**1.EcommercePlatformSearchFunction**

**Main.java**

package ecommercePlatformSearchFunction;

public class Main {

public static void main(String[] args) {

Product[] products = {

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

new Product(305, "Shirt", "Clothing"),

new Product(205, "Laptop", "Electronics"),

new Product(409, "Book", "Education"),

new Product(110, "Shoes", "Footwear")

};

Product result1 = SearchUtil.*linearSearch*(products, 205);

System.*out*.println("Linear Search Result: " + result1);

SearchUtil.*sortProductsById*(products);

Product result2 = SearchUtil.*binarySearch*(products, 205);

System.*out*.println("Binary Search Result: " + result2);

}

}

**Product.java**

ecommercePlatformSearchFunction;

public class Product {

int productId;

String productName;

String category;

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

this.productId = productId;

this.productName = productName;

package this.category = category;

}

public String toString() {

return productId + " - " + productName + " - " + category;

}

}

**Searchutil.java**

package ecommercePlatformSearchFunction;

public class SearchUtil {

public static Product linearSearch(Product[] products, int id) {

for (Product p : products) {

if (p.productId == id) return p;

}

return null;

}

public static Product binarySearch(Product[] products, int id) {

int left = 0;

int right = products.length - 1;

while (left <= right) {

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

if (products[mid].productId == id) return products[mid];

else if (products[mid].productId < id) left = mid + 1;

else right = mid - 1;

}

return null;

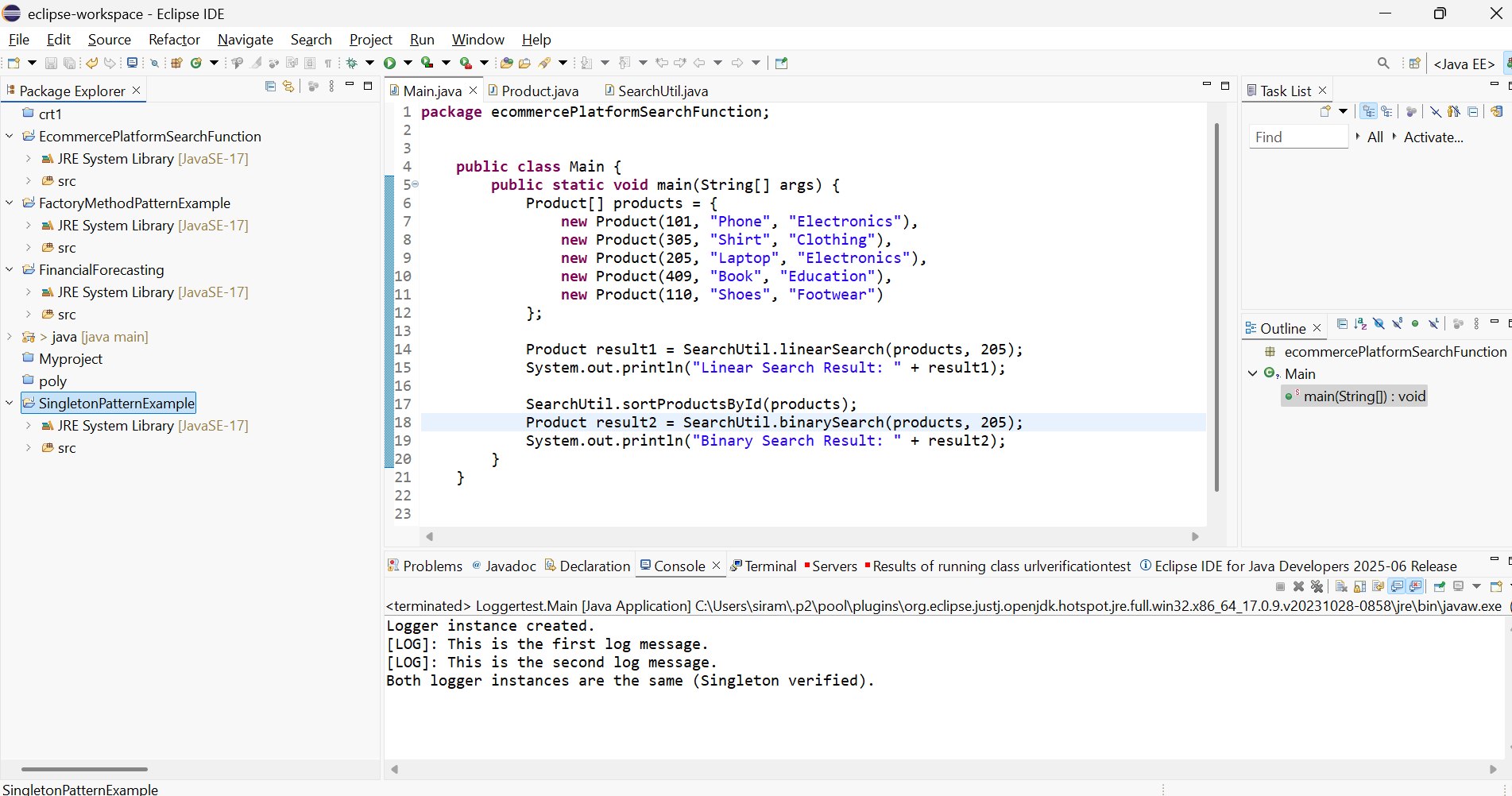
}

public static void sortProductsById(Product[] products) {

java.util.Arrays.*sort*(products, (a, b) -> a.productId - b.productId);

