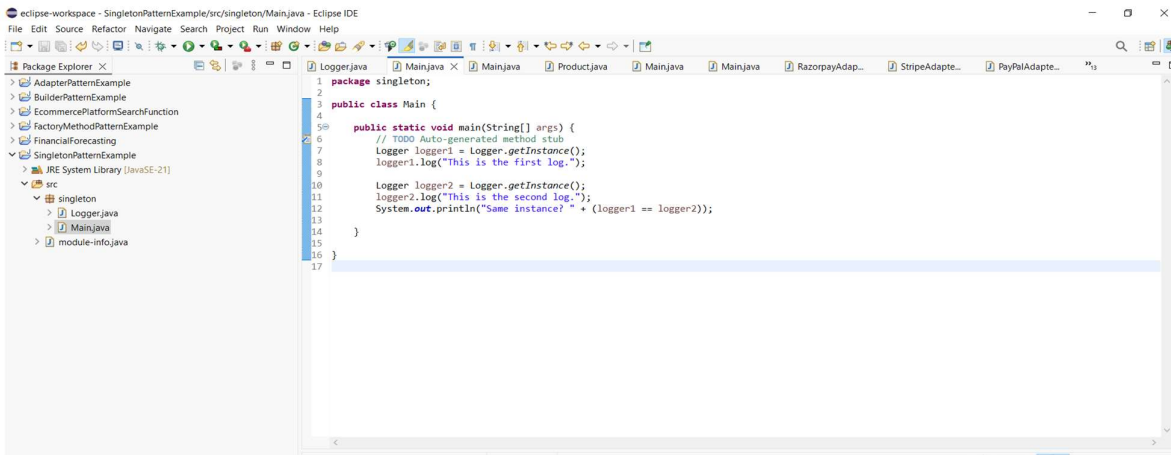
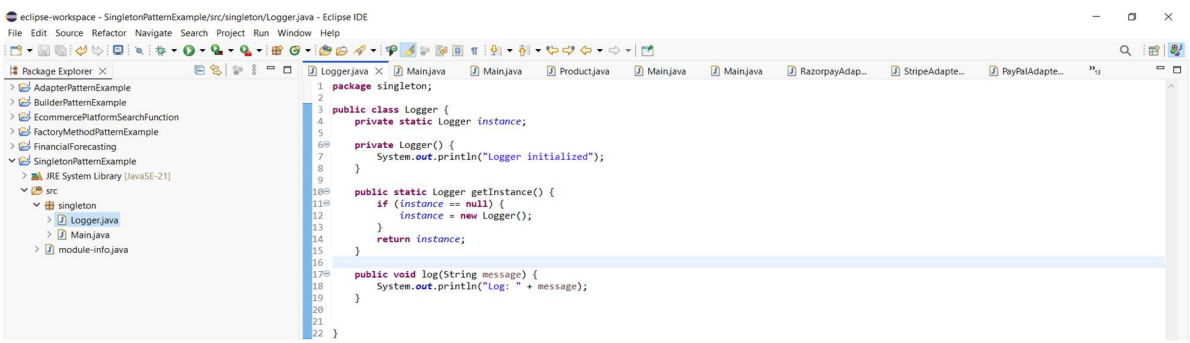
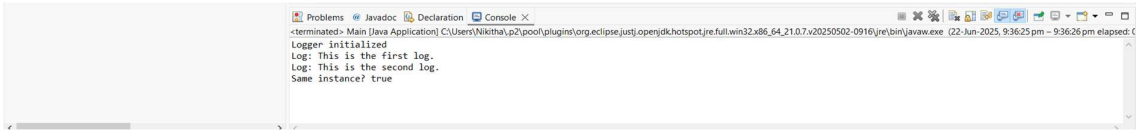
DESIGN PATTERNS AND PRINCIPLES

1) Implementing the Singleton Pattern





OUTPUT:



2) Implementing the Factory Method Pattern

eclipse-workspace - FactoryMethodPatternExample/src/factory/Main.java - Eclipse IDE

```java
package factory;

public class Main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        DocumentFactory wordFactory = new WordDocumentFactory();
        Document wordDoc = wordFactory.createDocument();
        wordDoc.open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();
        Document pdfDoc = pdfFactory.createDocument();
        pdfDoc.open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();
        Document excelDoc = excelFactory.createDocument();
        excelDoc.open();


    }

}
```

eclipse-workspace - FactoryMethodPatternExample/src/factory/Document.java - Eclipse IDE

```java
package factory;

public interface Document {

    void open();
}
```

eclipse-workspace - FactoryMethodPatternExample/src/factory/PdfDocument.java - Eclipse IDE

```java
package factory;

public class PdfDocument implements Document {
 @Override
 public void open() {
     System.out.println("Opening PDF document...");
 }
}
```

eclipse-workspace - FactoryMethodPatternExample/src/factory/ExcelDocumentFactory.java - Eclipse IDE

```java
package factory;

public class ExcelDocumentFactory extends DocumentFactory {
 @Override
 public Document createDocument() {
     return new ExcelDocument();
 }
}
```

eclipse-workspace - FactoryMethodPatternExample/src/factory/ExcelDocument.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- AdapterPatternExample
- BuilderPatternExample
- EcommercePlatformSearchFunction
- FactoryMethodPatternExample
  - JRE System Library [JavaSE-21]
  - src
    - factory
      - Document.java
      - DocumentFactory.java
      - ExcelDocument.java
      - ExcelDocumentFactory.java
      - Main.java
      - PdfDocument.java
      - PdfDocumentFactory.java
      - WordDocument.java
      - WordDocumentFactory.java
    - module-info.java
- FinancialForecasting
- SingletonPatternExample

Logger.java | Main.java | ExcelDocume... | Main.java | Document.java | Product.java | Main.java | Main.java | DocumentFac...

```java
package factory;

public class ExcelDocument implements Document {
 @Override
 public void open() {
     System.out.println("Opening Excel document...");
 }
}
```

eclipse-workspace - FactoryMethodPatternExample/src/factory/DocumentFactory.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- AdapterPatternExample
- BuilderPatternExample
- EcommercePlatformSearchFunction
- FactoryMethodPatternExample
  - JRE System Library [JavaSE-21]
  - src
    - factory
      - Document.java
      - DocumentFactory.java
      - ExcelDocument.java
      - ExcelDocumentFactory.java
      - Main.java
      - PdfDocument.java
      - PdfDocumentFactory.java
      - WordDocument.java
      - WordDocumentFactory.java
    - module-info.java
- FinancialForecasting
- SingletonPatternExample

Logger.java | Main.java | Main.java | Document.java | Product.java | Main.java | Main.java | RazorpayAdap... | DocumentFac...

```java
package factory;

public abstract class DocumentFactory {
 public abstract Document createDocument();
}
```

eclipse-workspace - FactoryMethodPatternExample/src/factory/WordDocument.java - Eclipse IDE

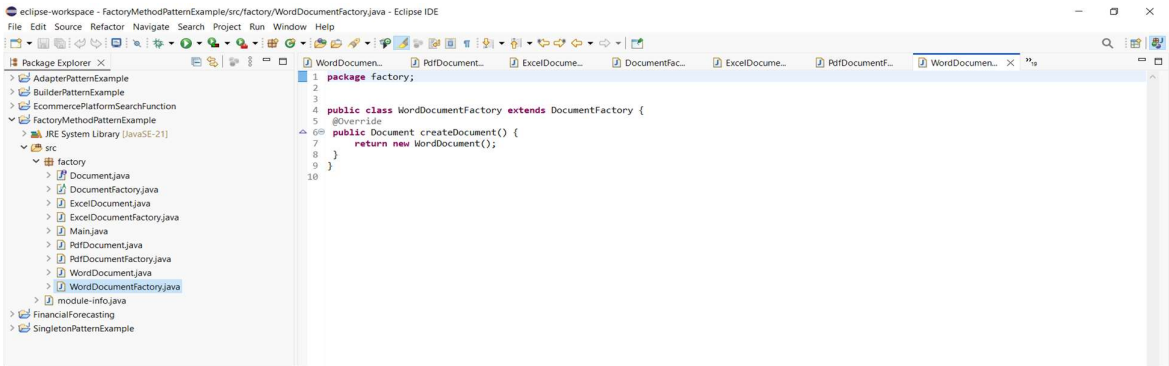
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

