**WEEK 1 : DATA STRUCTURES AND ALGORITHMS**

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

**Product.java**

public class Product {

public int productId;

public String productName;

public 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 + '\'' +

'}';

}

}

**LinearSearch.java**

public class LinearSearch {

public static Product searchByName(Product[] products, String key) {

for (Product product : products) {

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

return product;

}

}

return null;

}

}

**BinarySearch.java**

import java.util.Arrays;

import java.util.Comparator;

public class BinarySearch {

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

Arrays.sort(products, new Comparator<Product>() {

public int compare(Product p1, Product p2) {

return p1.productName.compareToIgnoreCase(p2.productName);

}

});

Product key = new Product(0, targetName, "");

int index = Arrays.binarySearch(products, key, new Comparator<Product>() {

public int compare(Product p1, Product p2) {

return p1.productName.compareToIgnoreCase(p2.productName);

}

});

return (index >= 0) ? products[index] : null;

}

}

**Main.java**

public class Main {

public static void main(String[] args) {

Product[] catalog = {

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

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

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

new Product(104, "Watch", "Accessories")

};

Product linearResult = LinearSearch.searchByName(catalog, "Watch");

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

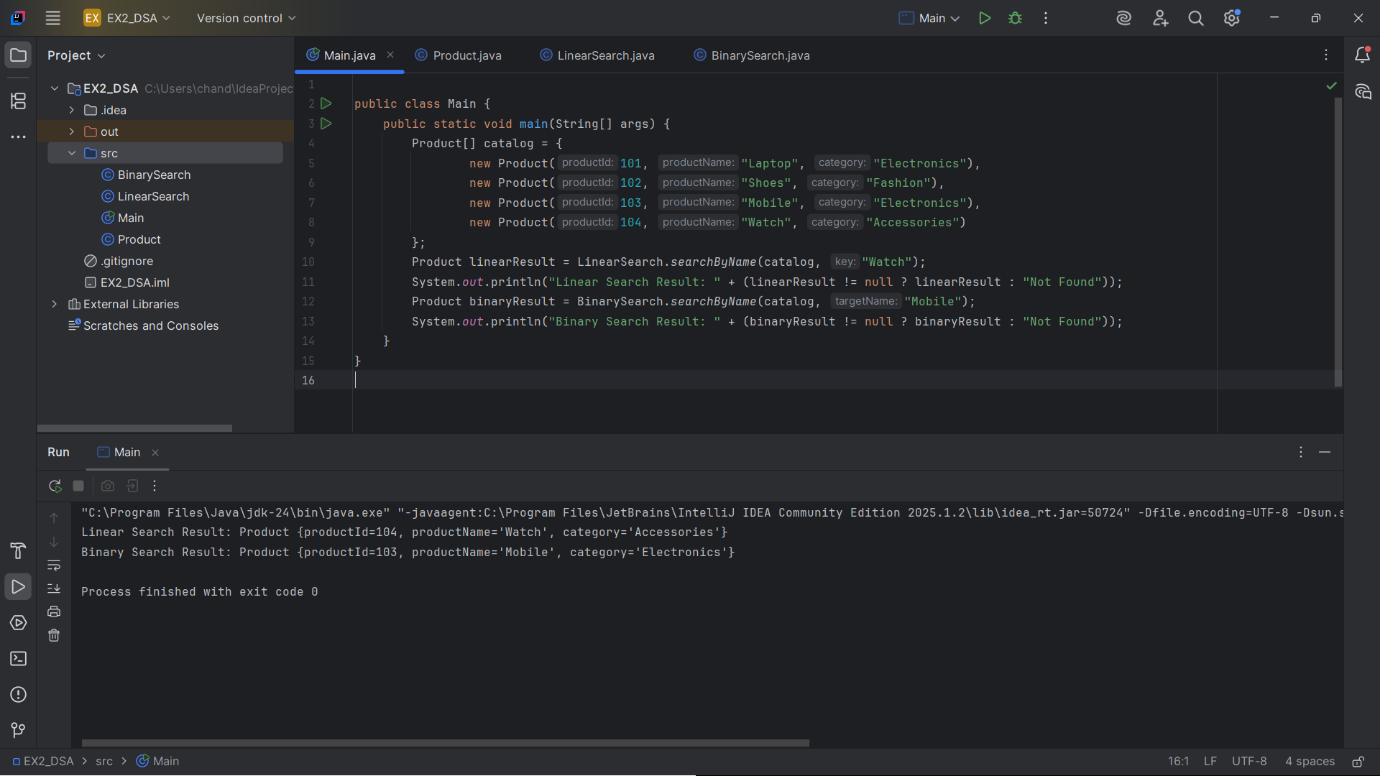
Product binaryResult = BinarySearch.searchByName(catalog, "Mobile");

