**ALGORITHMS AND 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.

**Product.java**

**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;**

**}**

**public String toString() {**

**return productId + " - " + productName + " (" + category + ")";**

**}**

**}**

**SearchOperations.java**

**import java.util.Arrays;**

**import java.util.Comparator;**

**public class SearchOperations {**

**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;**

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

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

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

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

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

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

**else right = mid - 1;**

**}**

**return null;**

**}**

**public static void sortProductsByName(Product[] products) {**

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

**}**

**}**

**Main.java**

**public class Main {**

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

**Product[] products = {**

**new Product(1, "Laptop", "Electronics"),**

**new Product(2, "Chair", "Furniture"),**

**new Product(3, "Shoes", "Apparel"),**

**new Product(4, "Phone", "Electronics"),**

**new Product(5, "Watch", "Accessories")**

**};**

**Product found1 = SearchOperations.*linearSearch*(products, "Shoes");**

**System.*out*.println("Linear Search Result: " + (found1 != null ? found1 : "Not Found"));**

**SearchOperations.*sortProductsByName*(products);**

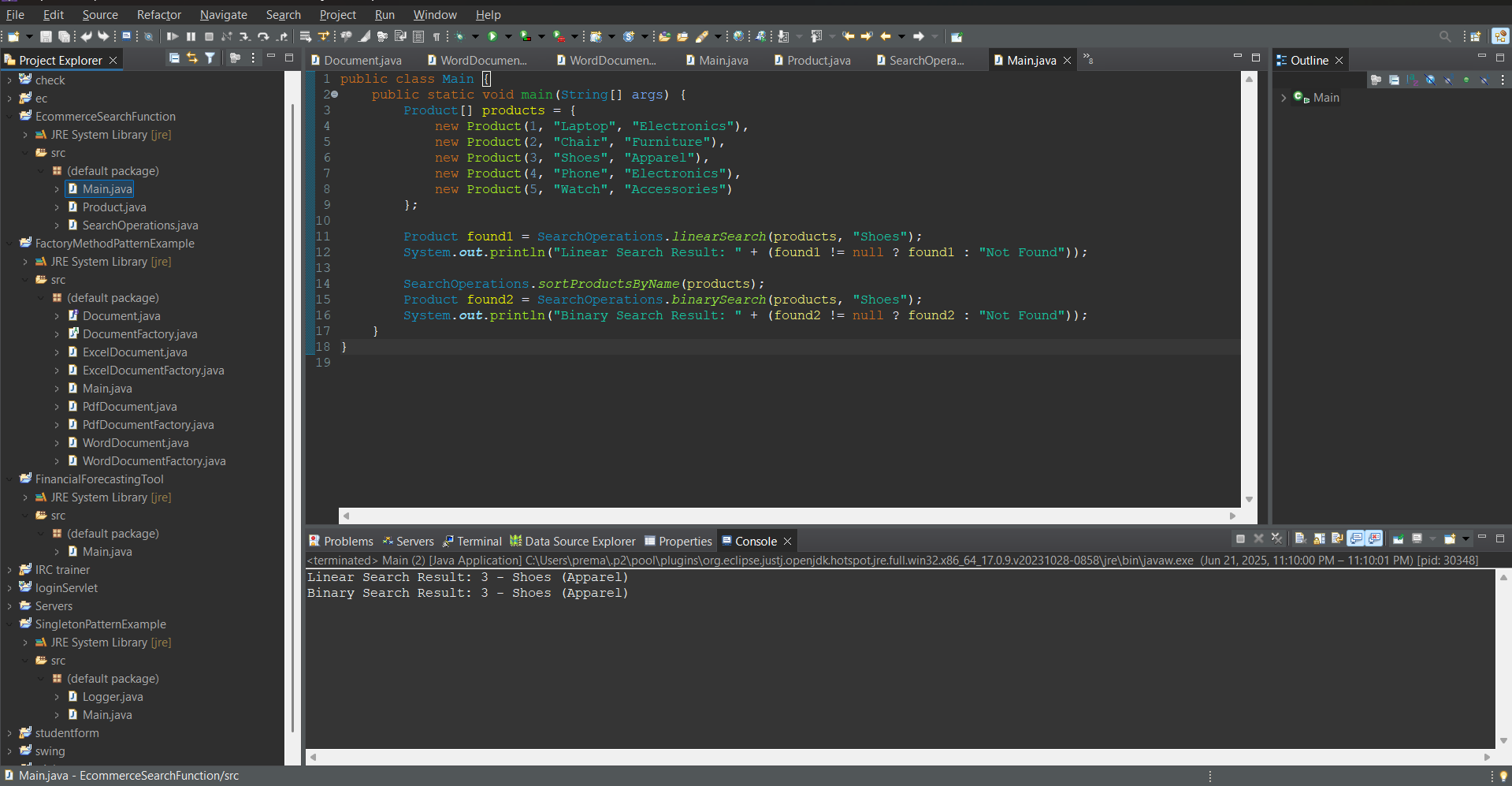
**Product found2 = SearchOperations.*binarySearch*(products, "Shoes");**

**System.*out*.println("Binary Search Result: " + (found2 != null ? found2 : "Not Found"));**

**}**

**}**

**OUTPUT**

****

**Exercise 7: Financial Forecasting**

**Scenario:**

You are developing a financial forecasting tool that predicts future values based on past data.