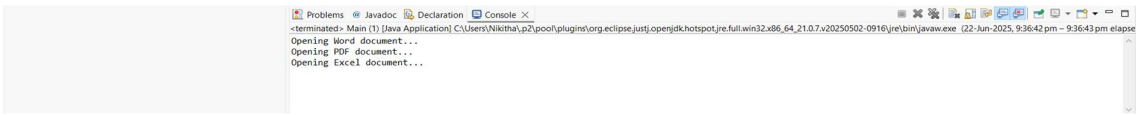
- AdapterPatternExample
- BuilderPatternExample
- EcommercePlatformSearchFunction
- FactoryMethodPatternExample
  - JRE System Library [JavaSE-21]
  - src
    - factory
      - Document.java
      - DocumentFactory.java
      - ExcelDocument.java
      - ExcelDocumentFactory.java
      - Main.java
      - PdfDocument.java
      - PdfDocumentFactory.java
      - WordDocument.java
      - WordDocumentFactory.java
    - module-info.java
- FinancialForecasting
- SingletonPatternExample

WordDocumen... | PdfDocument... | ExcelDocume... | Main.java | Document.java | DocumentFac... | ExcelDocume... | PdfDocumentF...

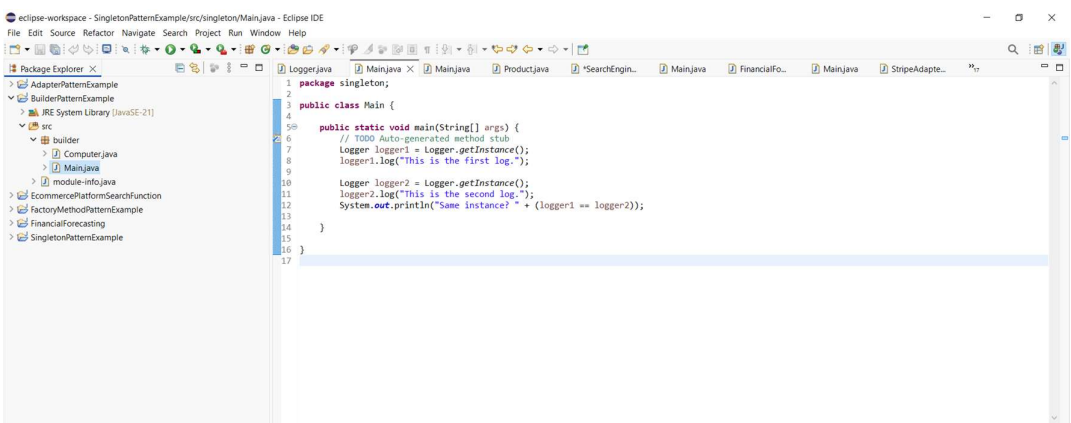
```java
package factory;

public class WordDocument implements Document{

 public void open() {
     System.out.println("Opening Word document...");
 }
}
```

