**DESIGN PATTERNS AND PRINCIPLES**

**Question 1**

**Exercise 1: Implementing the Singleton Pattern**

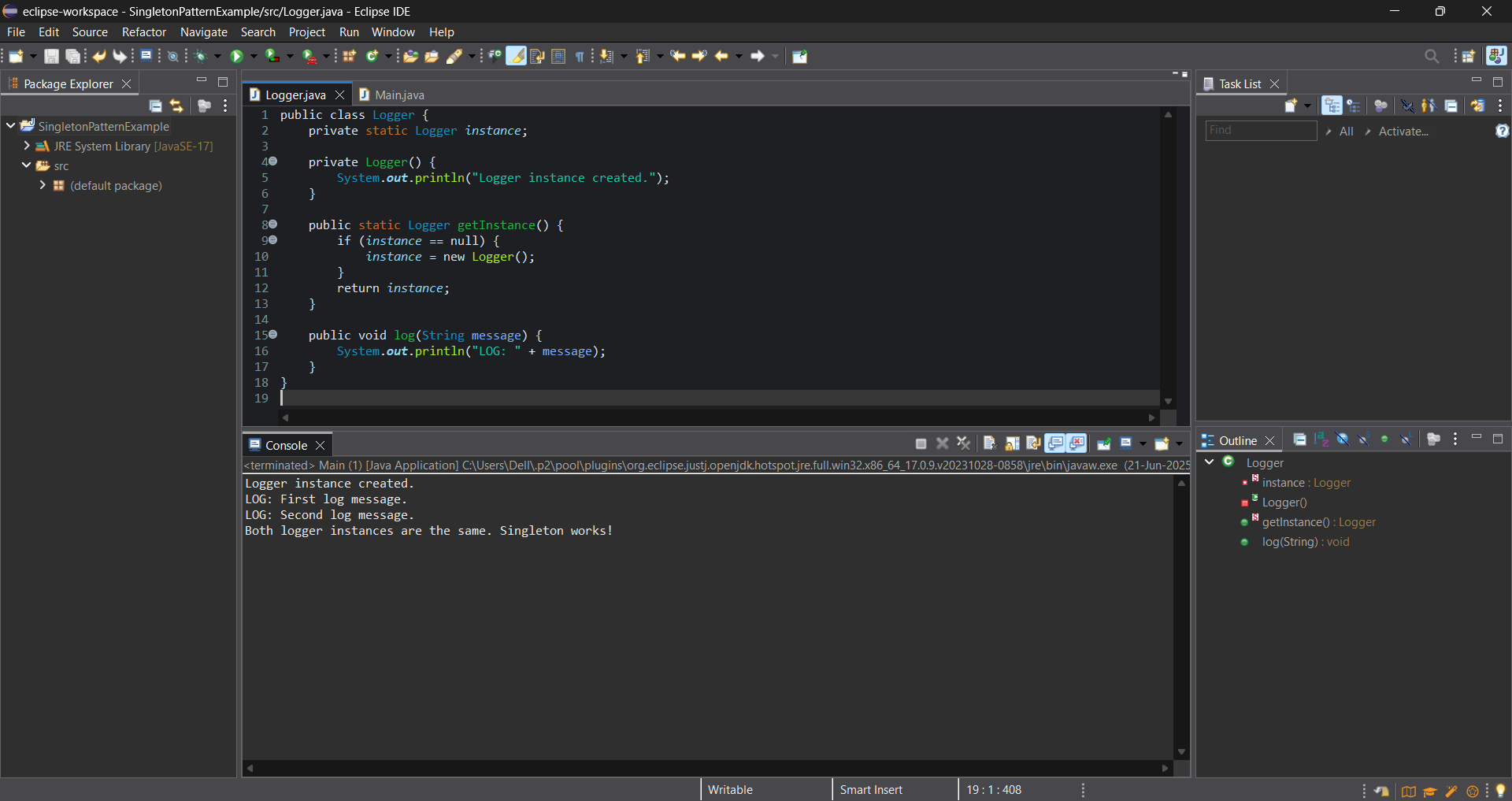