}

}



**2.FinancialForecasting**

**FinancialForecasting.java**

package financialForecasting;

public class FinancialForecast {

public static double predictFutureValue(double currentValue, double growthRate, int years) {

if (years == 0) return currentValue;

return *predictFutureValue*(currentValue \* (1 + growthRate), growthRate, years - 1);

}

public static void main(String[] args) {

double currentValue = 10000;

double annualGrowthRate = 0.08;

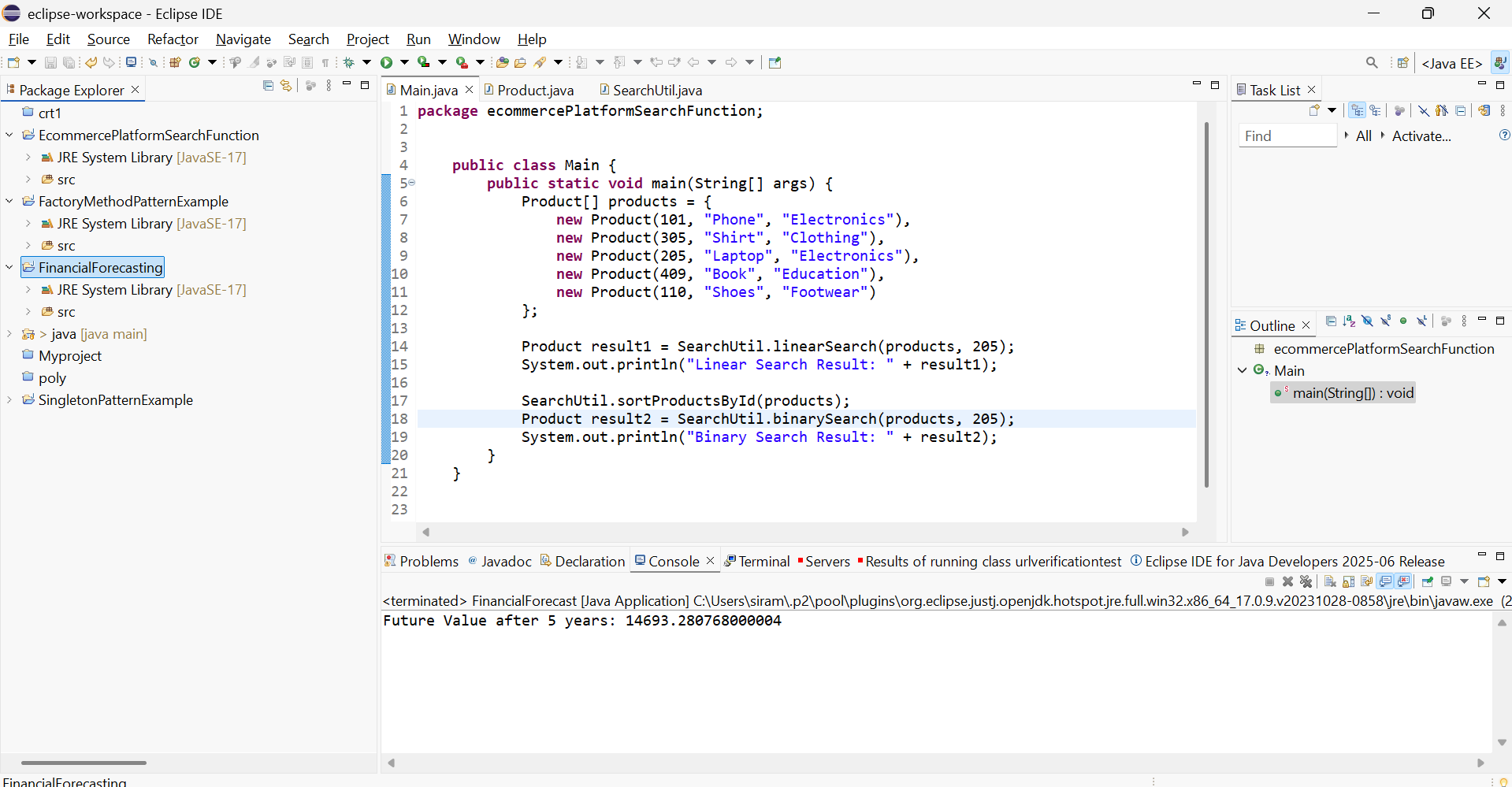
int years = 5;

double futureValue = *predictFutureValue*(currentValue, annualGrowthRate, years);

System.*out*.println("Future Value after " + years + " years: " + futureValue);

}

}

****