Exercise 2: E-commerce Platform Search Function

import java.util.\*;

class Product {

String name;

double price;

Product(String name, double price) {

this.name = name;

this.price = price;

}

@Override

public String toString() {

return name + " - $" + price;

}

}

public class ECommerceSearchFunction {

private static List<Product> searchProduct(List<Product> products, String query) {

List<Product> results = new ArrayList<>();

for (Product product : products) {

if (product.name.toLowerCase().contains(query.toLowerCase())) {

results.add(product);

}

}

return results;

}

public static void main(String[] args) {

List<Product> products = Arrays.asList(

new Product("Laptop", 999.99),

new Product("Smartphone", 499.99),

new Product("Tablet", 299.99),

new Product("Smartwatch", 199.99),

new Product("Headphones", 89.99)

);

Scanner scanner = new Scanner(System.in);

System.out.print("Enter search query: ");

String query = scanner.nextLine();

List<Product> results = searchProduct(products, query);

if (results.isEmpty()) {

System.out.println("No products found.");

} else {

System.out.println("Search Results:");

for (Product product : results) {

System.out.println(product);

}

}

scanner.close();

}

}

Exercise 7: Financial Forecasting

import java.util.\*;

public class FinancialForecasting {

public static double[] movingAverage(double[] data, int windowSize) {

if (data.length < windowSize || windowSize <= 0) return new double[0];

double[] result = new double[data.length - windowSize + 1];

double sum = 0;

for (int i = 0; i < windowSize; i++) {

sum += data[i];

}

result[0] = sum / windowSize;

for (int i = windowSize; i < data.length; i++) {

sum += data[i] - data[i - windowSize];

result[i - windowSize + 1] = sum / windowSize;

}

return result;

}

public static void main(String[] args) {

double[] revenueData = {1000, 1050, 980, 1200, 1150, 1250, 1300, 1400};

int windowSize = 3;

double[] forecast = movingAverage(revenueData, windowSize);

System.out.println("Financial Forecast (Moving Average):");

for (double value : forecast) {

System.out.printf("%.2f\n", value);

}

}

}