**DESIGN PRINCIPLES AND PATTERNS**

**Exercise 1: Implementing the Singleton Pattern**

**Scenario:**

You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging.

**Logger.java**

**public class Logger {**

**private static Logger *instance*;**

**private Logger() {**

**System.*out*.println("Logger initialized.");**

**}**

**public static Logger getInstance() {**

**if (*instance* == null) {**

***instance* = new Logger();**

**}**

**return *instance*;**

**}**

**public void log(String message) {**

**System.*out*.println("Log: " + message);**

**}**

**}**

**Main.java**

**public class Main {**

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

**Logger logger1 = Logger.*getInstance*();**

**Logger logger2 = Logger.*getInstance*();**

**logger1.log("First message");**

**logger2.log("Second message");**

**if (logger1 == logger2) {**

**System.*out*.println("Only one instance of Logger exists.");**

**} else {**

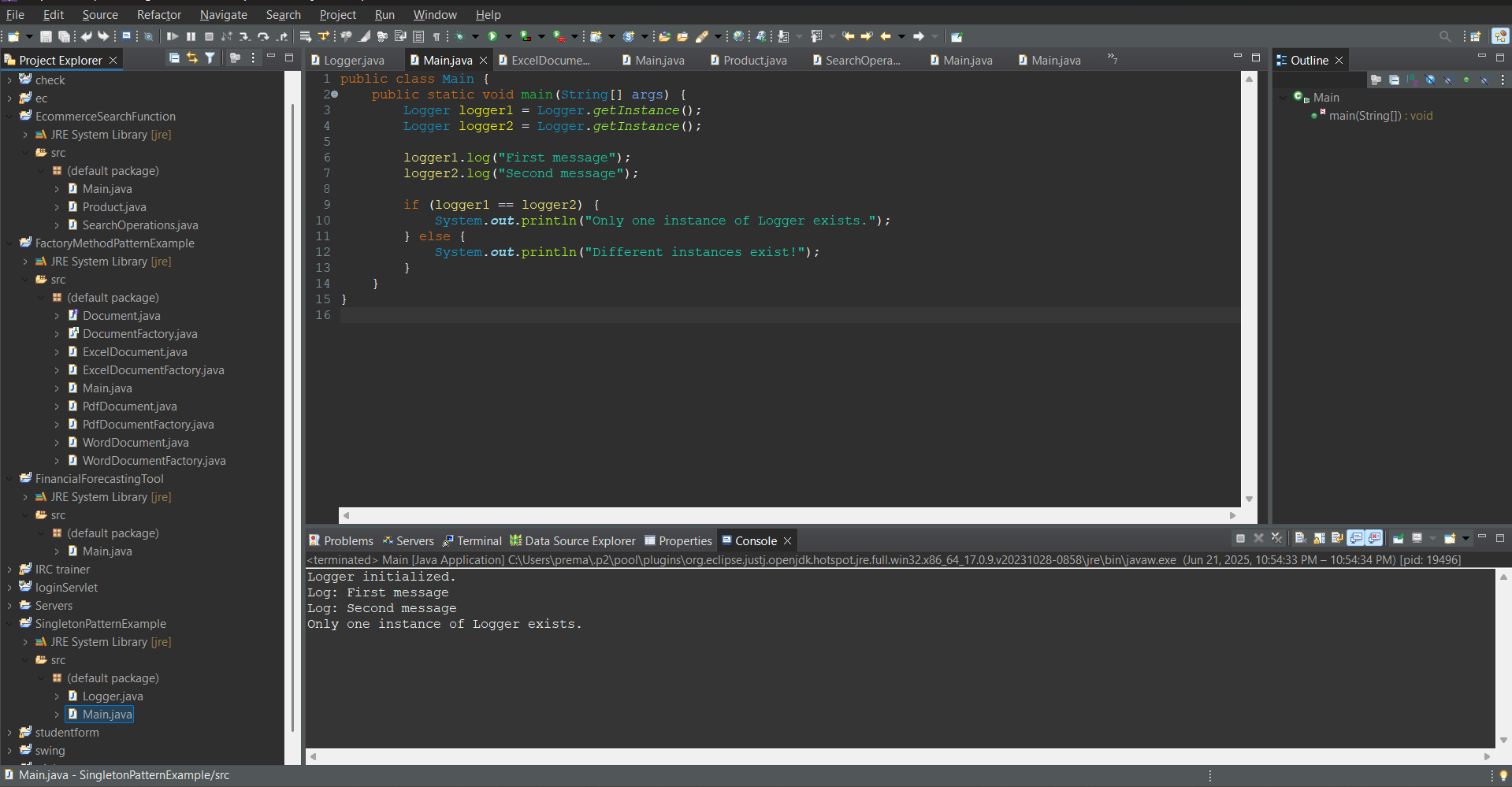
**System.*out*.println("Different instances exist!");**

**}**

**}**

**}**

**OUTPUT**

****

**Exercise 2: Implementing the Factory Method Pattern**

**Scenario:**

You are developing a document management system that needs to create different types of documents (e.g., Word, PDF, Excel). Use the Factory Method Pattern to achieve this.

