**DESIGN PATTERNS AND PRINCIPLES**

**Exercise 1: Implementing the Singleton Pattern**

1. **Logger class**

package singleton;

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;

} // Sample logging

public void log(String message) {

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

} }

1. **Main.java**

package singleton;

public class Main {

public static void main(String[] args) {

Logger log1 = Logger.getInstance();

log1.log("Hi I’m Abinaya");

Logger log2 = Logger.getInstance();

log2.log("From R.M.K Engineering college");

if (log1 == log2) {

System.out.println("Both are the same instance (Singleton confirmed).");

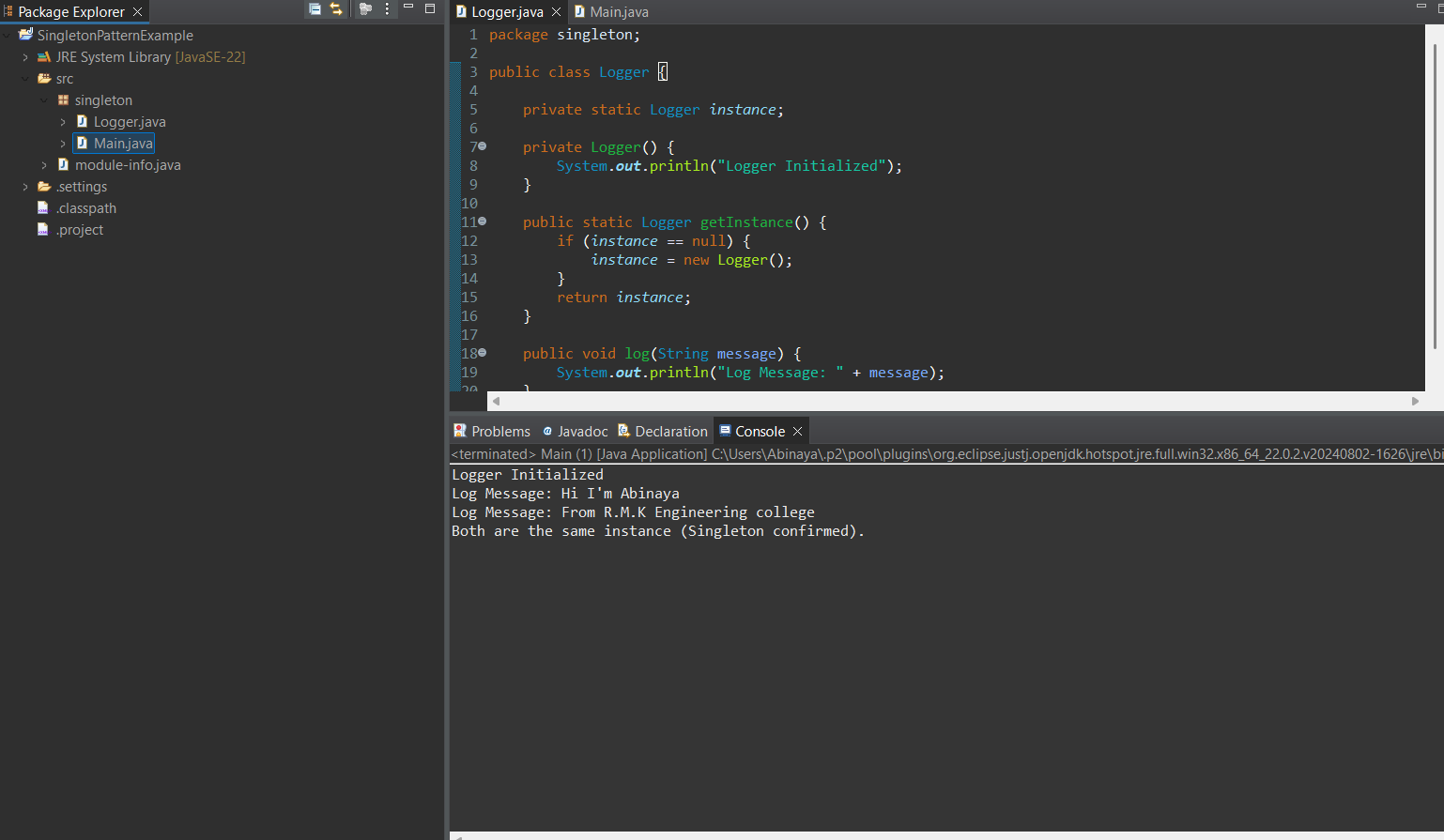
} else {

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

}

}

}

**Output :**

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

1. **Document.java**

package factorymethod;

public interface Document {

void open();