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

}

}

**OUTPUT:**

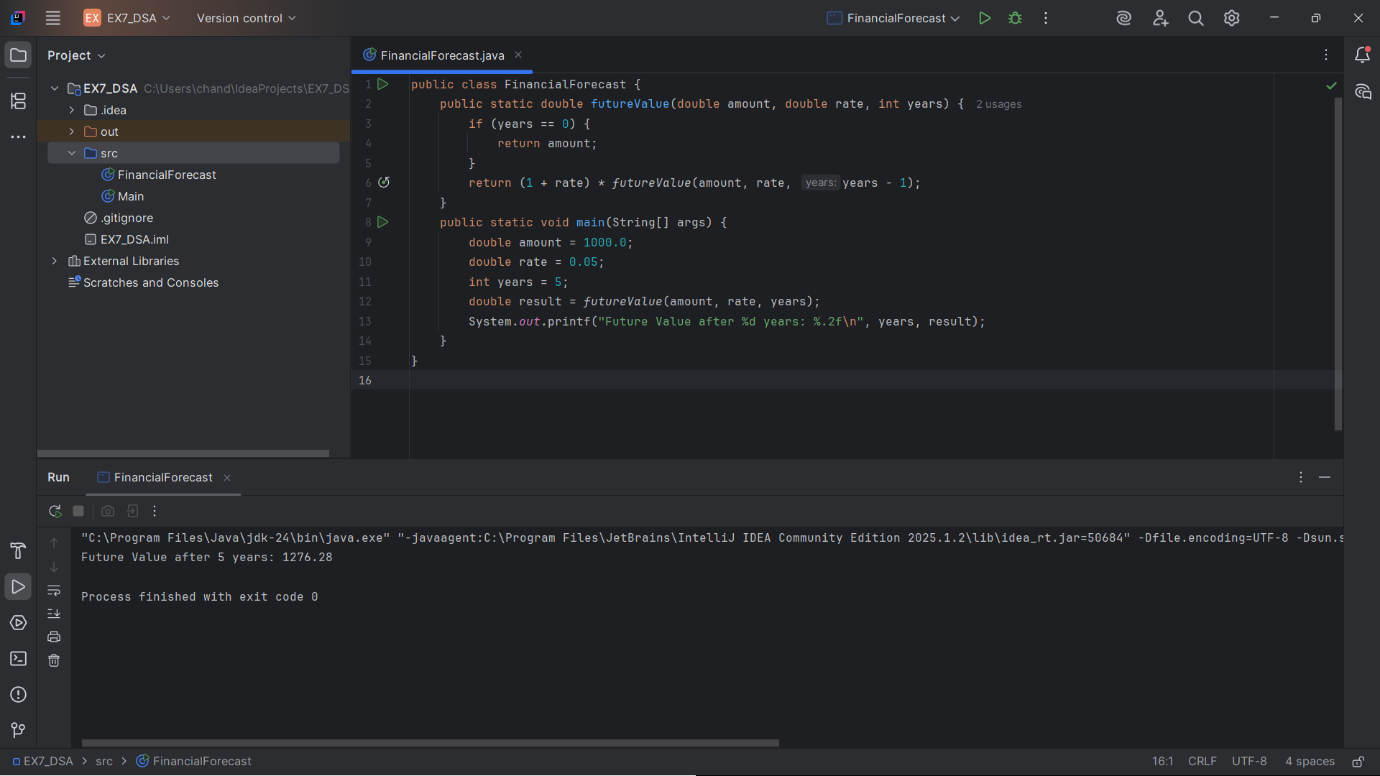


**Exercise 7: Financial Forecasting**

**FinancialForecast.java**

public class FinancialForecast {  
 public static double futureValue(double amount, double rate, int years) {  
 if (years == 0) {  
 return amount;  
 }  
 return (1 + rate) \* *futureValue*(amount, rate, years - 1);  
 }  
 public static void main(String[] args) {  
 double amount = 1000.0;  
 double rate = 0.05;  
 int years = 5;  
 double result = *futureValue*(amount, rate, years);  
 System.*out*.printf("Future Value after %d years: %.2f\n", years, result);  
 }  
}

**OUTPUT:**

****