**Document.java**

**public interface Document {**

**void open();**

**}**

**DocumentFactory.java**

**public abstract class DocumentFactory {**

**public abstract Document createDocument();**

**}**

**ExcelDocument.java**

**public class ExcelDocument implements Document {**

**public void open() {**

**System.*out*.println("Opening Excel document...");**

**}**

**}**

**ExcelDocumentFactory.java**

**public class ExcelDocumentFactory extends DocumentFactory {**

***@Override***

**public Document createDocument() {**

**return new ExcelDocument();**

**}**

**}**

**PdfDocument.java**

**public class PdfDocument implements Document {**

**public void open() {**

**System.*out*.println("Opening PDF document...");**

**}**

**}**

**PdfDocumentFactory.java**

**public class PdfDocumentFactory extends DocumentFactory {**

***@Override***

**public Document createDocument() {**

**return new PdfDocument();**

**}**

**}**

**WordDocument.java**

**public class WordDocument implements Document {**

**public void open() {**

**System.*out*.println("Opening Word document...");**

**}**

**}**

**WordDocumentFactory.java**

**public class WordDocumentFactory extends DocumentFactory {**

***@Override***

**public Document createDocument() {**

**return new WordDocument();**

**}**

**}**

**Main.java**

**public class Main {**

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

**DocumentFactory wordFactory = new WordDocumentFactory();**

**Document word = wordFactory.createDocument();**

**word.open();**

**DocumentFactory pdfFactory = new PdfDocumentFactory();**

**Document pdf = pdfFactory.createDocument();**

**pdf.open();**

**DocumentFactory excelFactory = new ExcelDocumentFactory();**

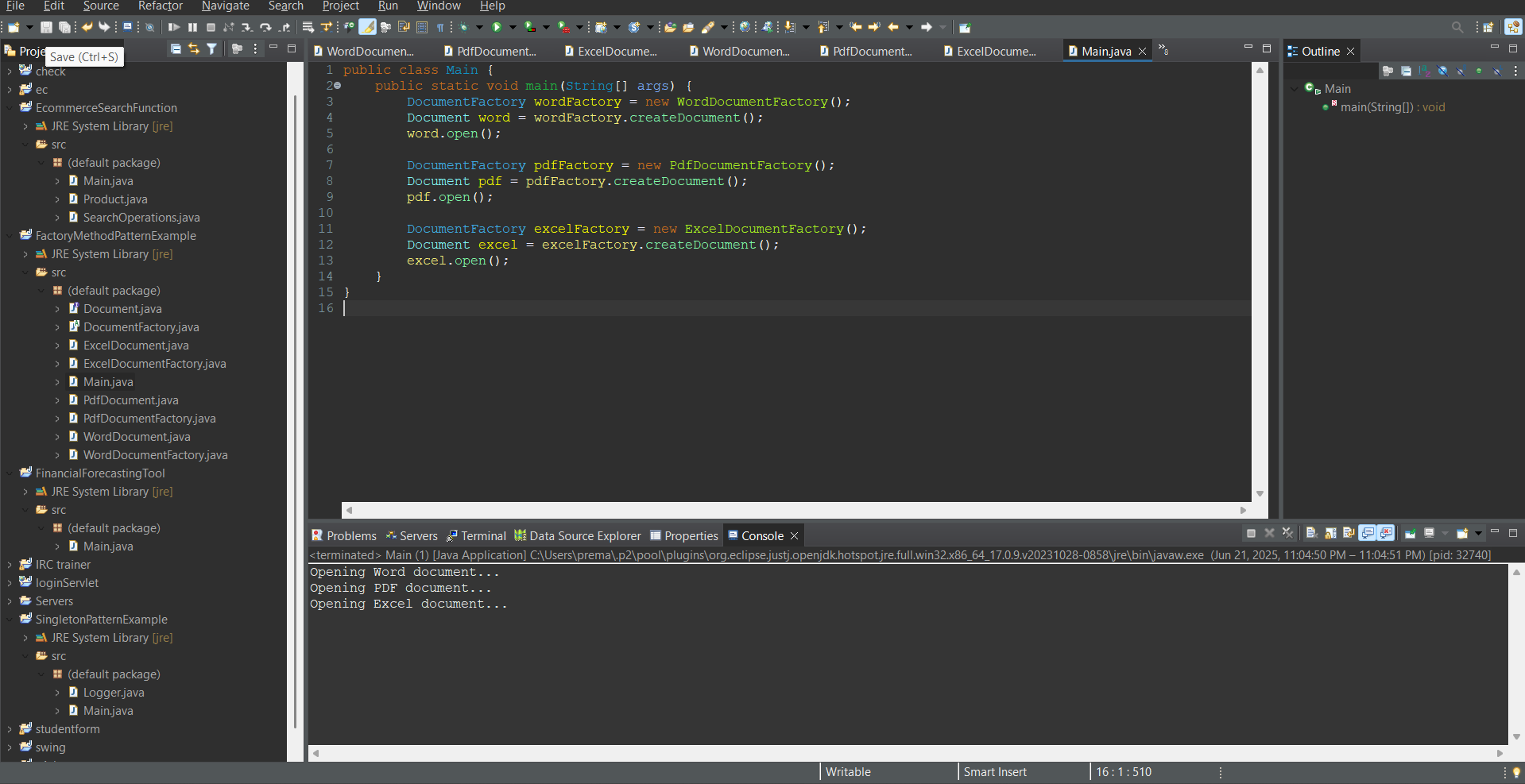
**Document excel = excelFactory.createDocument();**

**excel.open();**

**}**

**}**

**OUTPUT**

****