**Exercise 1: Implementing the Singleton Pattern**

**CODE :**

**Logger.java**

package singleton;

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);

    }

}

**LoggerTest.java**

package singleton;

public class LoggerTest {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        Logger logger2 = Logger.getInstance();

        logger1.log("First log message");

        logger2.log("Second log message");

        if (logger1 == logger2) {

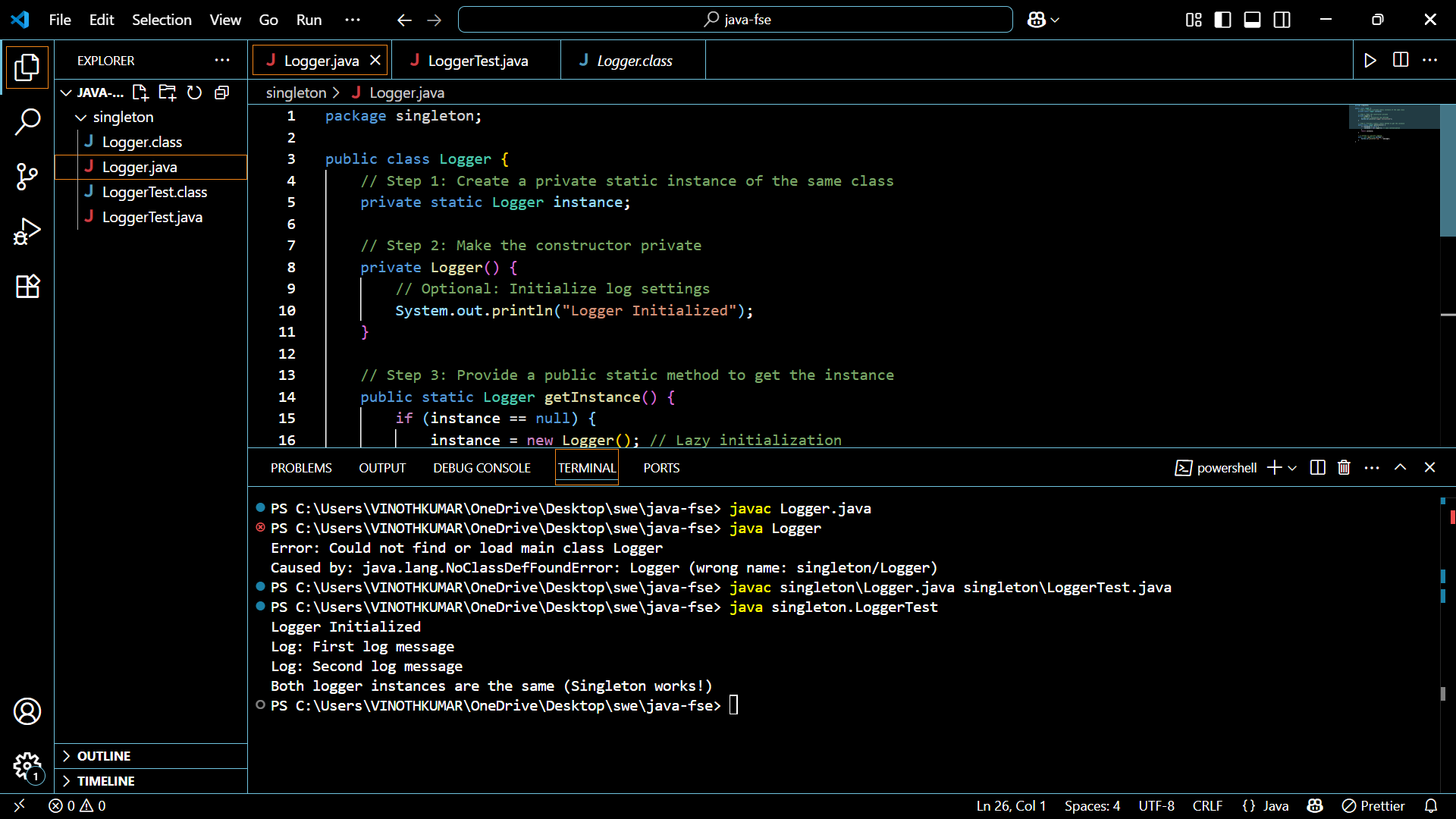
            System.out.println("Both logger instances are the same (Singleton works!)");

        } else {

            System.out.println("Different instances found (Singleton failed)");

        }    } }

**OUTPUT :**

****

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

**CODE :**

**Document.java**

package documents;

public interface Document {

    void open();

}

**ExcelDocument.java**

package documents;

public class ExcelDocument implements Document {

    public void open() {

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

    } }

**PdfDocument.java**

package documents;

public class PdfDocument implements Document {

    public void open() {

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

    }

}

**WordDocument.java**

package documents;

public class WordDocument implements Document {

    public void open() {

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

    }

}

**DocumentFactory.java**

package factories;

import documents.Document;

public abstract class DocumentFactory {

    public abstract Document createDocument();

}

**ExcelDocumentFactory.java**

package factories;

import documents.Document;

import documents.ExcelDocument;

public class ExcelDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new ExcelDocument();

    } }

**PdfDocumentFactory.java**

package factories;

import documents.Document;

import documents.PdfDocument;

public class PdfDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new PdfDocument();

    }

}

**WordDocumentFactory.java**

package factories;

import documents.Document;

import documents.WordDocument;

public class WordDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new WordDocument();

    }

}

**TestFactoryMethod.java**

import factories.\*;

import documents.Document;

public class TestFactoryMethod {

    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();

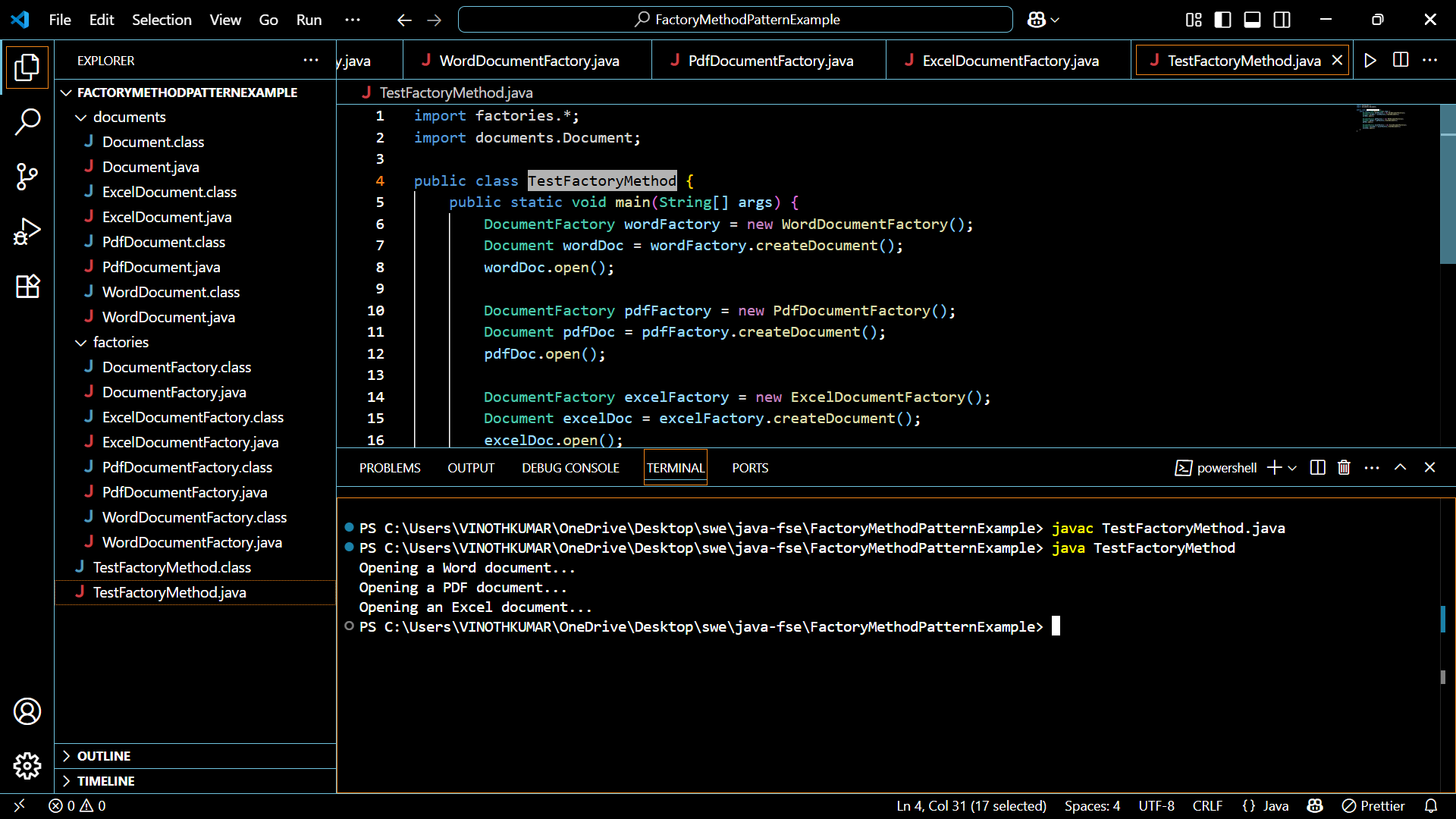
        Document excelDoc = excelFactory.createDocument();

        excelDoc.open();

    }

}

**OUTPUT :**

****