“DESIGN PATTERNS AND PRINCIPLES”

**Exercise 1: Implementing the Singleton Pattern**

**1.Logger class**

import java.util.\*;

public class Logger {

private static Logger *instance*; //static class

public Logger() {

System.*out*.println("singleton pattern Logger instance created.");

}

public static Logger getInstance() {

if (*instance* == null) {

*instance* = new Logger(); // one instance is created

}

return *instance*;

} // A sample logging method

public void log(String message) {

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

} }

**2.Logger Test Class**

public class LoggerTest {

public static void main(String[] args) {

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

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

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

if (logger1 == logger2) {

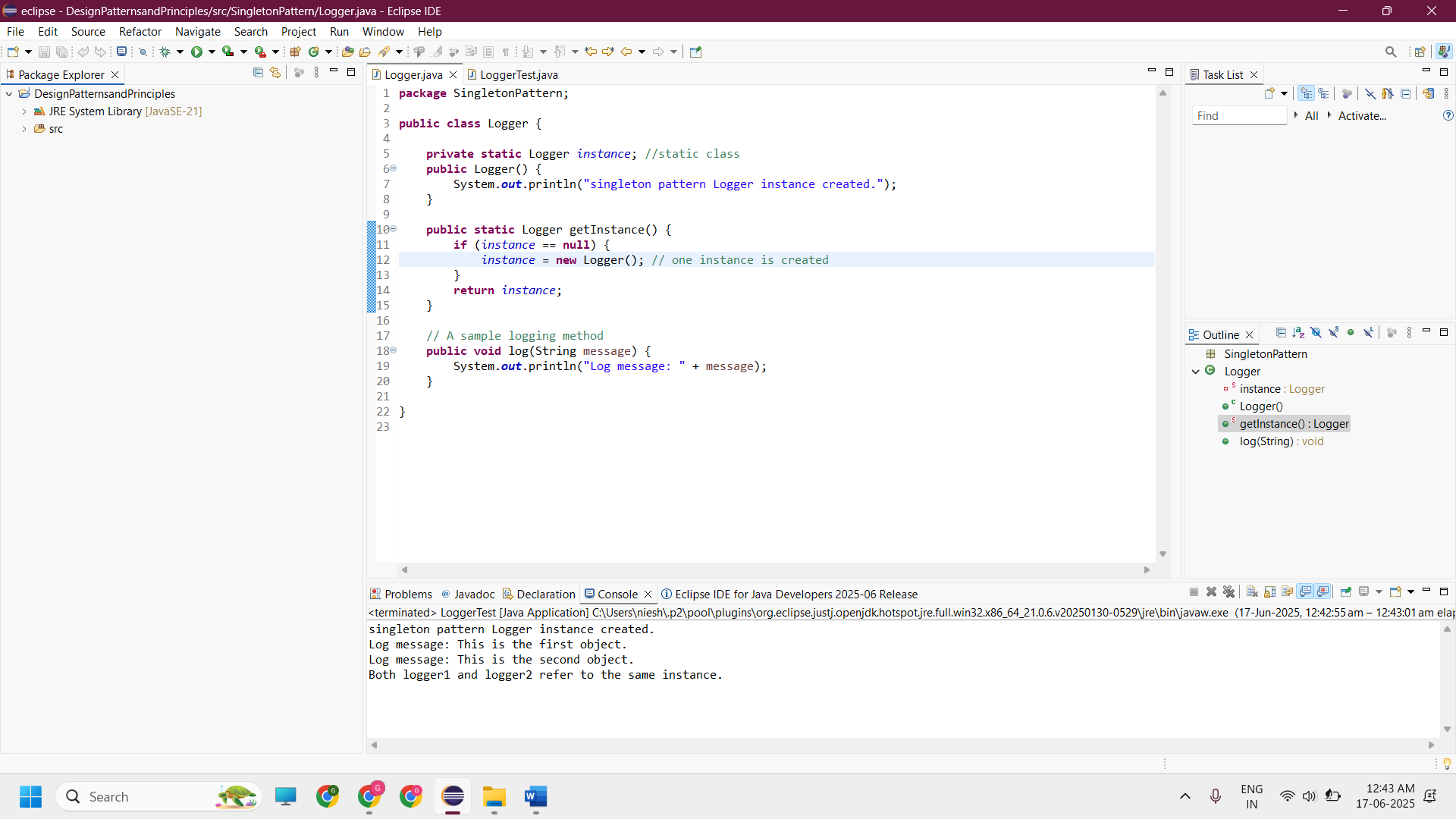
System.*out*.println("Both logger1 and logger2 refer to the same instance.");

} else {

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

} }

}

**---------------------------------------------------------------------------------------------------------------------------**

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

**1.Document.java**

package FactoryMethodCode;

public interface Document {

void access();

