenueHistory.size() - months;

for (int i = startIndex; i < revenueHistory.size(); i++) {

sum += revenueHistory.get(i);

}

return sum / months;

}

public double forecastFutureValueRecursive(double currentValue, double growthRate, int months) {

if (months == 0) {

return currentValue;

}

return forecastFutureValueRecursive(currentValue \* (1 + growthRate), growthRate, months - 1);

}

public void showRevenueHistory() {

System.out.println("Revenue History:");

for (int i = 0; i < revenueHistory.size(); i++) {

System.out.printf("Month %d: $%.2f\n", i + 1, revenueHistory.get(i));

}

}

public static void main(String[] args) {

FinancialForecast forecast = new FinancialForecast();

Scanner scanner = new Scanner(System.in);

System.out.print("Enter number of months of past revenue data: ");

int totalMonths = scanner.nextInt();

for (int i = 1; i <= totalMonths; i++) {

System.out.print("Enter revenue for month " + i + ": $");

double revenue = scanner.nextDouble();

forecast.addMonthlyRevenue(revenue);

}

forecast.showRevenueHistory();

System.out.print("Enter number of months to use for moving average forecast: ");

int averageMonths = scanner.nextInt();

try {

double predictedMA = forecast.forecastNextMonthMovingAverage(averageMonths);

System.out.printf("Forecasted revenue for next month (Moving Average): $%.2f\n", predictedMA);

} catch (IllegalArgumentException e) {

System.out.println("Error: " + e.getMessage());

}

System.out.print("Enter current revenue value to forecast recursively: $");

double currentValue = scanner.nextDouble();

System.out.print("Enter monthly growth rate as a decimal (e.g., 0.05 for 5%%): ");

double growthRate = scanner.nextDouble();

System.out.print("Enter number of months to forecast recursively: ");

int monthsToForecast = scanner.nextInt();

double predictedRecursive = forecast.forecastFutureValueRecursive(currentValue, growthRate, monthsToForecast);

}

}

Output:

Enter number of months of past revenue data: 6

Enter revenue for month 1: $10000

Enter revenue for month 2: $20000

Enter revenue for month 3: $30000

Enter revenue for month 4: $40000

Enter revenue for month 5: $50000

Enter revenue for month 6: $60000

Revenue History:

Month 1: $10000.00

Month 2: $20000.00

Month 3: $30000.00

Month 4: $40000.00

Month 5: $50000.00

Month 6: $60000.00

Enter number of months to use for moving average forecast: 4

Forecasted revenue for next month (Moving Average): $45000.00

Enter current revenue value to forecast recursively: $10000

Enter monthly growth rate as a decimal (e.g., 0.05 for 5%): 0.05

Enter number of months to forecast recursively: 6

Forecasted revenue after 6 months (Recursive Growth): $13400.95