**Scenario:**

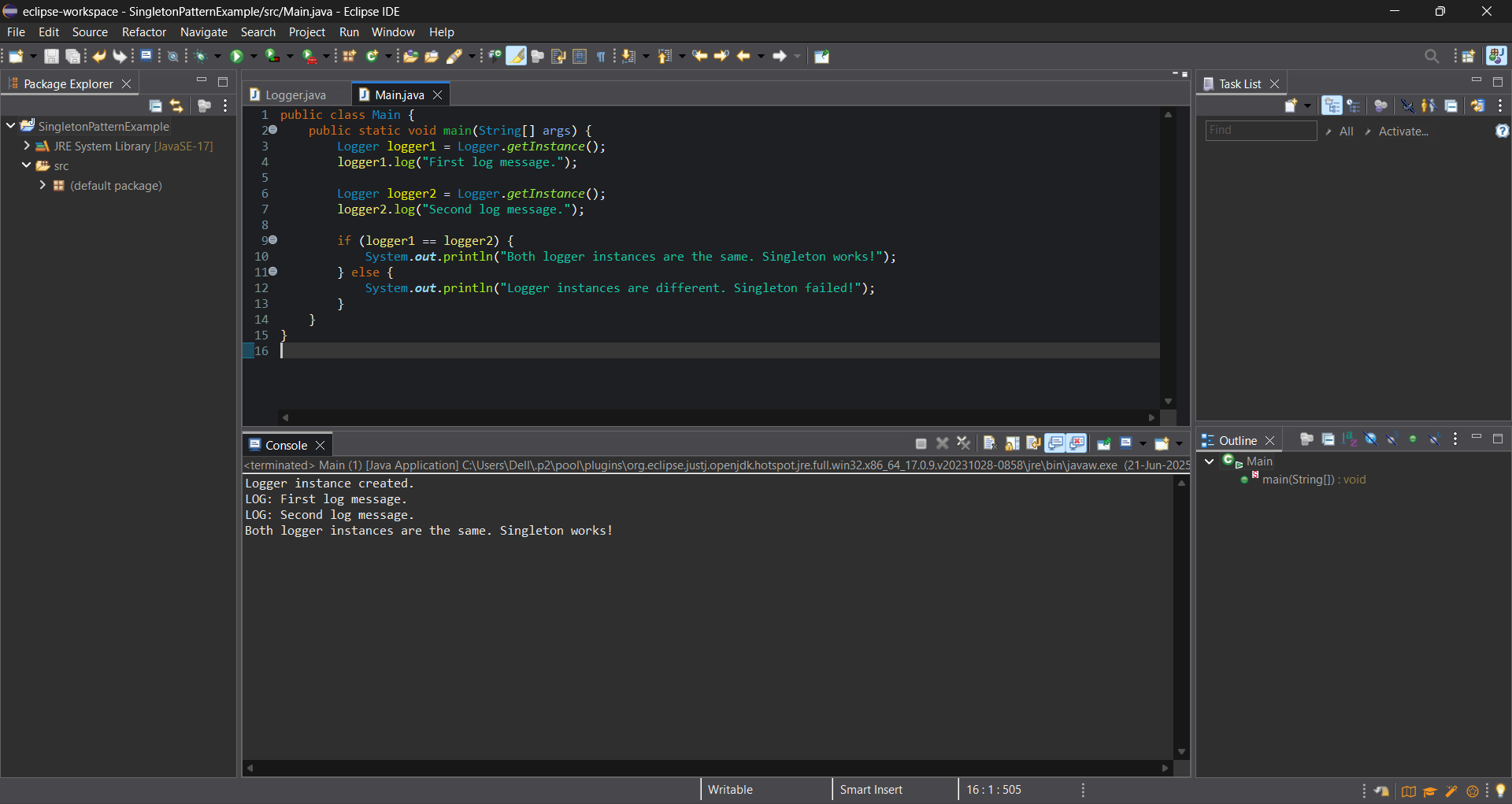
You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging.

