**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.

**Code:**

**Product.Java**

package dsa;

public class Product {

int productId;

String productName;

String category;

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

this.productId = productId;

this.productName = productName;

this.category = category;

}

@Override

public String toString() {

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

}

}

**Main.Java**

package dsa;

import java.util.Arrays;

import java.util.Comparator;

class ProductSearch {

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

for (Product p : products) {

if (p.productName.equalsIgnoreCase(targetName)) {

return p;

}

}

return null;

}

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

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

while (left <= right) {

int mid = (left + right) / 2;

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

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

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

else right = mid - 1;

}

return null;

}

}

public class Main {

public static void main(String[] args) {

Product[] products = {

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

new Product(102, "Dress", "Fashion"),

new Product(103, "Pens", "Stationary"),

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

};

Product Linear = ProductSearch.*linearSearch*(products, "Mouse");

System.*out*.println("Linear Search Result: " + (Linear != null ? Linear : "Not found"));

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

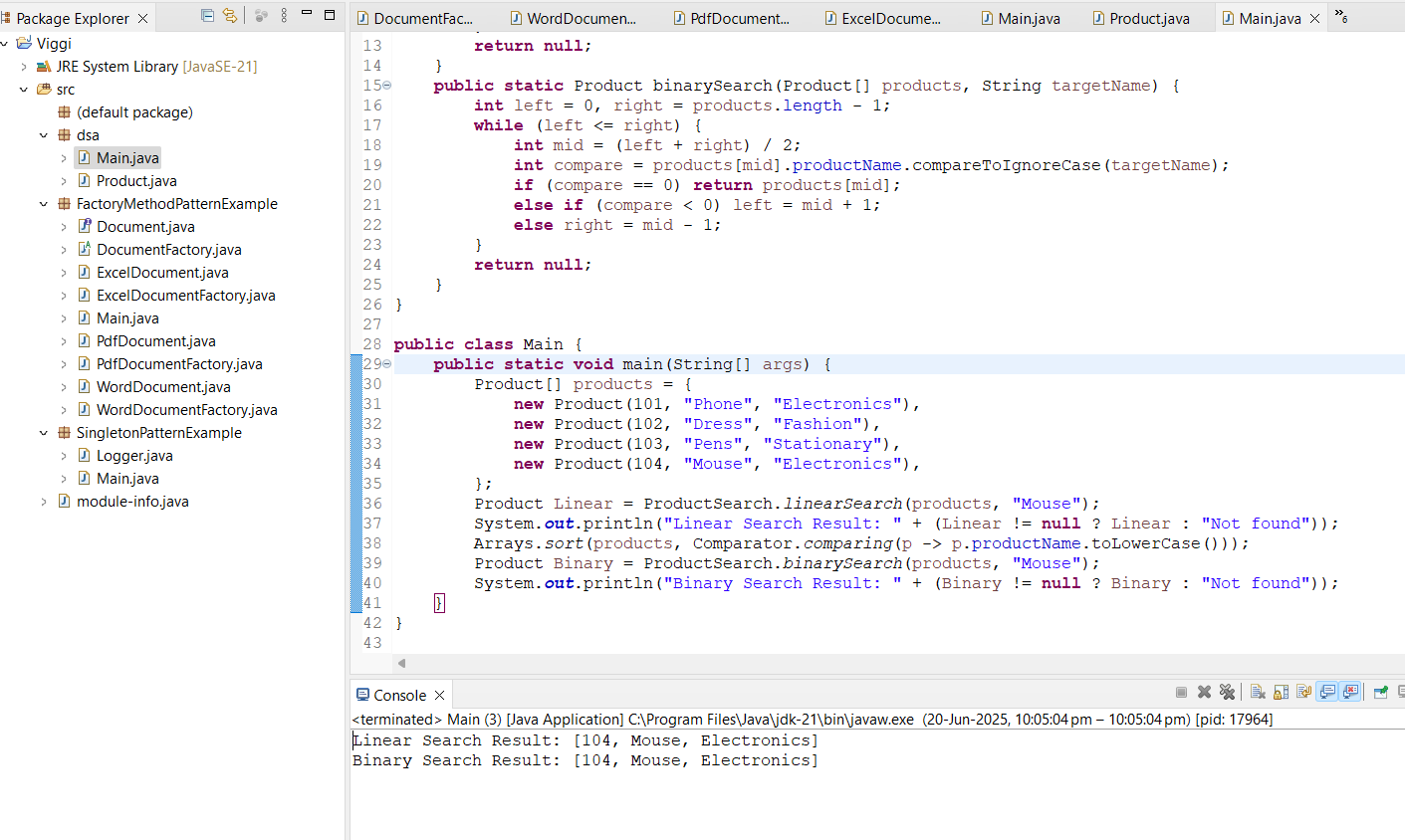
Product Binary = ProductSearch.*binarySearch*(products, "Mouse");

System.*out*.println("Binary Search Result: " + (Binary != null ? Binary : "Not found"));

}

}

**Output:**

****

* **Linear Search:**
  + Best: O(1)
  + Worst: O(n)
  + Works on unsorted data
* **Binary Search:**
  + Best: O(1)
  + Worst: O(log n)
  + Needs sorted data

**Recommendation:** For larger datasets, use binary search with sorted arrays. So for our E-commerce Platform Search Binary Search is optimal to use.