# E-COMMERCE PLATFORM SEARCH FUNCTION

# 1. Understanding Asymptotic Notation

## Big O Notation

Big O notation describes the time complexity of an algorithm — how the runtime increases as the input size grows. It helps us compare algorithms without implementation or hardware bias.  
  
Common complexities:  
- O(1): Constant time  
- O(n): Linear time  
- O(log n): Logarithmic time  
- O(n²): Quadratic time

## Best, Average, and Worst Case for Search

Linear Search:  
- Best Case: O(1) – match at first position  
- Average Case: O(n/2) -> O(n)  
- Worst Case: O(n)  
  
Binary Search:  
- Best Case: O(1) – match at middle  
- Average Case: O(log n)  
- Worst Case: O(log n)  
  
\* Binary Search requires sorted data.

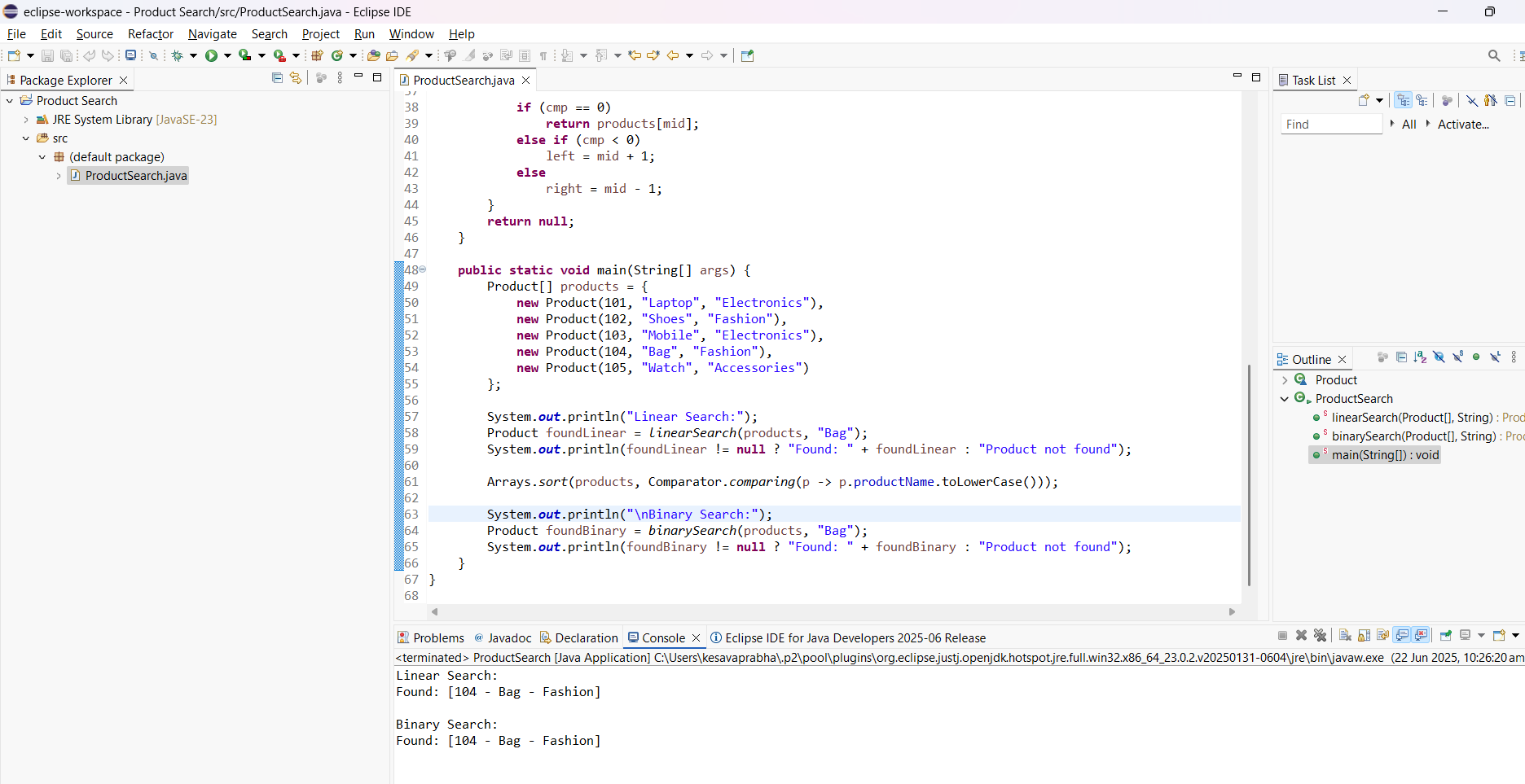
# 2. Setup: Product Class

The Product class contains attributes used for searching such as productId, productName, and category.  
  
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 + "]";  
 }  
}

# 3. Implementation: Linear and Binary Search

Both Linear and Binary search methods are implemented. Linear search goes through each element, while Binary search uses divide-and-conquer on a sorted array.

Output:



# 4. Analysis: Time Complexity & Suitability

Time Complexity:  
Linear Search: O(n)  
Binary Search: O(log n)  
  
Suitability:  
- Use Linear Search for small or unsorted datasets.  
- Use Binary Search for large, sorted datasets for better performance.