**Steps:**

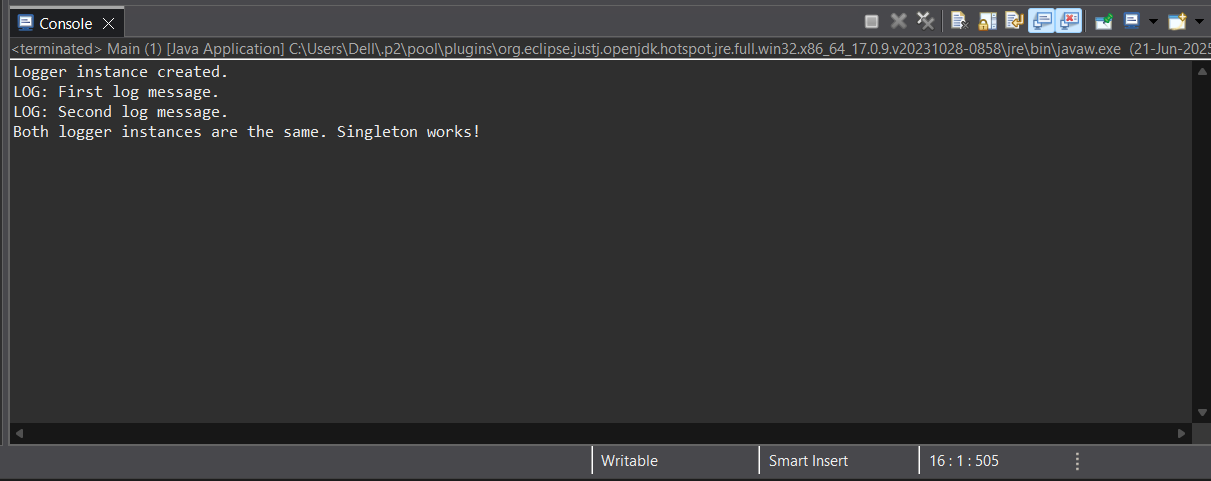
* **Create a New Java Project:**
  + Create a new Java project named **SingletonPatternExample**.
* **Define a Singleton Class:**
  + Create a class named Logger that has a private static instance of itself.
  + Ensure the constructor of Logger is private.
  + Provide a public static method to get the instance of the Logger class.
* **Implement the Singleton Pattern:**
  + Write code to ensure that the Logger class follows the Singleton design pattern.
* **Test the Singleton Implementation:**
  + Create a test class to verify that only one instance of Logger is created and used across the application.

**CODE:**





**OUTPUT:**

****

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

**Scenario:**

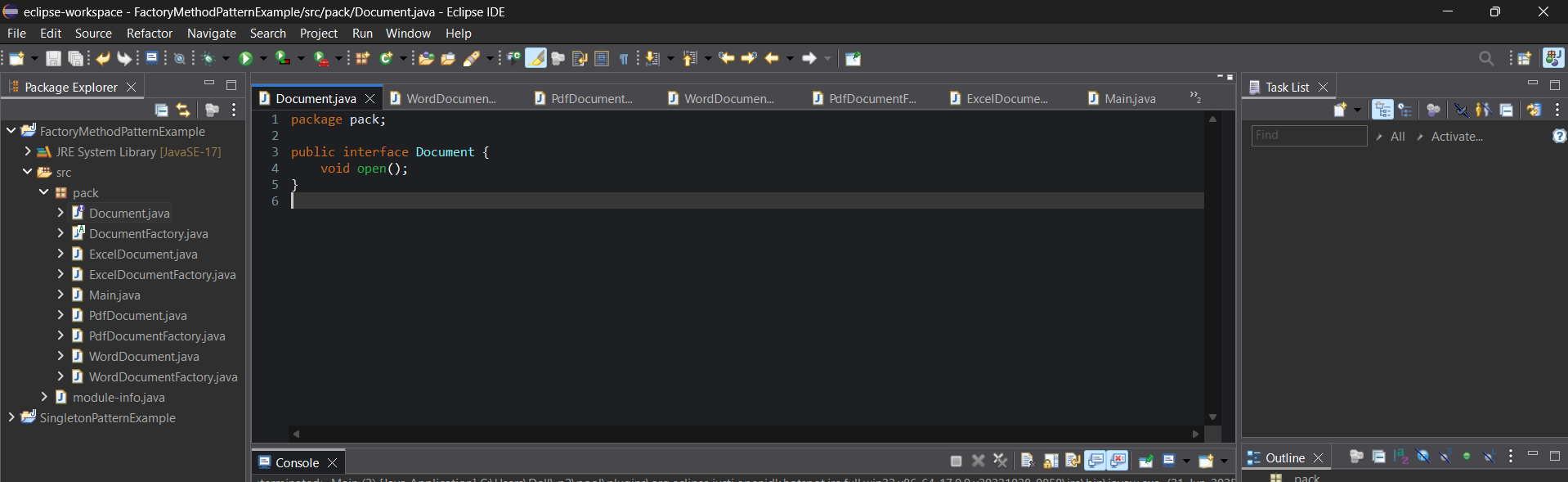
You are developing a document management system that needs to create different types of documents (e.g., Word, PDF, Excel). Use the Factory Method Pattern to achieve this.

