**Exercise 2: E-commerce Platform Search Function**

Product.java

package Ecommerce;

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 "[" + productId + "] " + productName + " - " + category;

}

}

SearchService.java

package Ecommerce;

import java.util.Arrays;

import java.util.Comparator;

public class SearchService {

// Linear Search

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

for (Product product : products) {

if (product.productName.equalsIgnoreCase(name)) {

return product;

}

}

return null;

}

// Binary Search (requires sorted array)

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

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

int left = 0, right = products.length - 1;

while (left <= right) {

int mid = (left + right) / 2;

int cmp = products[mid].productName.compareToIgnoreCase(name);

if (cmp == 0) return products[mid];

else if (cmp < 0) left = mid + 1;

else right = mid - 1;

}

return null;

}

}

Main.java

package Ecommerce;

public class Main {

public static void main(String[] args) {

Product[] products = {

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

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

new Product(103, "Mobile", "Electronics"),

new Product(104, "T-shirt", "Clothing")

};

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

Product result1 = SearchService.*linearSearch*(products, "Mobile");

if (result1 != null)

System.***out***.println("Found: " + result1);

else

System.***out***.println("Product not found");

System.***out***.println("\n🔍 Binary Search Result:");

Product result2 = SearchService.*binarySearch*(products, "Mobile");

if (result2 != null)

System.***out***.println("Found: " + result2);

else

System.***out***.println("Product not found");

}

}

Output:

Linear Search Result:

Found: [103] Mobile - Electronics

Binary Search Result:

Found: [103] Mobile - Electronics

**Exercise 7: Financial Forecasting**

ForecastCalculator.java

package Financial;

public class ForecastCalculator {

public double calculateRecursive(double baseValue, double rate, int years) {

if (years == 0) {

return baseValue;

} else {

return calculateRecursive(baseValue, rate, years - 1) \* (1 + rate);

}

}

public double calculateIterative(double baseValue, double rate, int years) {

double result = baseValue;

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

result \*= (1 + rate);

}

return result;

}

}

Main.java

package Financial;

public class Main {

public static void main(String[] args) {

ForecastCalculator calculator = new ForecastCalculator();

double base = 1000.0;

double rate = 0.10;

int years = 5;

double valueByRecursion = calculator.calculateRecursive(base, rate, years);

double valueByIteration = calculator.calculateIterative(base, rate, years);

System.***out***.printf("Future Value (Recursive): %.2f\n", valueByRecursion);

System.***out***.printf("Future Value (Iterative): %.2f\n", valueByIteration);

}

}

Output:

Future Value (Recursive): 1610.51

Future Value (Iterative): 1610.51