}

**2.WordDocument.java**

public class WordDocument implements Document {

public void access() {

System.out.println("Word Document is accessed");

}

}

**3.PdfDocument.java**

public class PdfDocument implements Document {

public void access() {

System.out.println("PDF Document is accessed");

}

}

**4.ExcelDocument.java**

public class ExcelDocument implements Document {

public void access() {

System.out.println("Excel Document is accessed");

}

}

**5.DocumentFactory.java**

public abstract class DocumentFactory {

public abstract Document createDocument();

}

**6.WordFactory.java**

public class WordFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new WordDocument();

}

}

**7.PdfFactory.java**

public class PdfFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new PdfDocument();

}

}

**8.ExcelFactory.java**

public class ExcelFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new ExcelDocument();

}

}

**9.FactoryPatternTest.java**

public class FactoryPatternTest {

public static void main(String[] args) {

DocumentFactory wordfactory = new WordFactory();

Document word = wordfactory.createDocument();

word.access();

DocumentFactory pdffactory = new PdfFactory();

Document pdf = pdffactory.createDocument();

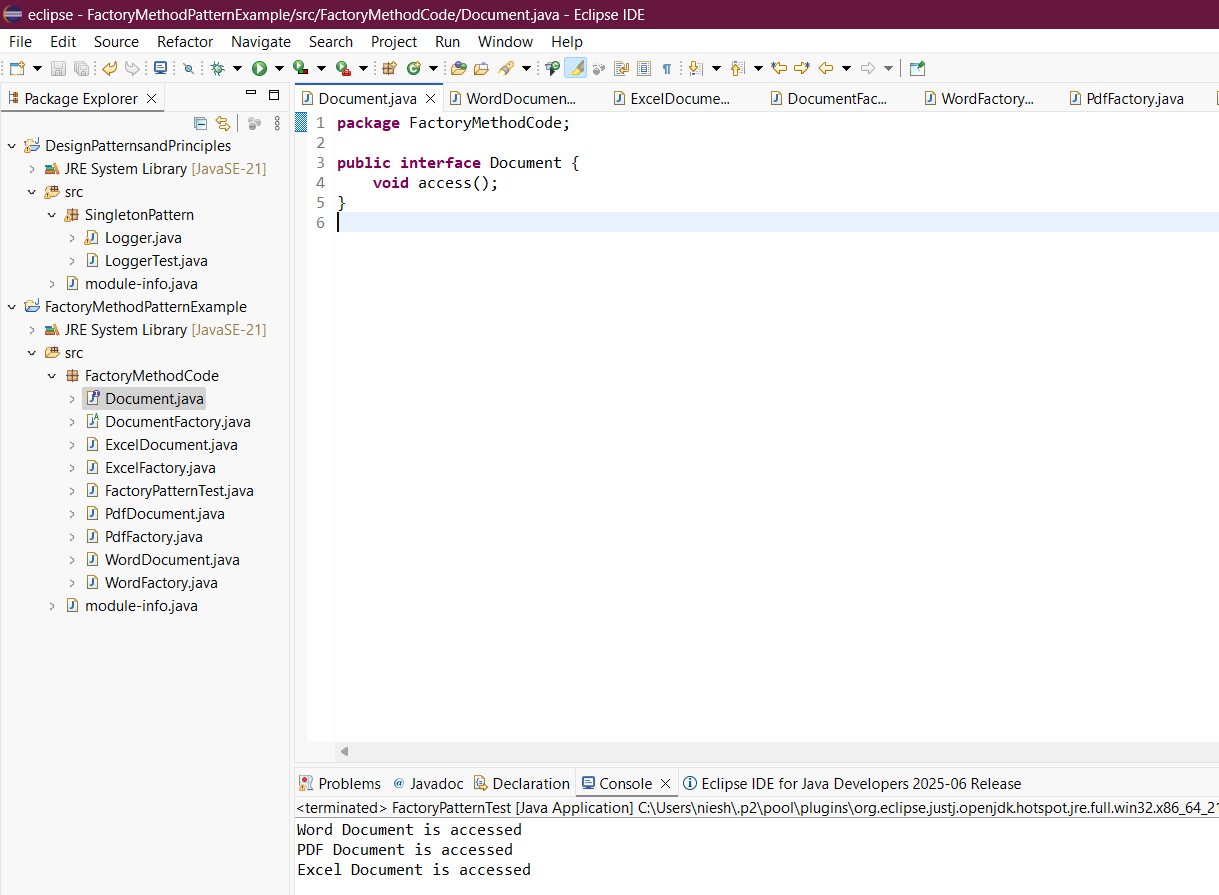
pdf.access();

DocumentFactory excelfactory = new ExcelFactory();

Document excel = excelfactory.createDocument();

excel.access();

}

}

---------------------------------------------------------------------------------------------------------------------------