**Exercise 1: Implementing the Singleton Pattern**

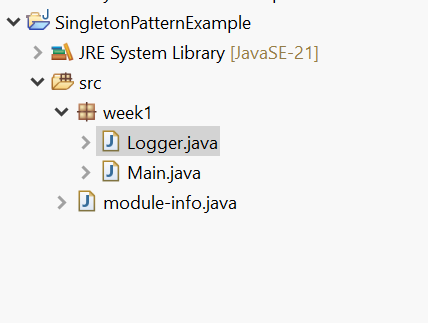
**Scenario:**

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:**

1. **Create a New Java Project:**
   * Create a new Java project named SingletonPatternExample.
2. **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.
3. **Implement the Singleton Pattern:**
   * Write code to ensure that the Logger class follows the Singleton design pattern.
4. **Test the Singleton Implementation:**
   * Create a test class to verify that only one instance of Logger is created and used across the application.

**Project Structure:**



**Code:**

**Logger.java:**

package week1;

public class Logger {

private static Logger *curlog* = new Logger();//log object

private Logger() {

System.*out*.println("Logger Class Called");

}

static Logger getInstance() {

return *curlog*;

}

//logging method

public void log(String msg) {

System.*out*.println(msg);

}

}

**Main.java:**

package week1;

public class Main {

public static void main(String[] args) {

Logger log1 = Logger.*getInstance*();

log1.log("log1");

Logger log2 = Logger.*getInstance*();

log2.log("log2");

if(log1==log2) {

System.*out*.println("log1 and log2 Objects are Same");

}else {

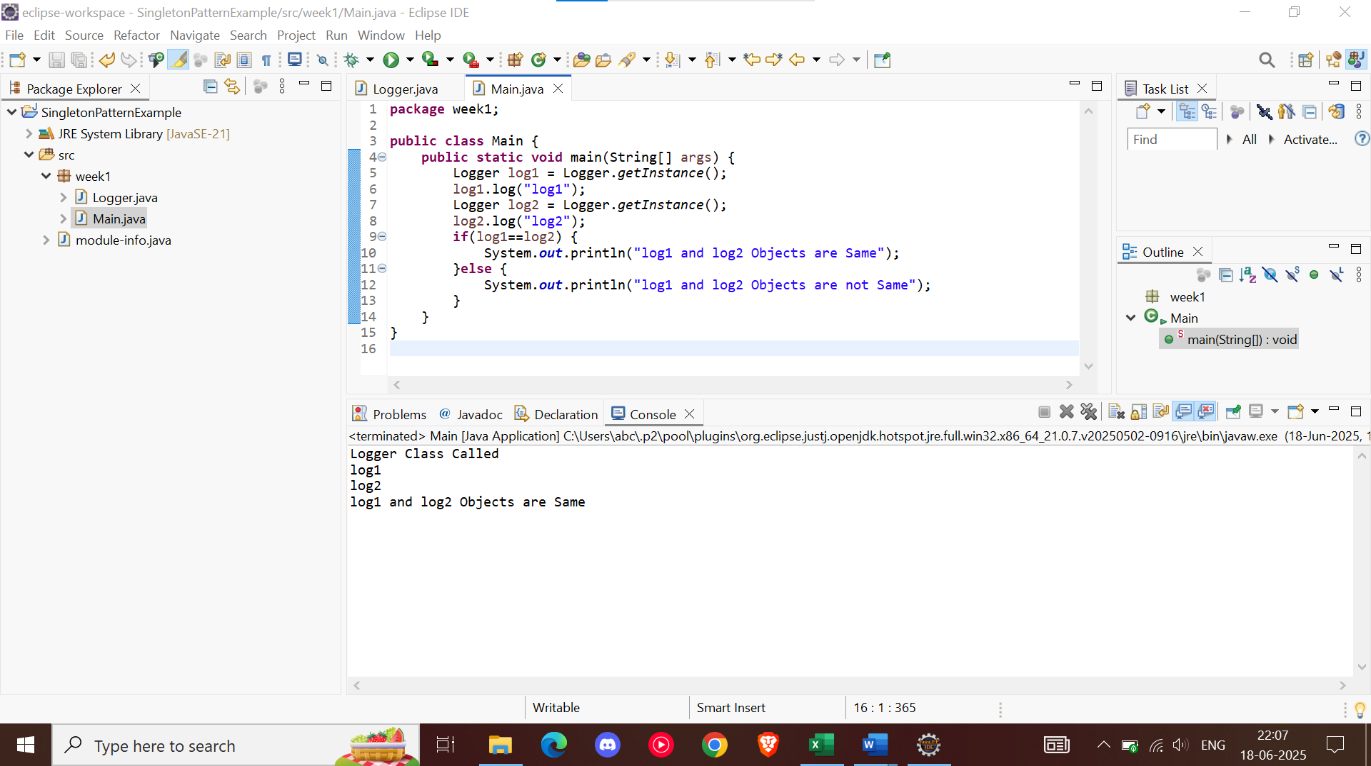
System.*out*.println("log1 and log2 Objects are not Same");

}

}

}

**Output:-**



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

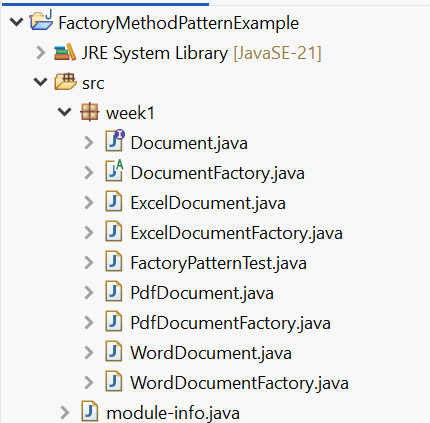
**Scenario:**

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.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **FactoryMethodPatternExample**.
2. **Define Document Classes:**
   * Create interfaces or abstract classes for different document types such as **WordDocument**, **PdfDocument**, and **ExcelDocument**.
3. **Create Concrete Document Classes:**
   * Implement concrete classes for each document type that implements or extends the above interfaces or abstract classes.
4. **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.
5. **Test the Factory Method Implementation:**
   * Create a test class to demonstrate the creation of different document types using the factory method.

