**WEEK-01**

**MODULE-01:DESIGN PATTERN AND PRINCIPLES**

**MANDATORY HANDSON**

**EXERCISE -01: Singleton Pattern**

Logger.java

**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);

}

}

TestLogger.java

**public** **class** TestLogger {

**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 logger1 and logger2 are the same instance.");

} **else** {

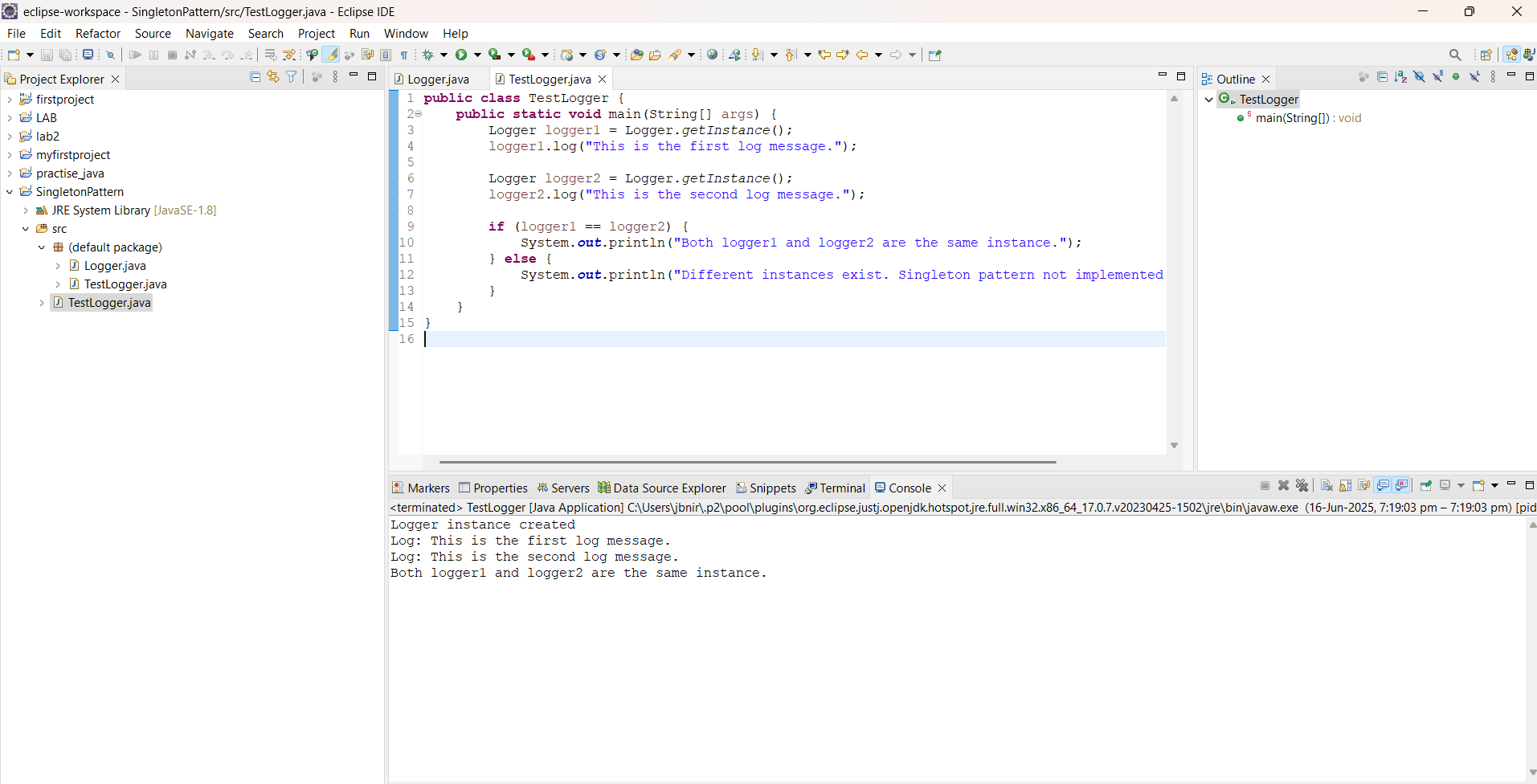
System.***out***.println("Different instances exist. Singleton pattern not implemented correctly.");

}

}

}

**OUTPUT:**

****



**EXERCISE -02: Factory Method Pattern**

Document.java

public interface Document {

void open();

}

WordDocument.java

public class WordDocument implements Document {

@Override

public void open() {

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

}

}

PdfDocument.java

public class PdfDocument implements Document {

@Override

public void open() {

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

}

}

ExcelDocument.java

public class ExcelDocument implements Document {

@Override

public void open() {

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

}

}

DocumentFactory.java

public abstract class DocumentFactory {

public abstract Document createDocument();