**Main.java**

public class Main {

public static double forecastValue(int years, double initialValue, double growthRate) {

if (years == 0) {

return initialValue;

}

return *forecastValue*(years - 1, initialValue, growthRate) \* (1 + growthRate);

}

public static void main(String[] args) {

int years = 5;

double initialValue = 10000.0;

double growthRate = 0.10;

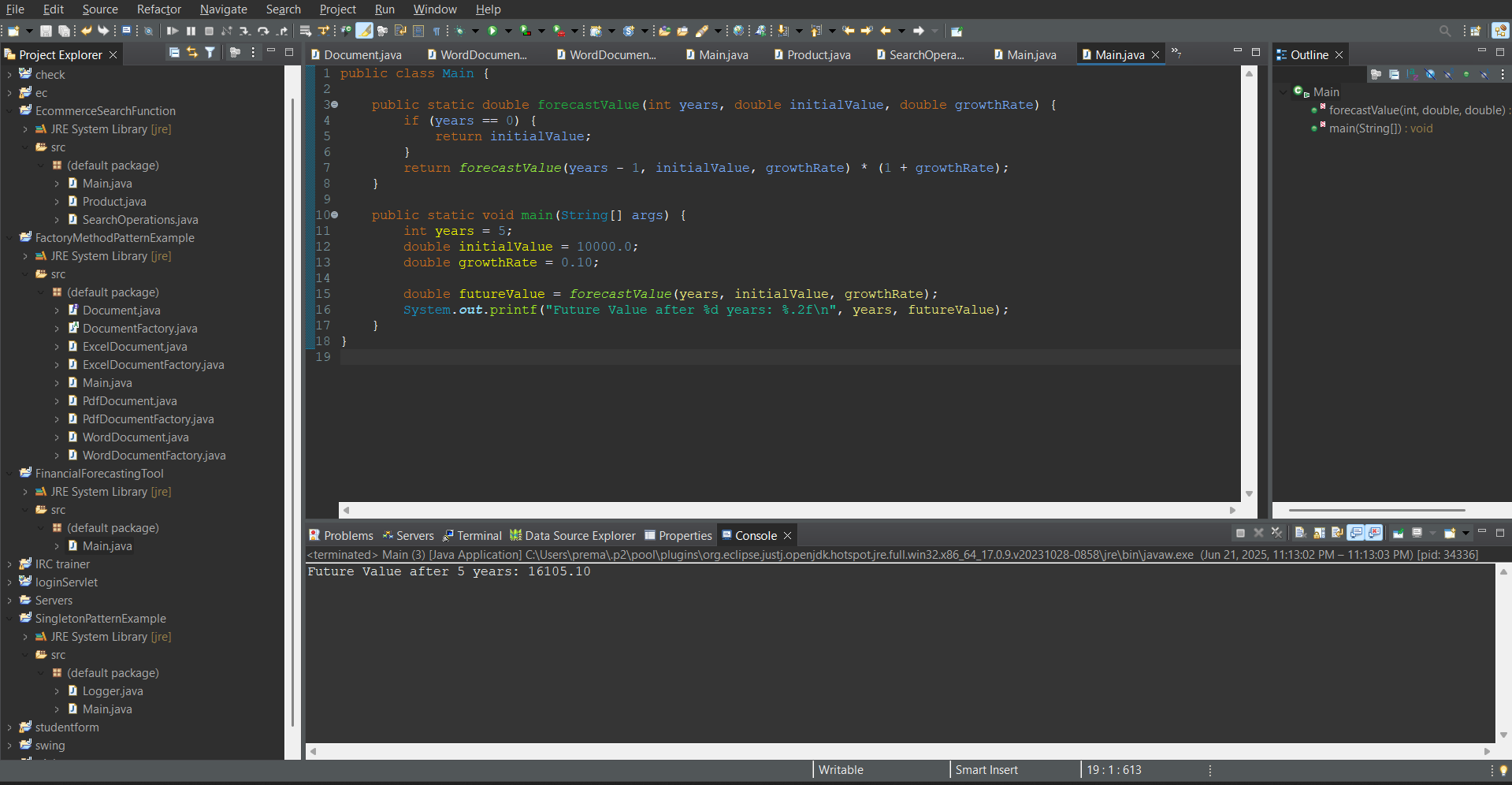
double futureValue = *forecastValue*(years, initialValue, growthRate);

System.***out***.printf("Future Value after %d years: %.2f\n", years, futureValue);

}

}

**OUTPUT**

****