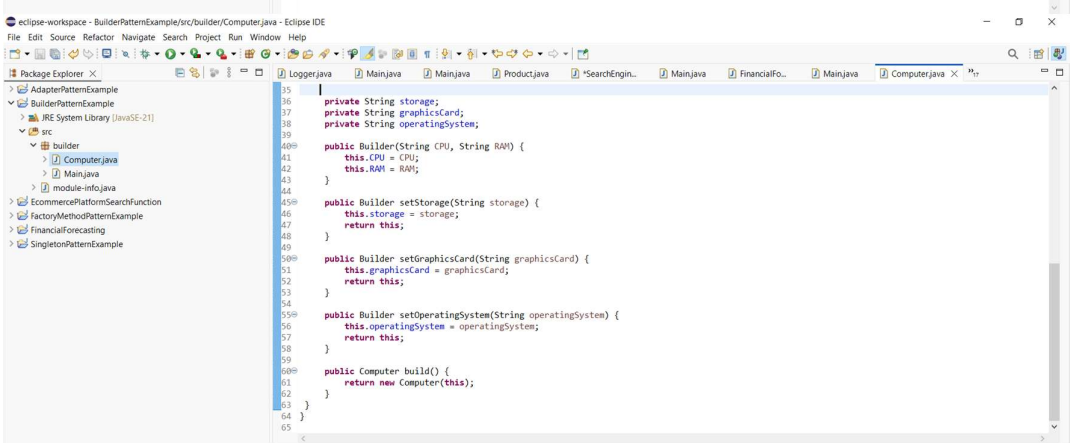


OUTPUT :