}

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();

}

}

TestFactory.java

import java.util.Scanner;

public class TestFactory {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

DocumentFactory factory = null;

System.out.println("Select Document Type: ");

System.out.println("1. Word Document");

System.out.println("2. PDF Document");

System.out.println("3. Excel Document");

System.out.print("Enter your choice (1/2/3): ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

factory = new WordDocumentFactory();

break;

case 2:

factory = new PdfDocumentFactory();

break;

case 3:

factory = new ExcelDocumentFactory();

break;

default:

System.out.println("Invalid choice.");

System.exit(0);

}

Document document = factory.createDocument();

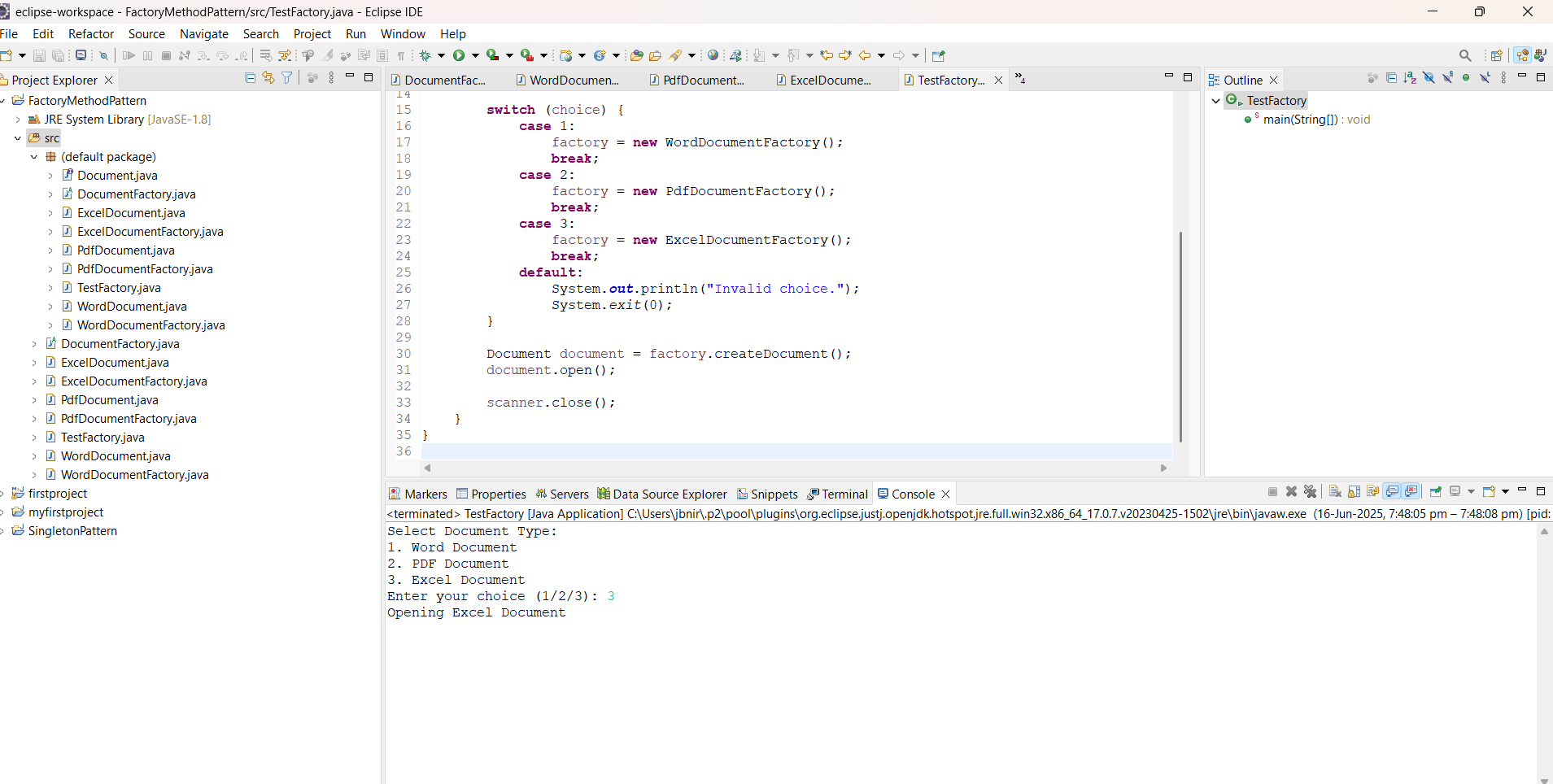
document.open();

scanner.close();

}

}

**OUTPUT:**

****