}

1. **WordDocument.java**

package factorymethod;

public class WordDocument implements Document {

@Override

public void open() {

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

}

}

1. **PdfDocument.java**

package factorymethod;

public class PdfDocument implements Document {

@Override

public void open() {

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

}

}

1. **ExcelDocument.java**

package factorymethod;

public class ExcelDocument implements Document {

@Override

public void open() {

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

}

}

1. **DocumentFactory.java**

package factorymethod;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

1. **WordDocumentFactory.java**

package factorymethod;

public class WordDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new WordDocument();

}

}

1. **PdfDocumentFactory.java**

package factorymethod;

public class PdfDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new PdfDocument();

}

}

1. **ExcelDocumentFactory.java**

**package factorymethod;**

public class ExcelDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new ExcelDocument();

}

}

1. **Main.java**

package factorymethod;

public class Main {

public static void main(String[] args) {

// Create Word document

DocumentFactory wordFactory = new WordDocumentFactory();

Document wordDoc = wordFactory.createDocument();

wordDoc.open();

// Create PDF document

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

// Create Excel document

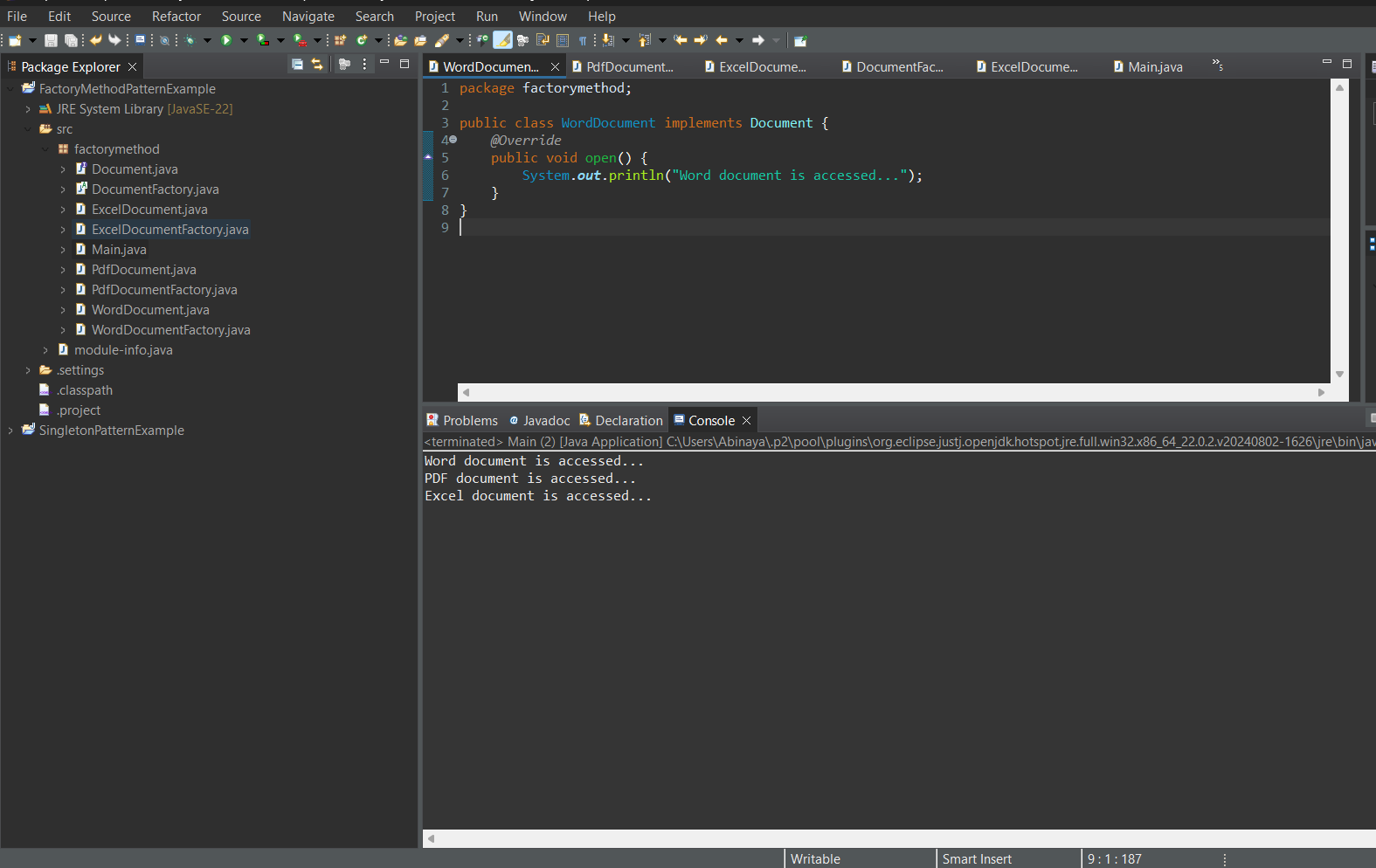
DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

**Output:**