**Steps:**

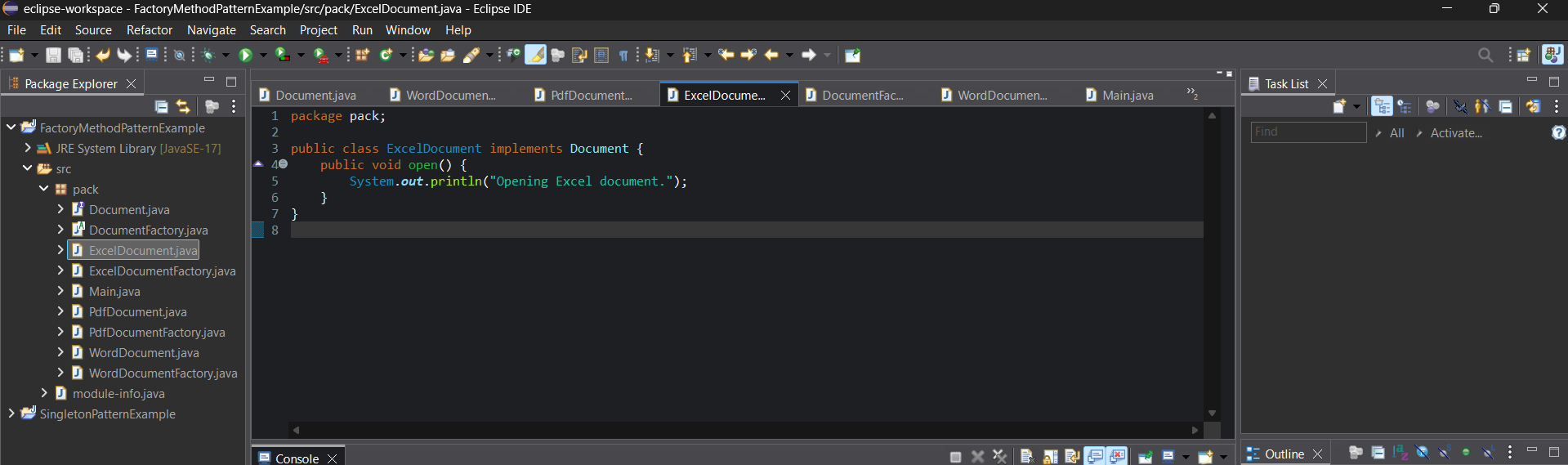
* **Create a New Java Project:**
  + Create a new Java project named **FactoryMethodPatternExample**.
* **Define Document Classes:**
  + Create interfaces or abstract classes for different document types such as **WordDocument**, **PdfDocument**, and **ExcelDocument**.
* **Create Concrete Document Classes:**
  + Implement concrete classes for each document type that implements or extends the above interfaces or abstract classes.
* **Implement the Factory Method:**
  + Create an abstract class **DocumentFactory** with a method **createDocument()**.
  + Create concrete factory classes for each document type that extends DocumentFactory and implements the **createDocument()** method.
* **Test the Factory Method Implementation:**
  + Create a test class to demonstrate the creation of different document types using the factory method.