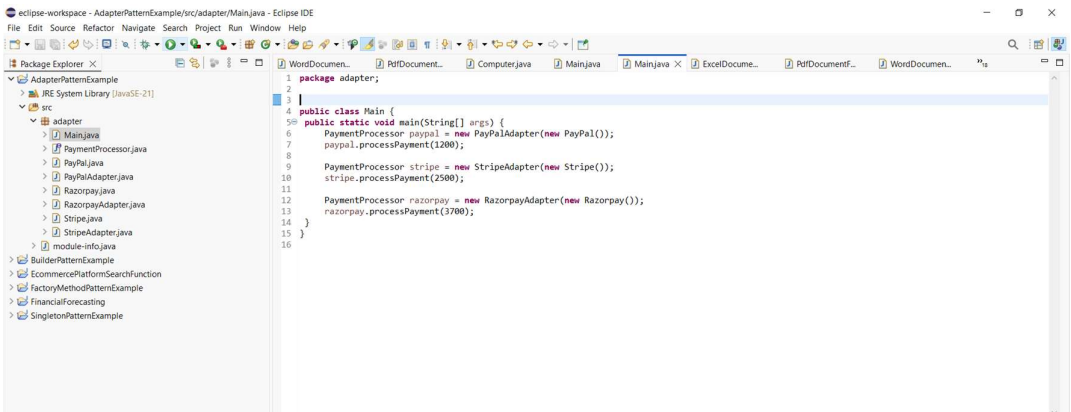
3) Implementing the Builder Pattern

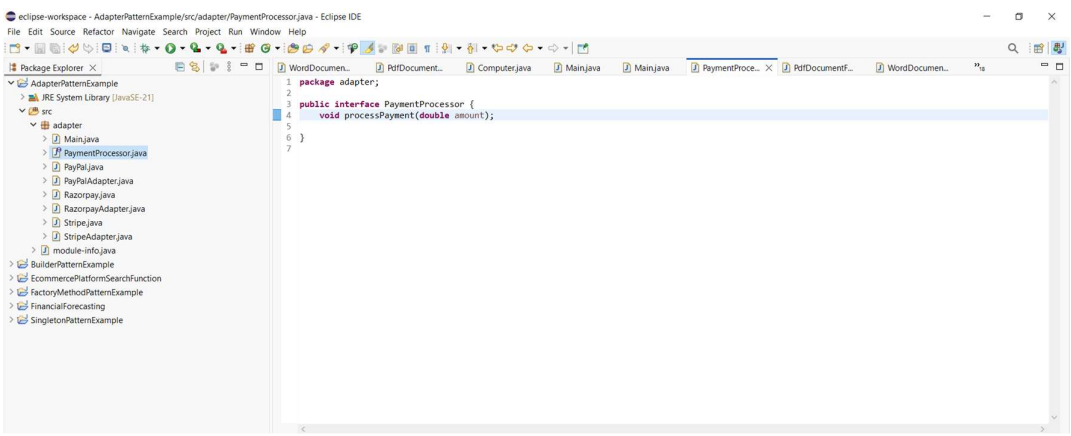


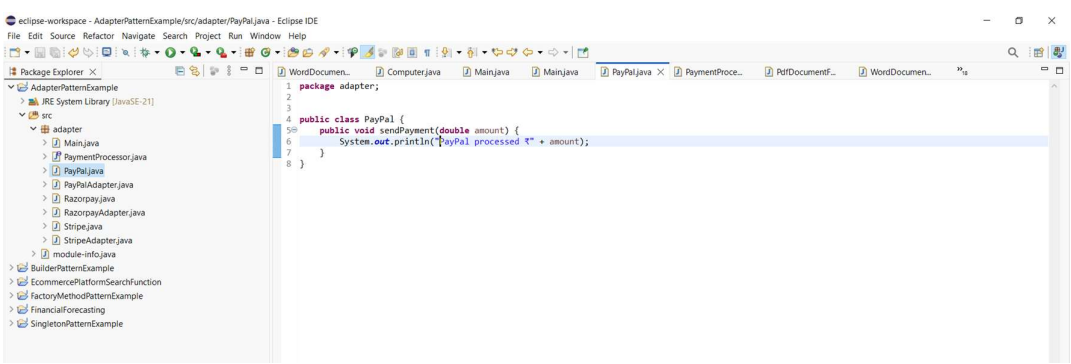


OUTPUT:

4) Implementing the Adapter Pattern







eclipse-workspace - AdapterPatternExample/src/adapter/PayPalAdapter.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- AdapterPatternExample
  - JRE System Library [JavaSE-21]
  - src
    - adapter
      - Main.java
      - PaymentProcessor.java
      - PayPal.java
      - PayPalAdapter.java
      - Razorpay.java
      - RazorpayAdapter.java
      - Stripe.java
      - StripeAdapter.java
    - module-info.java
- BuilderPatternExample
- EcommercePlatformSearchFunction
- FactoryMethodPatternExample
- FinancialForecasting
- SingletonPatternExample

WordDocumen... | Computer.java | Main.java | Main.java | PayPalAdapte... | PayPal.java | PaymentProce... | WordDocumen...

```java
package adapter;

public class PayPalAdapter implements PaymentProcessor {
 private PayPal payPal;

 public PayPalAdapter(PayPal payPal) {
     this.payPal = payPal;
 }

 @Override
 public void processPayment(double amount) {
     payPal.sendPayment(amount);
 }
}
```

eclipse-workspace - AdapterPatternExample/src/adapter/Stripe.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- AdapterPatternExample
  - JRE System Library [JavaSE-21]
  - src
    - adapter
      - Main.java
      - PaymentProcessor.java
      - PayPal.java
      - PayPalAdapter.java
      - Razorpay.java
      - RazorpayAdapter.java
      - Stripe.java
      - StripeAdapter.java
    - module-info.java
- BuilderPatternExample
- EcommercePlatformSearchFunction
- FactoryMethodPatternExample
- FinancialForecasting
- SingletonPatternExample

Computer.java | Main.java | Main.java | RazorpayAdap... | PayPalAdapte... | Razorpay.java | PayPal.java | Stripe.java | PaymentProce...

```java
package adapter;


public class Stripe {
 public void makeStripePayment(double amount) {
     System.out.println("Stripe charged ₹" + amount);
 }
}
```

eclipse-workspace - AdapterPatternExample/src/adapter/StripeAdapter.java - Eclipse IDE

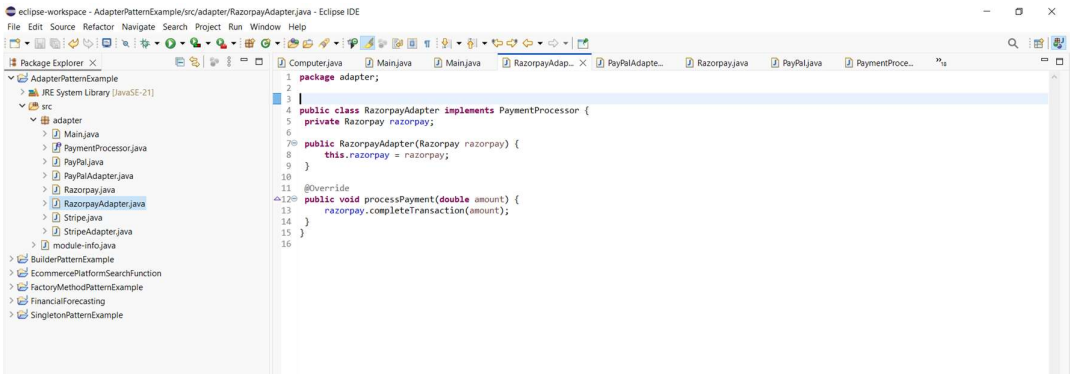
File Edit Source Refactor Navigate Search Project Run Window Help

