**Exercise 1: Implementing the Singleton Pattern**

LoggerTest.java

package ImplementingtheSingletonPattern;

public class LoggerTest {

public static void main(String[] args) {

// Attempt to get two instances of Logger

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

// Test to check if both references point to the same object

if (logger1 == logger2) {

System.***out***.println("Both logger1 and logger2 are the same instance.");

} else {

System.***out***.println("Different instances exist!");

}

// Using the Logger

logger1.log("First message");

logger2.log("Second message");

}

}

**Logger.java**

package ImplementingtheSingletonPattern;

public class Logger {

// Private static instance of the same class

private static Logger *instance*;

// Private constructor to prevent instantiation

private Logger() {

System.***out***.println("Logger Initialized.");

}

// Public method to provide access to the instance

public static Logger getInstance() {

if (*instance* == null) {

*instance* = new Logger();

}

return *instance*;

}

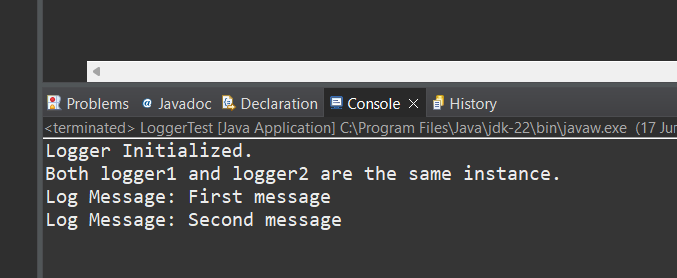
// Sample method to simulate logging

public void log(String message) {

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

}

}



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

**Document.java**

package FactoryMethodPattern;

public interface Document { void open();

}

**DocumentFactory.java**

package FactoryMethodPattern;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

**ExcelDocument.java**

package FactoryMethodPattern;

public class ExcelDocument implements Document{

public void open() {

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

}

}

**ExcelDocumentFactory.java**

package FactoryMethodPattern;

public class ExcelDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

**PdfDocument.java**

package FactoryMethodPattern;

public class PdfDocument implements Document {

public void open() {

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

}

}

**PdfDocumentFactory.java**

package FactoryMethodPattern;

public class PdfDocumentFactory extends DocumentFactory {

public Document createDocument() {

return (Document) new PdfDocument();

}

}

**WordDocument.java**

package FactoryMethodPattern;

public class WordDocument implements Document {

public void open() {

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

}

}

**WordDocumentFactory.java**

package FactoryMethodPattern;

public class WordDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

**FactoryMethodTest.java**

package FactoryMethodPattern;

public class FactoryMethodTest {

public static void main(String[] args) {

// Create Word Document using Factory

DocumentFactory wordFactory = new WordDocumentFactory();

Document wordDoc = wordFactory.createDocument();

wordDoc.open();

// Create PDF Document using Factory

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

// Create Excel Document using Factory

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