**CODE:**

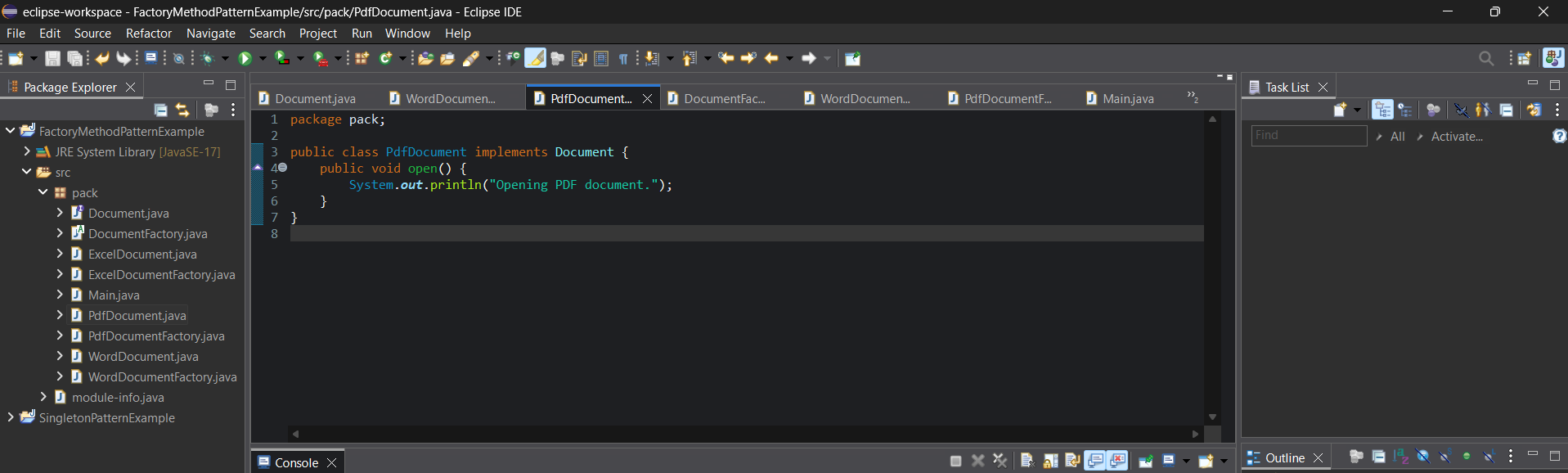
**Document.java**

****

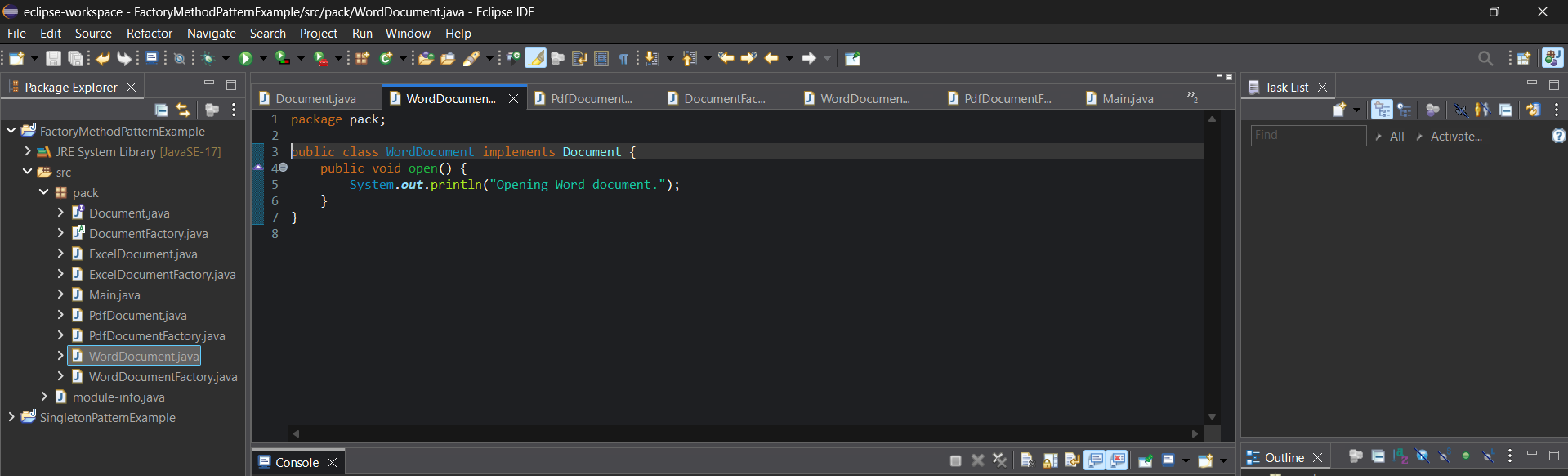
**ExcelDocument.java**

****

