**Week-1 : Design Patterns and Principles**

**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.

**Steps:**

**1. Create a New Java Project:**

* Create a new Java project named SingletonPatternExample.

**2. Define a Singleton Class:**

* Create a class named Logger that has a private static instance of itself.
* Ensure the constructor of Logger is private.
* Provide a public static method to get the instance of the Logger class.

**3. Implement the Singleton Pattern:**

* Write code to ensure that the Logger class follows the Singleton design pattern.

**4. Test the Singleton Implementation:**

* Create a test class to verify that only one instance of Logger is created and used across the application.

**Code Implementation:**

***Logger.java***

package com.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);

}

}

***LoggerTest.java***

package com.singletonPatternExample;

public class LoggerTest {

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

Logger logger2 = Logger.getInstance();

logger1.log("First message");

logger2.log("Second message");

if (logger1 == logger2) {

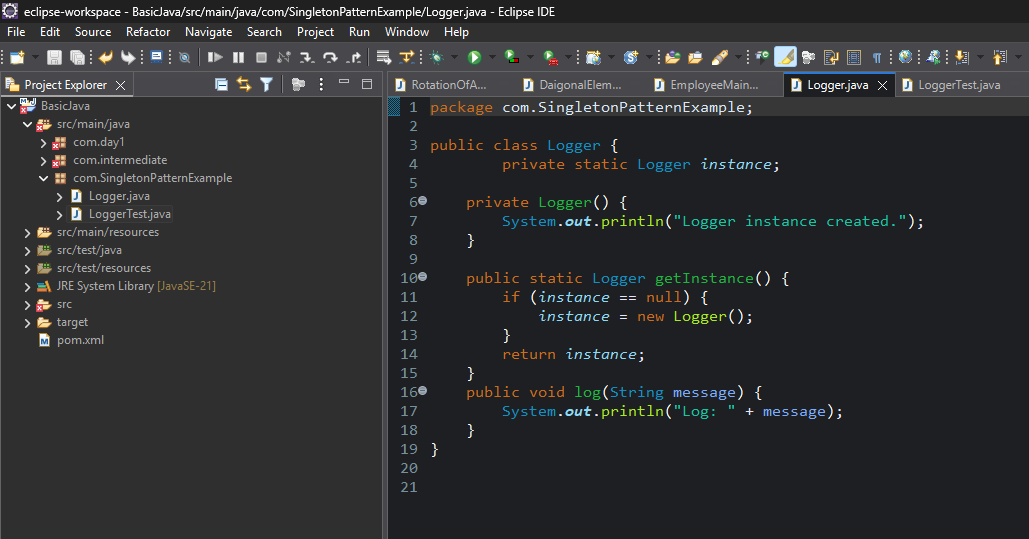
System.out.println("Both logger1 and logger2 are the same instance.");

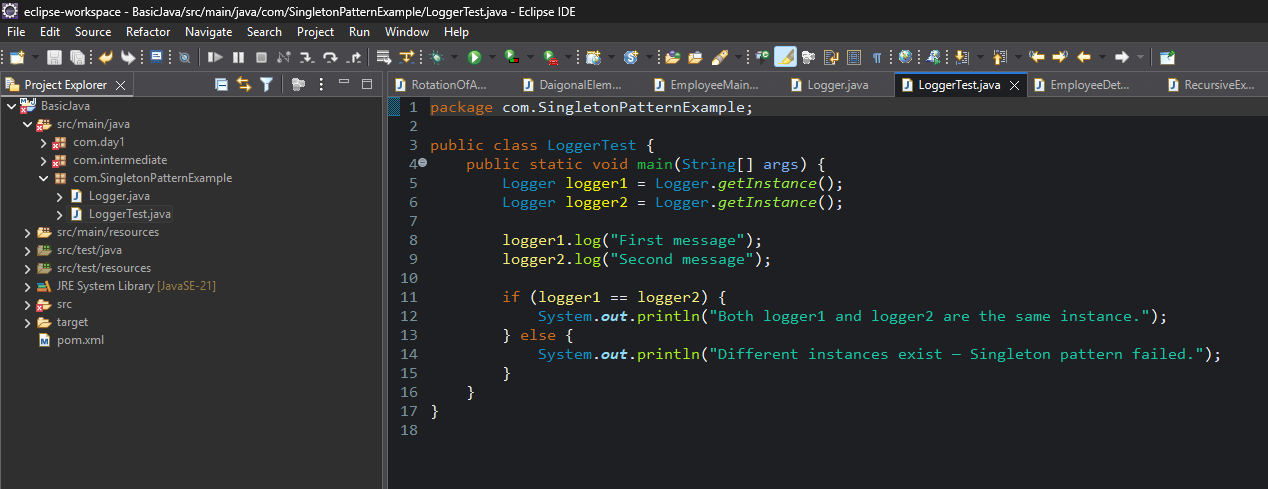
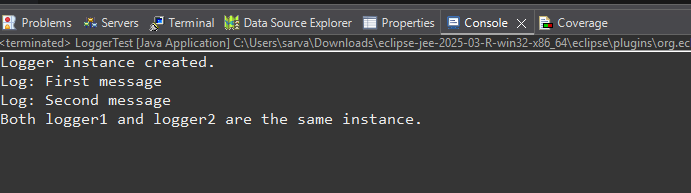
} else {

System.out.println("Different instances exist — Singleton pattern failed.");

}

}

}

**Output :**

**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.

Steps:

**1. Create a New Java Project:**

* Create a new Java project named FactoryMethodPatternExample.

**2. Define Document Classes:**

* Create interfaces or abstract classes for different document types such as WordDocument, PdfDocument, and ExcelDocument.

**3. Create Concrete Document Classes:**

* Implement concrete classes for each document type that implements or extends the above interfaces or abstract classes.

4**. Implement the Factory Method:**

* Create an abstract class DocumentFactory with a method createDocument().
* Create concrete factory classes for each document type that extends DocumentFactory and implements the createDocument() method.

**5. Test the Factory Method Implementation:**

* Create a test class to demonstrate the creation of different document types using the factory method.

1. ***Document.java (Interface)***

package com.factorymethod;

public interface Document {

void open();

}

1. ***WordDocument.java***

package com.factorymethod;

public class WordDocument implements Document {

public void open() {

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

}

}

1. ***PdfDocument.java***

package com.factorymethod;

public class PdfDocument implements Document {

public void open() {

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

}

}

1. ***ExcelDocument.java***

package com.factorymethod;

public class ExcelDocument implements Document {

public void open() {

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

}

}

1. ***DocumentFactory.java (Abstract Factory)***

package com.factorymethod;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

1. ***WordDocumentFactory.java***

package com.factorymethod;

public class WordDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

1. ***PdfDocumentFactory.java***

package com.factorymethod;

public class PdfDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

1. ***ExcelDocumentFactory.java***

package com.factorymethod;

public class ExcelDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

1. ***DocumentTest.java (Test Class)***

package com.factorymethod;

public class DocumentTest {

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();

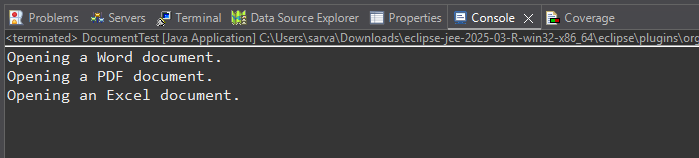
Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

**Output :**

****