**Algorithms Data Structures**

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

**SCENARIO :**

You are working on the search functionality of an e-commerce platform. The search needs to be optimized for fast performance.

**SOURCE CODE :**

**Product.java**

package com.ecommerce.search;

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 [productId=" + productId + ", productName=" + productName + ", category=" + category + "]";

}

}

**SearchDemo.java**

package com.ecommerce.search;

import java.util.\*;

public class SearchDemo {

public static Product linearSearchByName(Product[] products, String productName) {

for (Product product : products) {

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

return product;

}

}

return null;

}

public static Product linearSearchById(Product[] products, int id) {

for (Product product : products) {

if (product.productId == id) {

return product;

}

}

return null;

}

public static List<Product> linearSearchByCategory(Product[] products, String category) {

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

for (Product product : products) {

if (product.category.equalsIgnoreCase(category)) {

results.add(product);

}

}

return results;

}

public static Product binarySearchByName(Product[] products, String productName) {

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

while (left <= right) {

int mid = left + (right - left) / 2;

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

if (result == 0) {

return products[mid];

} else if (result < 0) {

right = mid - 1;

} else {

left = mid + 1;

}

}

return null;

}

public static void main(String[] args) {

Product[] products = {

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

new Product(102, "Shirt", "Clothing"),

new Product(103, "Book", "Education"),

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

new Product(105, "Shoes", "Footwear")

};

// Sort for binary search

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

Scanner scanner = new Scanner(System.*in*);

int choice;

do {

System.*out*.println("\n=== E-Commerce Product Search ===");

System.*out*.println("1. Search by Product Name (Linear)");

System.*out*.println("2. Search by Product Name (Binary)");

System.*out*.println("3. Search by Product ID");

System.*out*.println("4. Search by Category");

System.*out*.println("0. Exit");

System.*out*.print("Enter your choice: ");

choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

System.*out*.print("Enter product name: ");

String name1 = scanner.nextLine();

Product linearResult = *linearSearchByName*(products, name1);

System.*out*.println(linearResult != null ? linearResult : "Product not found.");

break;

case 2:

System.*out*.print("Enter product name: ");

String name2 = scanner.nextLine();

Product binaryResult = *binarySearchByName*(products, name2);

System.*out*.println(binaryResult != null ? binaryResult : "Product not found.");

break;

case 3:

System.*out*.print("Enter product ID: ");

int id = scanner.nextInt();

Product idResult = *linearSearchById*(products, id);

System.*out*.println(idResult != null ? idResult : "Product not found.");

break;

case 4:

System.*out*.print("Enter category: ");

String category = scanner.nextLine();

List<Product> categoryResults = *linearSearchByCategory*(products, category);

if (categoryResults.isEmpty()) {

System.*out*.println("No products found in this category.");

} else {

for (Product p : categoryResults) {

System.*out*.println(p);

}

}

break;

case 0:

System.*out*.println("Exiting... Thank you!");

break;

default:

System.*out*.println("Invalid choice. Try again.");

}

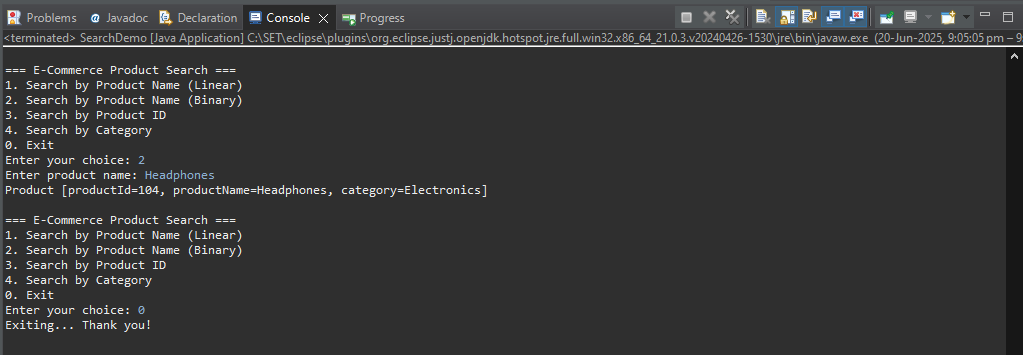
} while (choice != 0);

scanner.close();

}

}

**OUTPUT :**

****