**Data Structures and Algorithms**

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

**Package : search**

**Class 1: Product.java**

package search;

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 + "]";

}

}

**Class 2: SearchTest.java**

package search;

import java.util.Arrays;

import java.util.Comparator;

public class SearchTest {

public static void main(String[] args) {

Product[] products = {

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

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

new Product(103, "Watch", "Accessories"),

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

}

Product found1 = linearSearch(products, "Watch");

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

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

Product found2 = binarySearch(products, "Watch");

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

}

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

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

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

else right = mid - 1;

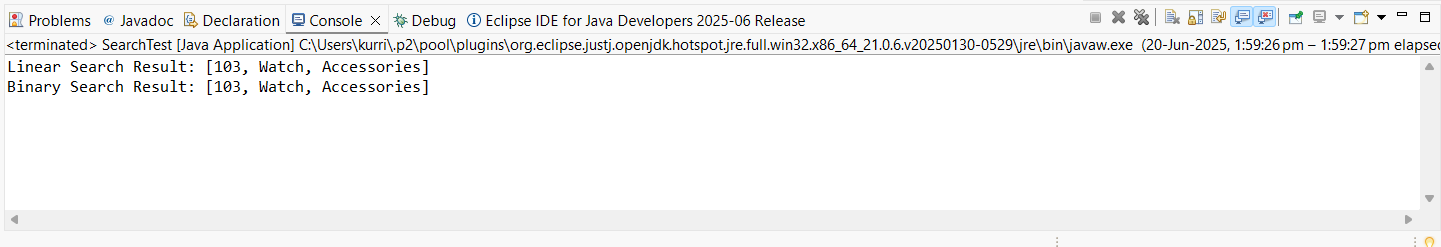
}

return null;

}

}

**Output:**

****

**Exercise 7: Financial Forecasting**

**Package: FinancialForecasting**

**Class: FinancialForecast**

package FinancialForecasting;

public class FinancialForecast {

public static double futureValue (double principal, double rate, int years) {

if (years == 0) {

return principal;

}

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

}

public static void main (String [] args) {

double initial = 10000;

double growthRate = 0.05;

int period = 5;

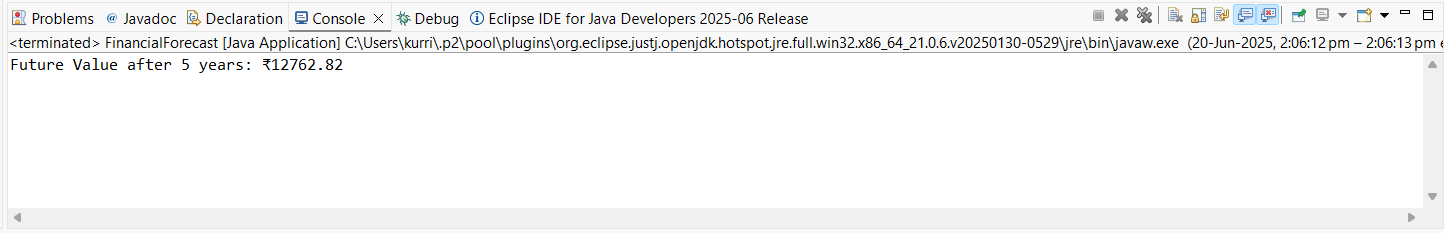
double future = *futureValue* (initial, growthRate, period);

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

}

}

**Output:**

****