**Project Structure:**

****

**Code:**

**Document.java:**

package week1;

public interface Document{

void open();

}

**DocumentFactory.java:**

package week1;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

**ExcelDocument.java:**

package week1;

public class ExcelDocument implements Document {

public void open() {

System.***out***.println("Excel Document Opened");

}

}

**ExcelDocumentFactory.java:**

package week1;

public class ExcelDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

**PdfDocument.java:**

package week1;

public class PdfDocument implements Document {

public void open() {

System.***out***.println("Pdf Document Opened");

}

}

**PdfDocumentFactory.java:**

package week1;

public class PdfDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

**WordDocument.java:**

package week1;

public class WordDocument implements Document{

public void open() {

System.***out***.println("Word Document Opened");

}

}

**WordDocumentFactory.java:**

package week1;

public class WordDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

**FactoryPatternTest.java:**

package week1;

public class FactoryPatternTest {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordDocumentFactory();

Document wordDoc = wordFactory.createDocument();

wordDoc.open();

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

DocumentFactory excelFactory = new ExcelDocumentFactory();

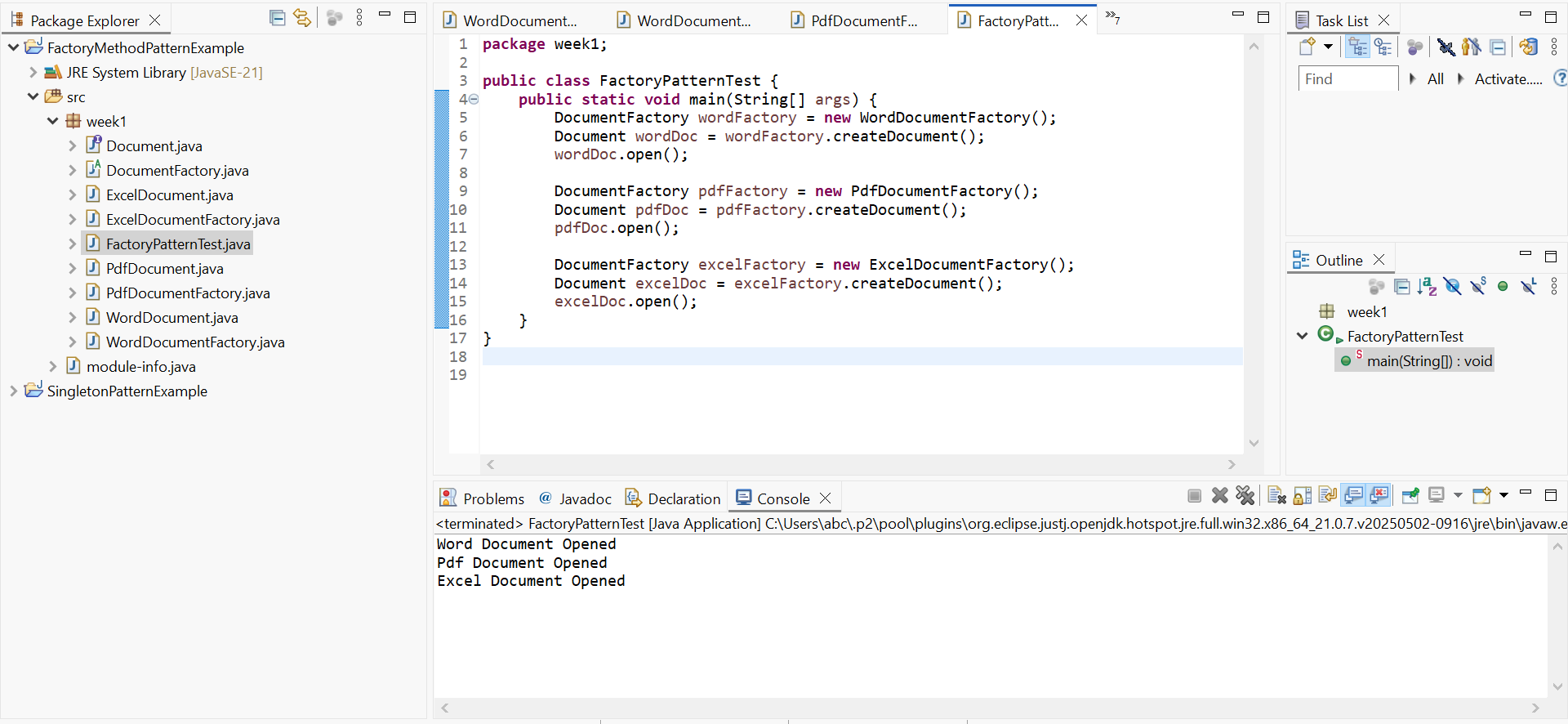
Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

**Output:**

****