**01: Implementing the Singleton Pattern**

* **DEFINING A SINGLETON FILE**
  + **Logger Class**

package week\_1\_Design\_Pattern\_Principles\_Singleton\_Pattern;

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);

}

}

}

* + **CLASS FOR TESTING**

package week\_1\_Design\_Pattern\_Principles\_Singleton\_Pattern;

public class LoggerTest {

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 verified).");

} else {

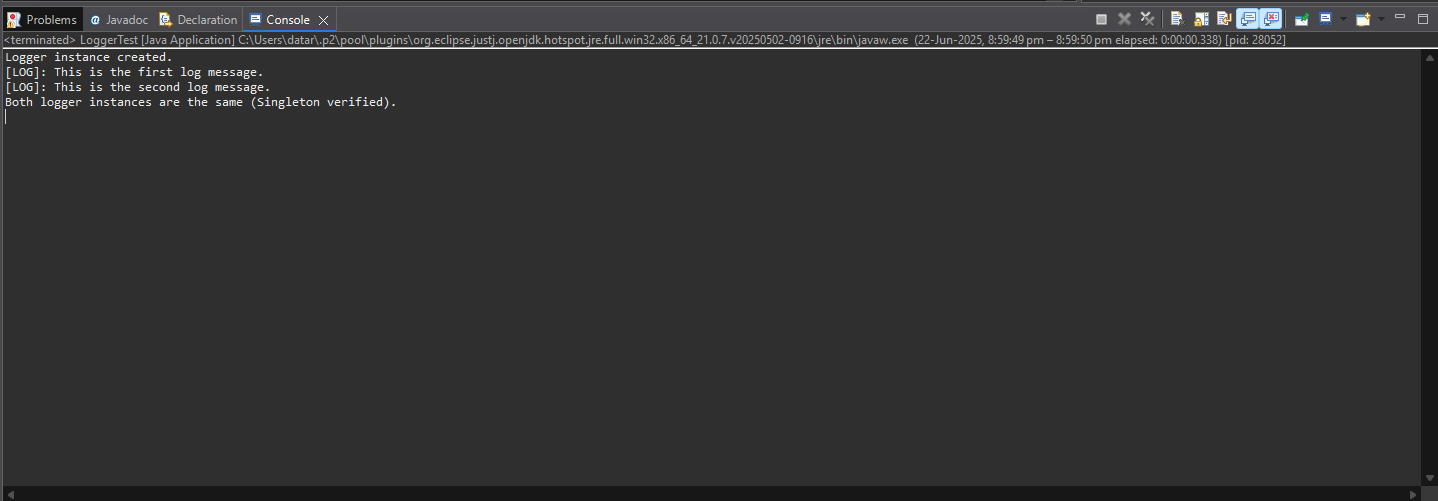
System.***out***.println("Different logger instances exist (Singleton violated).");

}

}

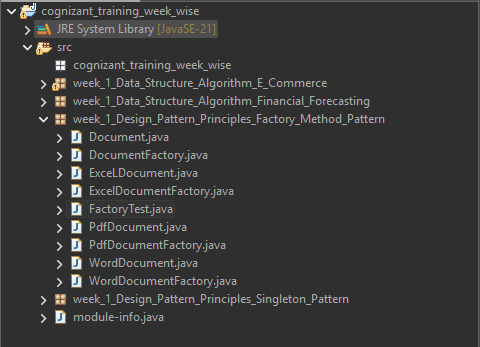
}

**Output:**



**02 : Implementing the Factory Method Pattern**

* **PROJECT STRUCTURE**



* **DOCUMENT CLASS**

public interface Document {

void open();

}

* **CONCRETE DOCUMENT CLASS**

// WordDocument.java

public class WordDocument implements Document {

@Override

public void open() {

System.out.println("Wait…..Opening Word Document...");

}

}

// PdfDocument.java

public class PdfDocument implements Document {

@Override

public void open() {

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

}

}

// ExcelDocument.java

public class ExcelDocument implements Document {

@Override

public void open() {

System.out.println("Wait…..Opening Excel document...");

}

}

* **DOCUMENT FACTORY CLASS**

// DocumentFactory.java

public abstract class DocumentFactory {

public abstract Document createDocument();

}

* **CONCRETE FACTORIES**

// WordDocumentFactory.java

public class WordDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new WordDocument();

}

}

// PdfDocumentFactory.java

public class PdfDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new PdfDocument();

}

}

// ExcelDocumentFactory.java

public class ExcelDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new ExcelDocument();

}

}

* **FACTORY TEST (TESTING MAIN) METHOD**

// FactoryTest.java

public class FactoryTest {

public static void main(String[] args) {

// Create and open a Word document

DocumentFactory wordFactory = new WordDocumentFactory();

Document word = wordFactory.createDocument();

word.open();

// Create and open a PDF document

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdf = pdfFactory.createDocument();

pdf.open();

// Create and open an Excel document

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excel = excelFactory.createDocument();

excel.open();

}

}

* **OUTPUT**

