**Exercise 1: Implementing the Singleton Pattern**

**Logger.java**

package SingletonPatternExample;

public 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.java**

**package** SingletonPatternExample;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

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

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

**if** (logger1 == logger2) {

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

} **else** {

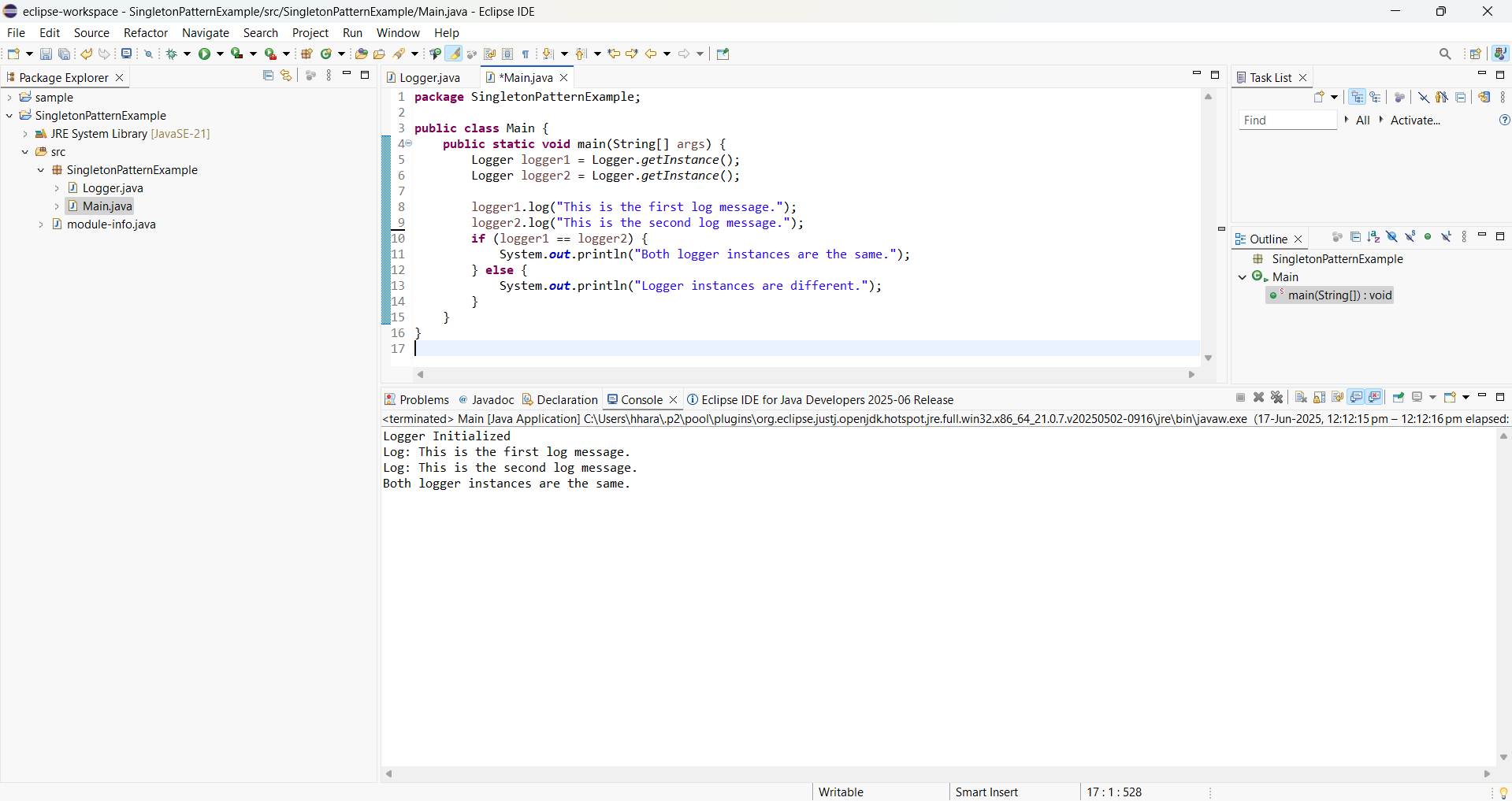
System.***out***.println("Logger instances are different.");

}

}

}

**OUTPUT:**



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

**Document.java**

package FactoryMethodPatternExample;

public interface Document {

void open();

}

**WordDocument.java**

package FactoryMethodPatternExample;

public class WordDocument implements Document {

@Override

public void open() {

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

}

}

**PdfDocument.java**

package FactoryMethodPatternExample;

public class PdfDocument implements Document {

@Override

public void open() {

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

}

}

**ExcelDocument.java**

package FactoryMethodPatternExample;

public class ExcelDocument implements Document {

@Override

public void open() {

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

}

}

**DocumentFactory.java**

package FactoryMethodPatternExample;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

**WordDocumentFactory.java**

package FactoryMethodPatternExample;

public class WordDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new WordDocument();

}

}

**PdfDocumentFactory.java**

package FactoryMethodPatternExample;

public class PdfDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new PdfDocument();

}

}

**ExcelDocumentFactory.java**

package FactoryMethodPatternExample;

public class ExcelDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new ExcelDocument();

}

}

**Main.java**

package FactoryMethodPatternExample;

public class Main {

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:**