**Exercise 1: Implementing the Singleton Pattern**

**Logger.java:**

package SingletonPatternExample;

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);

}

}

**Main.java:**

package SingletonPatternExample;

public class Main {

public static void main(String[] args) {

Logger logger1 = Logger.*getInstance*();

logger1.log("This is the first log message.");

Logger logger2 = Logger.*getInstance*();

logger2.log("This is the second log message.");

if (logger1 == logger2) {

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

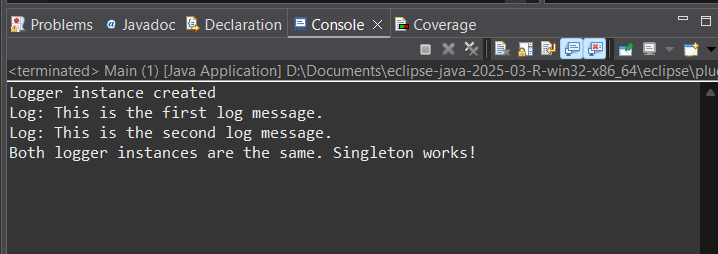
} else {

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

}

}

}

**OUTPUT:** 

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

**Document.java:**

package FactoryMethodPatternExample;

public interface Document {

void open();

}

**DocumentFactory.java:**

package FactoryMethodPatternExample;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

**ExcelDocument.java:**

package FactoryMethodPatternExample;

public class ExcelDocument implements Document {

@Override

public void open() {

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

}

}

**ExcelDocumentFactory.java:**

package FactoryMethodPatternExample;

public class ExcelDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new ExcelDocument();

}

}

**PdfDocument.java:**

package FactoryMethodPatternExample;

public class PdfDocument implements Document {

@Override

public void open() {

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

}

}

**PdfDocumentFactory.java:**

package FactoryMethodPatternExample;

public class PdfDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new PdfDocument();

}

}

**WordDocument.java:**

package FactoryMethodPatternExample;

public class PdfDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new PdfDocument();

}

}

**WordDocumentFactory.java:**

package FactoryMethodPatternExample;

public class WordDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new WordDocument();

}

}

**Main.java:**

package FactoryMethodPatternExample;

public class Main {

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();

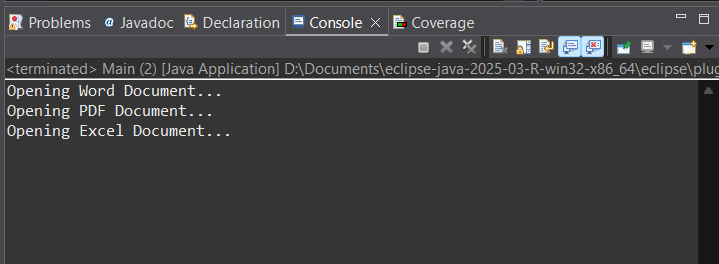
Document excel = excelFactory.createDocument();

excel.open();

}

}

**Output:**

****