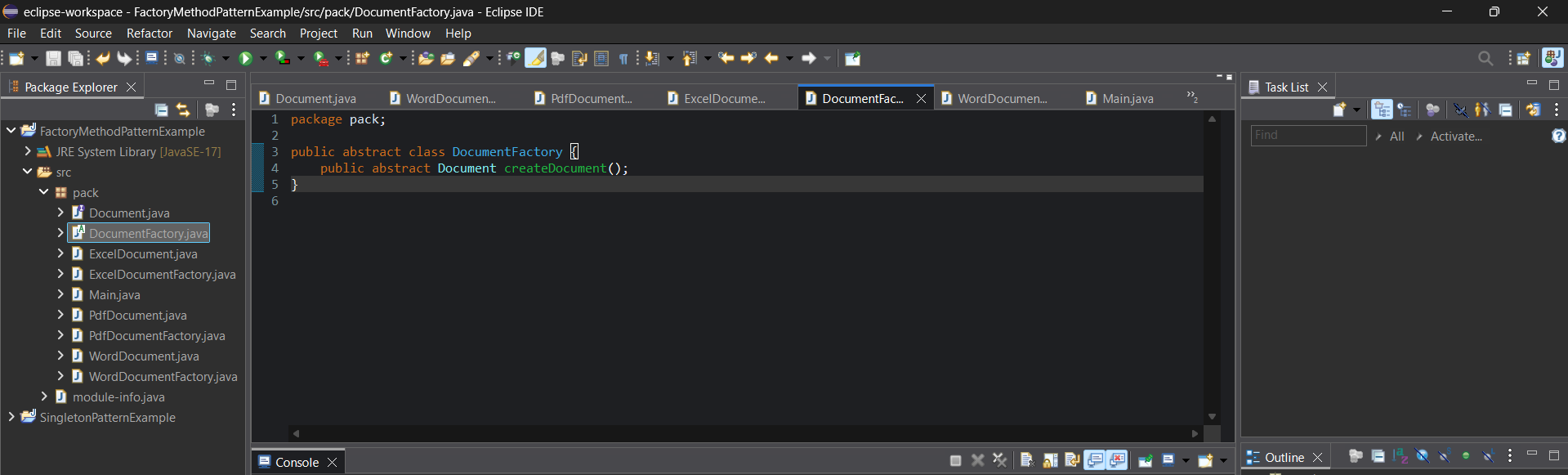
**PdfDocument.java**

****

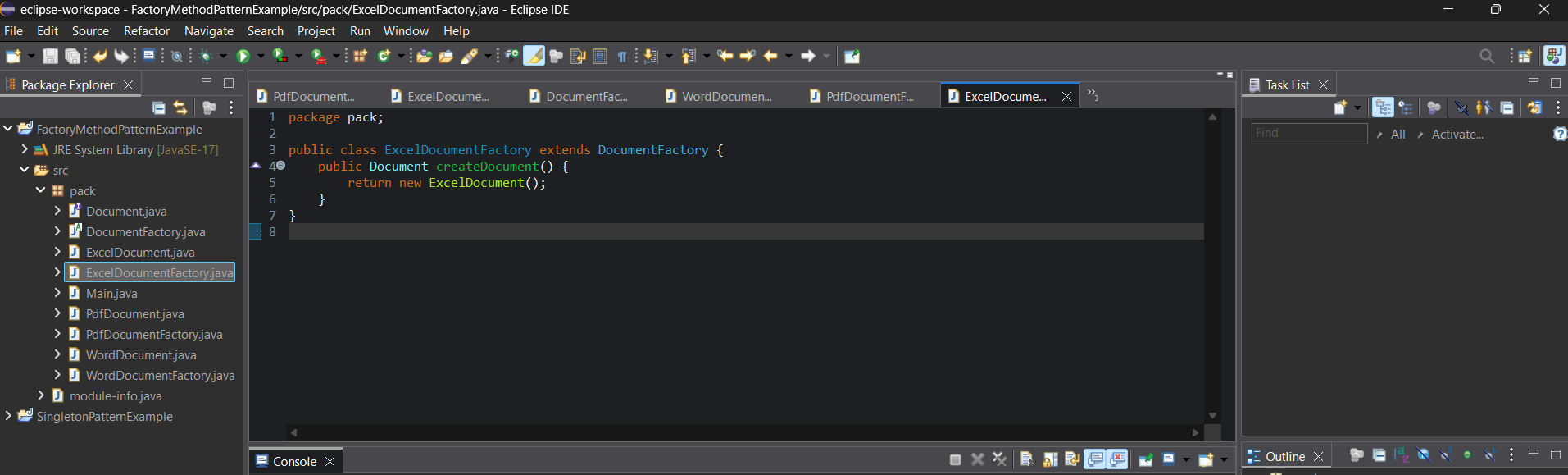
**WordDocument.java**

****

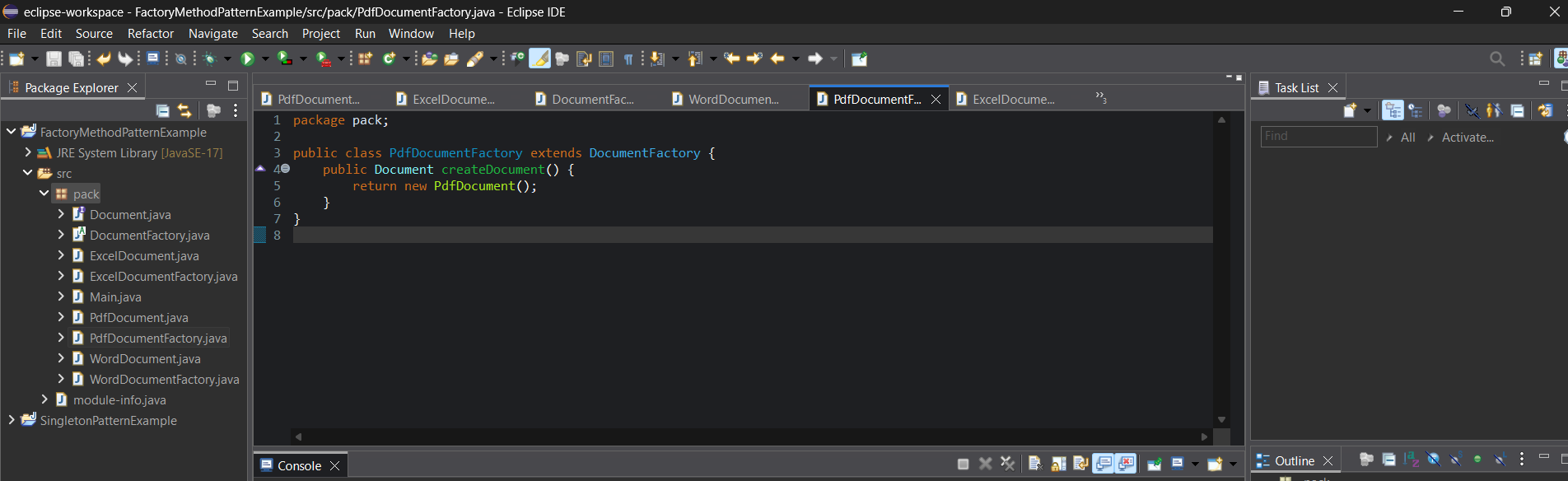