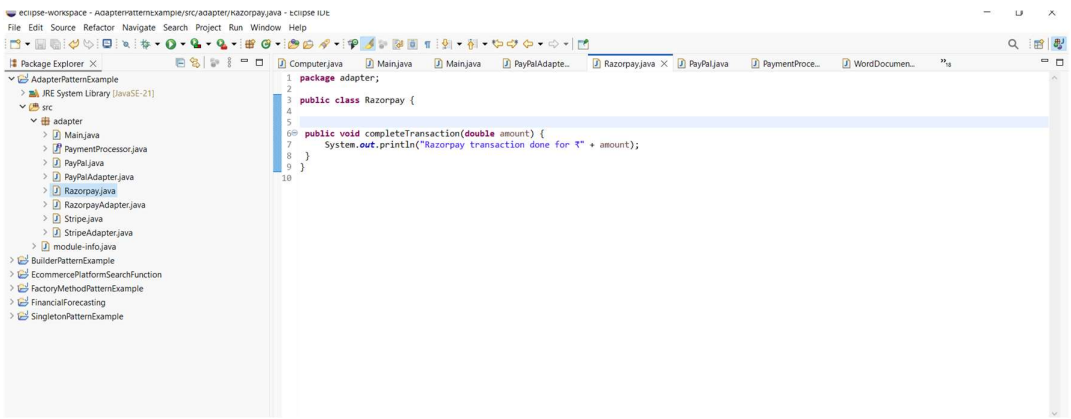
Package Explorer

- AdapterPatternExample
  - JRE System Library [JavaSE-21]
  - src
    - adapter
      - Main.java
      - PaymentProcessor.java
      - PayPal.java
      - PayPalAdapter.java
      - Razorpay.java
      - RazorpayAdapter.java
      - Stripe.java
      - StripeAdapter.java
    - module-info.java
- BuilderPatternExample
- EcommercePlatformSearchFunction
- FactoryMethodPatternExample
- FinancialForecasting
- SingletonPatternExample

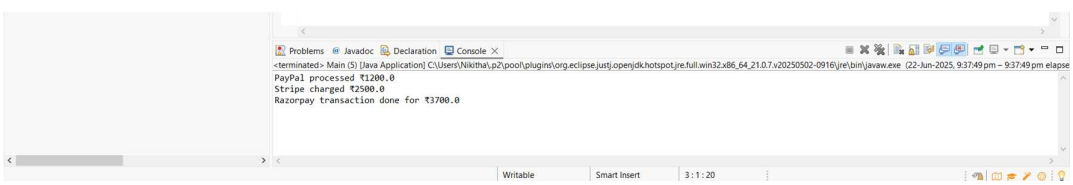
Main.java | RazorpayAdap... | StripeAdapte... | PayPalAdapte... | Razorpay.java | PayPal.java | Stripe.java | PaymentProce...

```java
package adapter;

public class StripeAdapter implements PaymentProcessor {
 private Stripe stripe;

 public StripeAdapter(Stripe stripe) {
     this.stripe = stripe;
 }

 @Override
 public void processPayment(double amount) {
     stripe.makeStripePayment(amount);
 }
}
```





OUTPUT:



ALGORITHMS AND DATA STRUCTURES

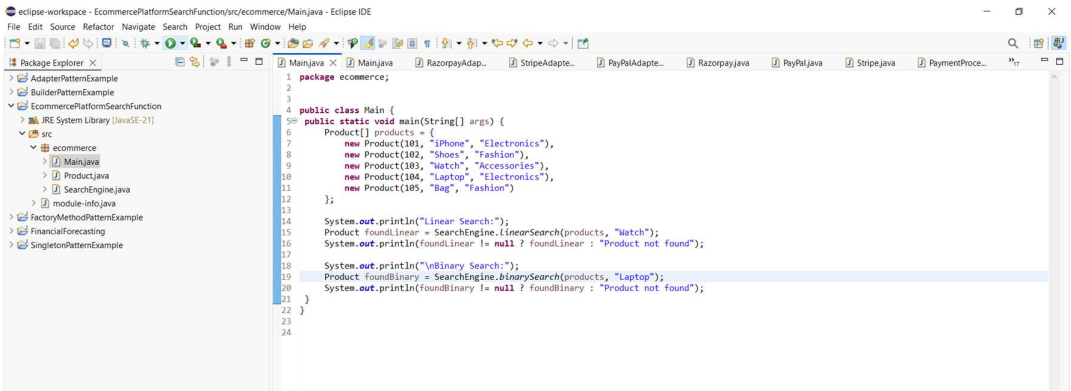
Exercise 2: E-commerce Platform Search Function

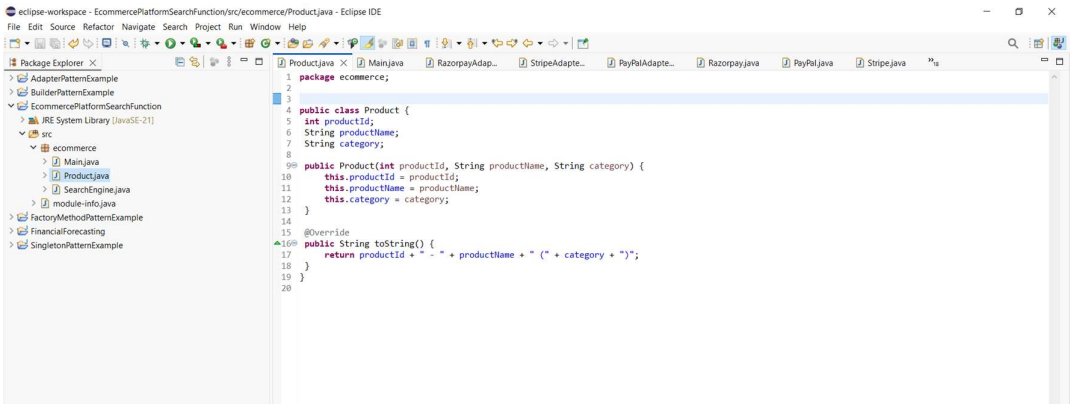
Big O notation is used to describe the performance of an algorithm as the input size increases. It helps in understanding how the algorithm scales.

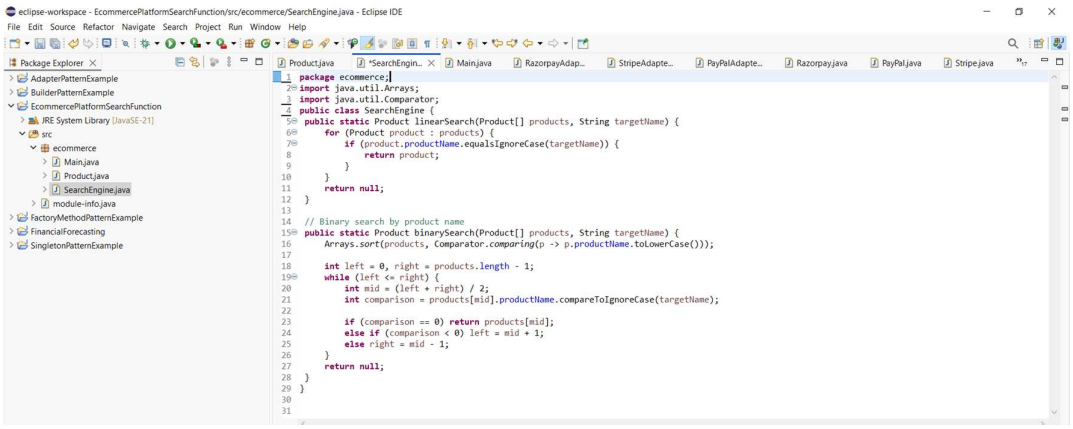
- Best Case: The minimum time required (e.g., item is at the beginning).

