**WEEK-1\_** **Design Patterns and Principles\_HandsOn**

**Exercise 1: Implementing the Singleton Pattern**

**Logger.java**

package singletonexample;

public class Logger {

private static Logger instance;

private Logger() {

System.out.println("Logger instance created.");

}

public static Logger getInstance() {

if (instance == null) {

instance = new Logger();

}

return instance;

}

public void log(String message) {

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

}

}

**LogTester.java**

package singletonexample;

public class LogTester {

public static void main(String[] args) {

Logger logger1 = Logger.*getInstance*();

logger1.log("First message");

Logger logger2 = Logger.*getInstance*();

logger2.log("Second message");

if (logger1 == logger2) {

System.*out*.println("Both logger instances are the same (Singleton verified).");

} else {

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

}

}

}

**OUTPUT**

**A screenshot of a computer program

AI-generated content may be incorrect.**

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

**DocumentFactory.java**

package factorymethodexample;

public abstract class DocumentFactory {

public abstract Object createDocument();

}

**Interfaces:**

package factorymethodexample;

public interface ExcelDocument {

void openExcel();

}

package factorymethodexample;

public interface WordDocument {

void openWord();

}

package factorymethodexample;

public interface PdfDocument {

void openPdf();

}

**Concrete Factories**

package factorymethodexample;

public class WordDocumentFactory extends DocumentFactory {

public WordDocument createDocument() {

return new WordDocumentImpl();

}

}

package factorymethodexample;

public class PdfDocumentFactory extends DocumentFactory {

public PdfDocument createDocument() {

return new PdfDocumentImpl();

}

}

package factorymethodexample;

public class WordDocumentFactory extends DocumentFactory {

public WordDocument createDocument() {

return new WordDocumentImpl();

}

}

**Implementations**

package factorymethodexample;

public class WordDocumentImpl implements WordDocument {

public void openWord() {

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

}

}

package factorymethodexample;

public class PdfDocumentImpl implements PdfDocument {

public void openPdf() {

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

}

}

package factorymethodexample;

public class ExcelDocumentImpl implements ExcelDocument {

public void openExcel() {

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

}

}

**TestFactoryMethod:**

package factorymethodexample;

public class TestFactoryMethod {

public static void main(String[] args) {

WordDocumentFactory wordFactory = new WordDocumentFactory();

WordDocument word = wordFactory.createDocument();

word.openWord();

PdfDocumentFactory pdfFactory = new PdfDocumentFactory();

PdfDocument pdf = pdfFactory.createDocument();

pdf.openPdf();

ExcelDocumentFactory excelFactory = new ExcelDocumentFactory();

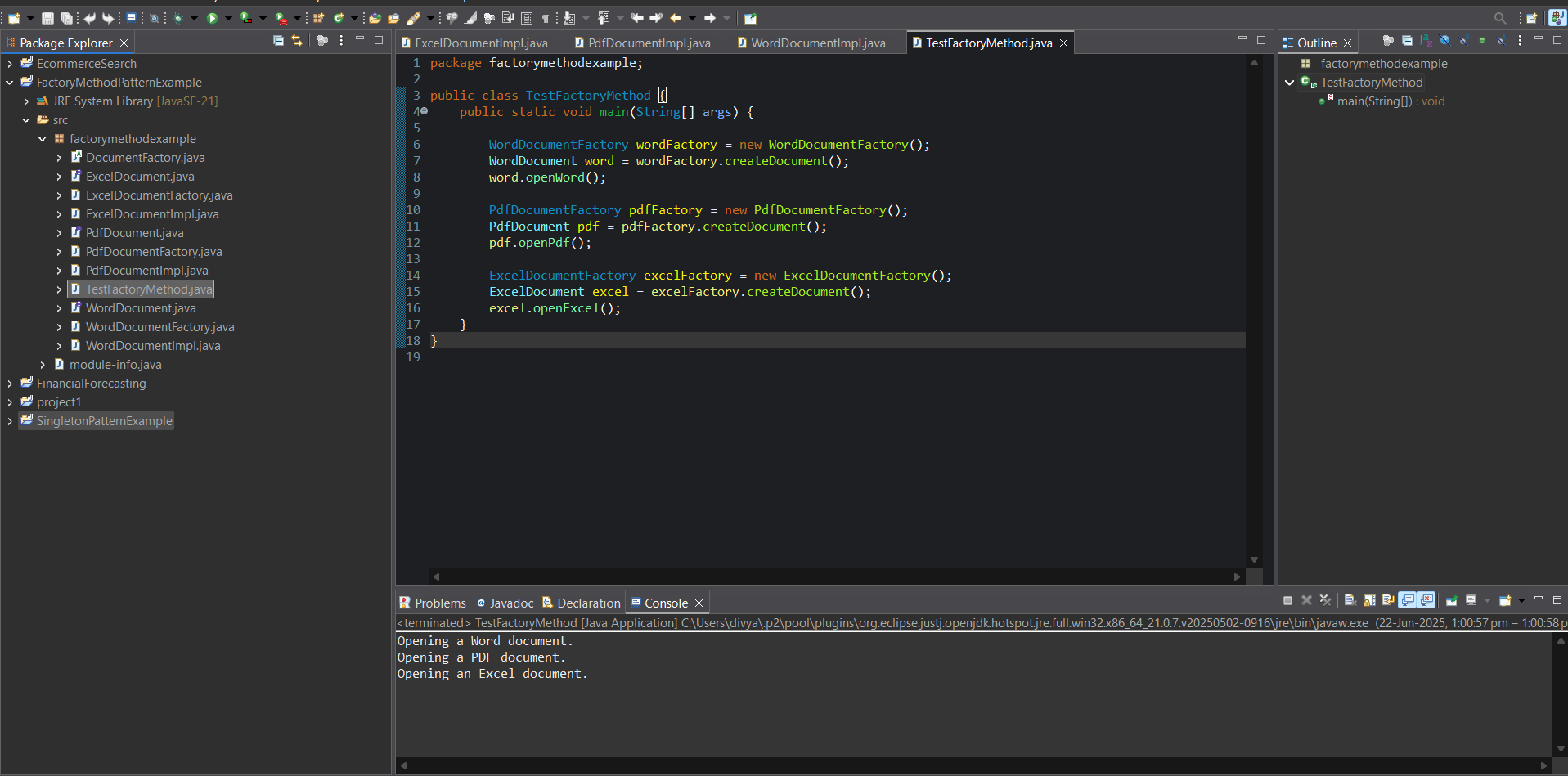
ExcelDocument excel = excelFactory.createDocument();

excel.openExcel();

}

}

**OUTPUT:**

****