**“ALGORITHMS AND DATA STRUCTURES”**

**EXERCISE 2 : E-Commerce Platform Search Function**

**1.Product.java**

**package** E\_Commerce;

**class** Product {

**int** productId;

String productName;

String category;

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

// **TODO** Auto-generated constructor stub

**this**.productId = productId;

**this**.productName = productName;

**this**.category = category;

}

@Override

**public** String toString() {

**return** "Product { " +

"ID = " + productId +

", Name = '" + productName + '\'' +

", Category = '" + category+'\''+'}';

}

}

**2.Search.java**

**package** E\_Commerce;

**import** java.util.Arrays;

**import** java.util.Comparator;

**public** **class** Search {

// Linear Search

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

**for** (Product product : products) {

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

**return** product;

}

}

**return** **null**;

}

// Binary Search (on sorted array)

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

**int** left = 0;

**int** right = products.length - 1;

**while** (left <= right) {

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

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

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

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

**else** right = mid - 1;

}

**return** **null**;

}

**public** **static** **void** main(String[] args) {

// Sample products

Product[] products = {

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

**new** Product(102, "Shirt", "Apparel"),

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

**new** Product(104, "Book", "Stationery"),

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

};

// Linear Search Test

System.***out***.println("Linear Search for 'Mobile':");

Product foundLinear = *linearSearch*(products, "Mobile");

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

// Sort array for binary search

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

// Binary Search Test

System.***out***.println("\nBinary Search for 'Mobile':");

Product foundBinary = *binarySearch*(products, "Mobile");

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

// Binary Search for missing product

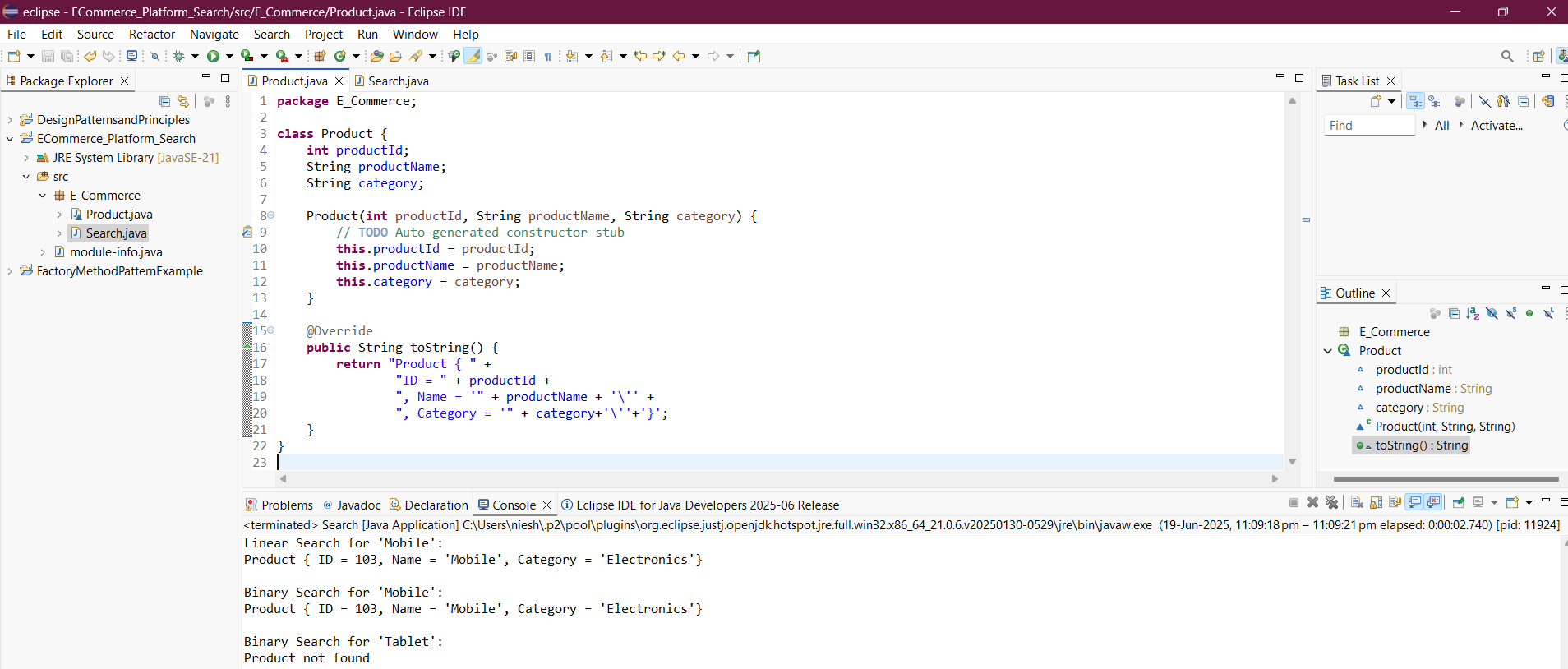
System.***out***.println("\nBinary Search for 'Tablet':");