- Average Case: Expected time for typical inputs.

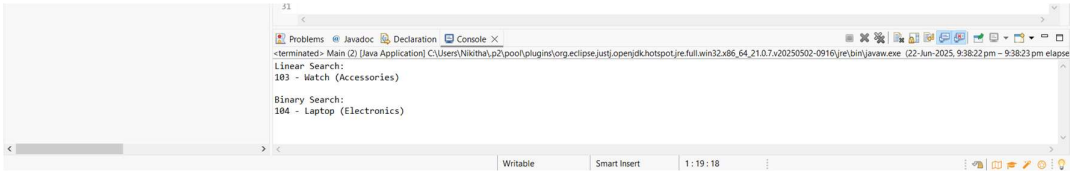
- Worst Case: Maximum time required (e.g., item not found or at the end).







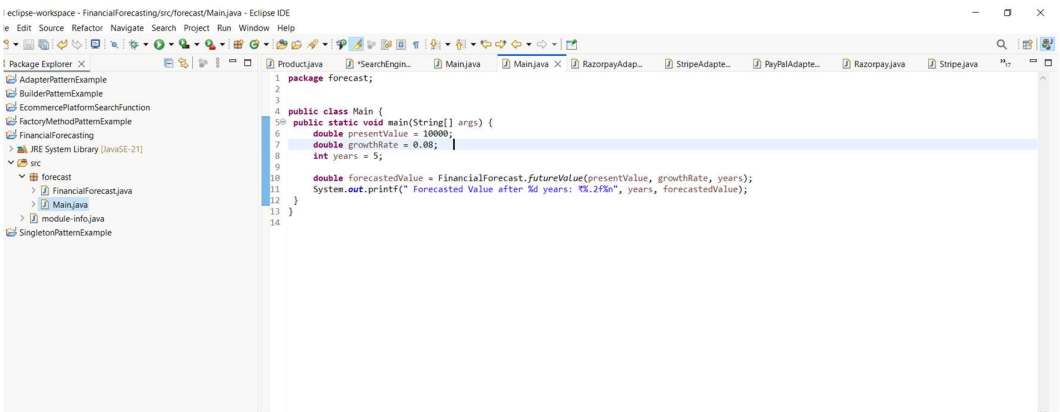
OUTPUT:

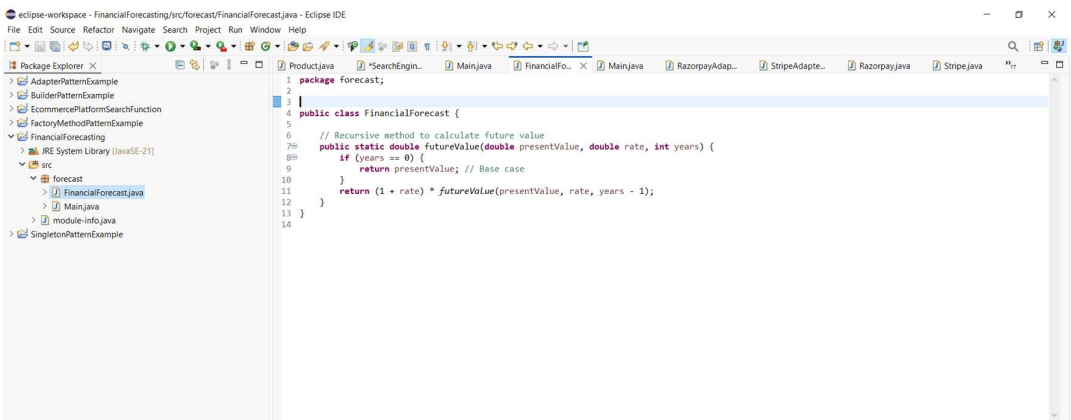


Exercise 7: Financial Forecasting

Recursion is a technique where a function calls itself to solve a problem. It is particularly useful in scenarios where a problem can be divided into subproblems of the same type.

Example: Predicting future value based on compound growth — where each year's value depends on the previous year's.





OUTPUT: