Exercise 2: E-commerce Platform Search Function

Product.java:

public class Product {

    int productId;

    String productName;

    String category;

    public Product(int productId, String productName, String category) {

        this.productId = productId;

        this.productName = productName;

        this.category = category;

    }

    @Override

    public String toString() {

        return "Product(ID=" + productId + ", Name='" + productName + "', Category='" + category + "')";

    }

}

ECommerceSearch.java:

import java.util.\*;

public class ECommerceSearch {

      public static void main(String[] args) {

            Product[] products = {

                        new Product(101, "Laptop", "Electronics"),

                        new Product(102, "Shoes", "Footwear"),

                        new Product(103, "Watch", "Accessories"),

                        new Product(104, "Phone", "Electronics"),

                        new Product(105, "Bag", "Fashion")

            };

            System.out.println("Linear Search Result:");

            Product result1 = linearSearch(products, "Phone");

            System.out.println(result1);

            Arrays.sort(products, Comparator.comparing(p -> p.productName));

            System.out.println("\nBinary Search Result:");

            Product result2 = binarySearch(products, "Phone");

            System.out.println(result2);

      }

      public static Product linearSearch(Product[] products, String targetName) {

            for (Product product : products) {

                  if (product.productName.equals(targetName)) {

                        return product;

                  }

            }

            return null;

      }

      public static Product binarySearch(Product[] products, String targetName) {

            int left = 0;

            int right = products.length - 1;

            while (left <= right) {

                  int mid = (left + right) / 2;

                  String midName = products[mid].productName;

                  int comparison = midName.compareTo(targetName);

                  if (comparison == 0) {

                        return products[mid];

                  } else if (comparison < 0) {

                        left = mid + 1;

                  } else {

                        right = mid - 1;

                  }

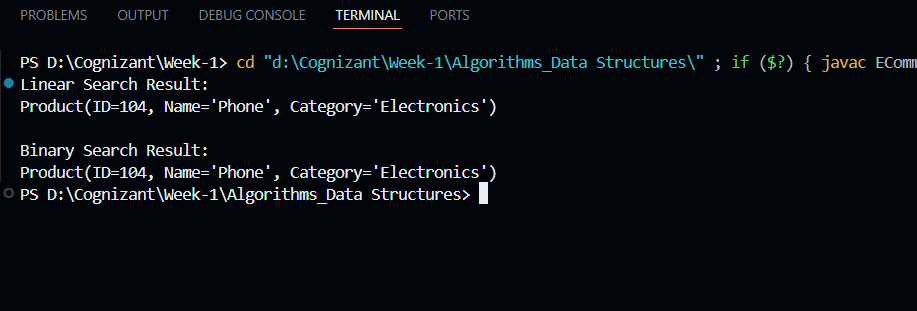
            }

            return null;

      }

}

Output:



Exercise 7: Financial Forecasting

FinancialForecast.java:

public class FinancialForecast {

      public static double forecastFutureValue(double initialAmount, double growthRate, int periods) {

            if (periods == 0) {

                  return initialAmount;

            }

            return forecastFutureValue(initialAmount, growthRate, periods - 1) \* (1 + growthRate);

      }

      public static void main(String[] args) {

            double initialAmount = 1000.0;

            double growthRate = 0.05;

            int periods = 5;

            double futureValue = forecastFutureValue(initialAmount, growthRate, periods);

            System.out.println("Future Value after " + periods + " periods: $" + futureValue);

      }

}

Output:

