**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.*

// SingletonPatternExample.java

public class SingletonPatternExample {

// Singleton Logger Class

static 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 Method to Test Singleton

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

logger1.log("First log message");

Logger logger2 = Logger.getInstance();

logger2.log("Second log message");

if (logger1 == logger2) {

System.out.println("Both loggers are the same instance.");

} else {

System.out.println("Different instances (Error!).");

}

}

}

A screenshot of a computer program

AI-generated content may be incorrect.

**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.*

*// FactoryMethodPatternExample.java*

*public class FactoryMethodPatternExample {*

*// Document Interface*

*interface Document {*

*void open();*

*}*

*// Concrete Document Classes*

*static class WordDocument implements Document {*

*public void open() {*

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

*}*

*}*

*static class PdfDocument implements Document {*

*public void open() {*

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

*}*

*}*

*static class ExcelDocument implements Document {*

*public void open() {*

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

*}*

*}*

*// Abstract Factory*

*static abstract class DocumentFactory {*

*public abstract Document createDocument();*

*}*

*// Concrete Factories*

*static class WordDocumentFactory extends DocumentFactory {*

*public Document createDocument() {*

*return new WordDocument();*

*}*

*}*

*static class PdfDocumentFactory extends DocumentFactory {*

*public Document createDocument() {*

*return new PdfDocument();*

*}*

*}*

*static class ExcelDocumentFactory extends DocumentFactory {*

*public Document createDocument() {*

*return new ExcelDocument();*

*}*

*}*

*// Main Method to Test Factory Pattern*

*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();*

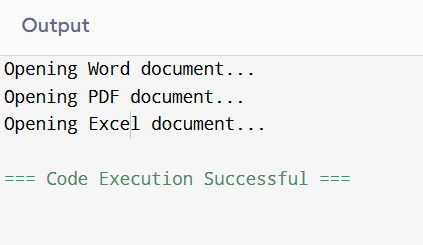
*DocumentFactory excelFactory = new ExcelDocumentFactory();*

*Document excel = excelFactory.createDocument();*

*excel.open();*

*}*

*}*

**