Product missing = *binarySearch*(products, "Tablet");

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

}

}



**----------------------------------------------------------------------------------------------------------------**

**Exercise 7: Financial Forecasting**

**1.FinancialForecast.java**

**package** FinancialForecastExample;

**public** **class** FinancialForecast {

// Recursive method

**public** **static** **double** futureValueRecursive(**double** principal, **double** rate, **int** years) {

**if** (years == 0) {

**return** principal;

}

**return** *futureValueRecursive*(principal, rate, years - 1) \* (1 + rate);

}

// Iterative method

**public** **static** **double** futureValueIterative(**double** principal, **double** rate, **int** years) {

**double** result = principal;

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

result \*= (1 + rate);

}

**return** result;

}

**public** **static** **void** main(String[] args) {

// Example:

**double** principal = 10000.0; // ₹10,000 initial investment

**double** rate = 0.10; // 10% annual growth

**int** years = 5; // Forecast for 5 years

// Calculate future values

**double** recursiveResult = *futureValueRecursive*(principal, rate, years);

**double** iterativeResult = *futureValueIterative*(principal, rate, years);

// Display results

System.***out***.println("Example:");

System.***out***.println("Principal = ₹" + principal);

System.***out***.println("Rate = 10% per year");

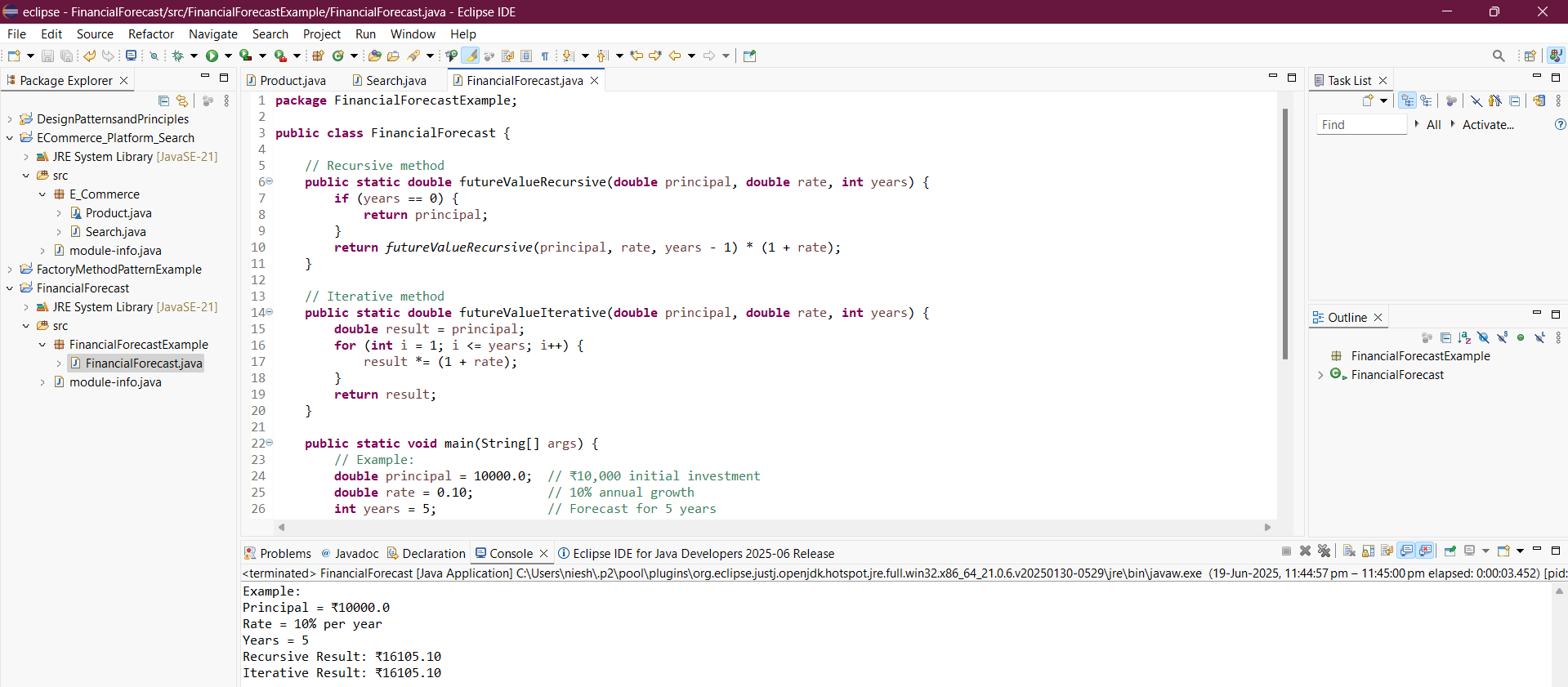
System.***out***.println("Years = " + years);

System.***out***.printf("Recursive Result: ₹%.2f\n", recursiveResult);

System.***out***.printf("Iterative Result: ₹%.2f\n", iterativeResult);

}

}



**----------------------------------------------------------------------------------------------------------------**