**DocumentFactory.java**

****

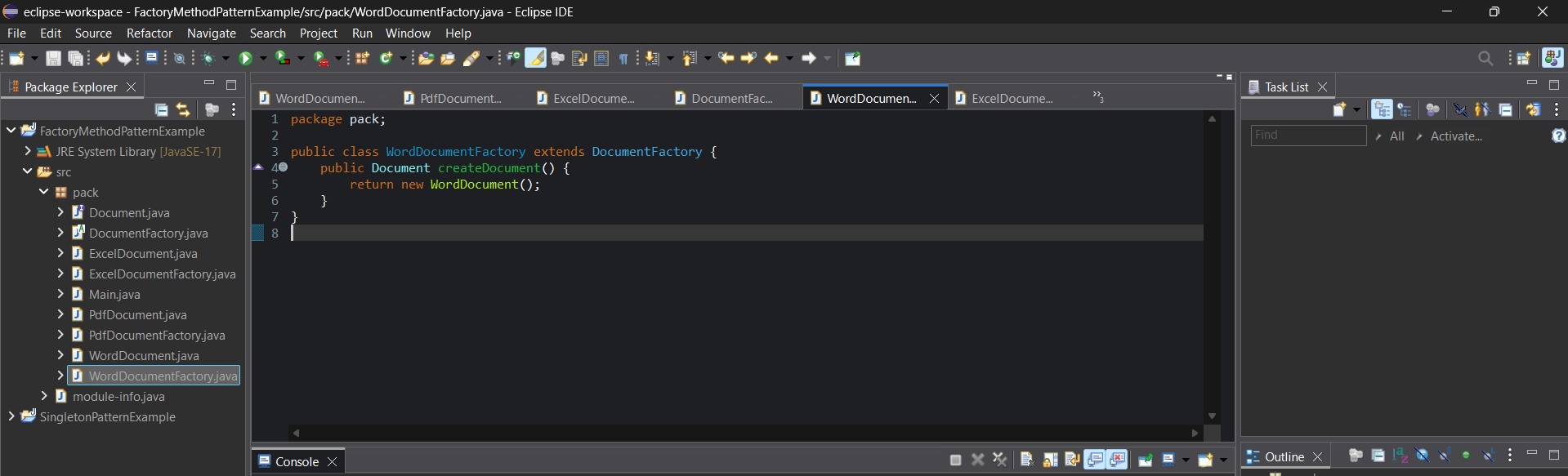
**ExcelDocumentFactory.java**

****

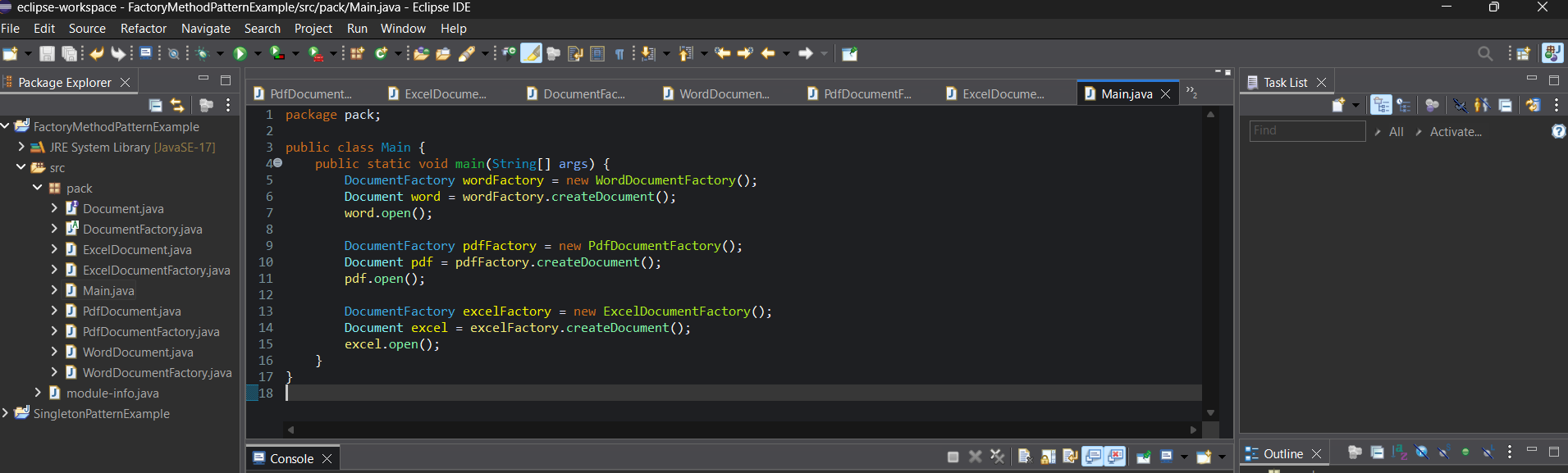
**PdfDocumentFactory.java**

****

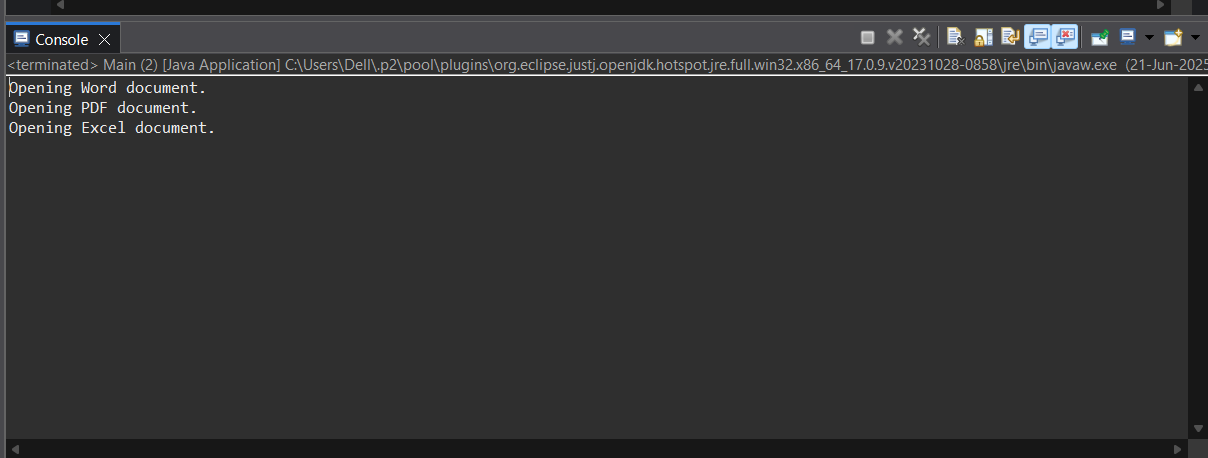
**WordDocumentFactory.java**

****

**Main.java**

****

**OUTPUT:**

****