**1.SingletonPatternExample**

**Logger.java**

package com.singleton;

public class Logger {

    private static Logger singleInstance;

    private Logger() {

        System.out.println("Logger initialized.");

    }

    public static Logger getInstance() {

        if (singleInstance == null) {

            singleInstance = new Logger();

        }

        return singleInstance;

    }

    public void logMessage(String message) {

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

    }

}

**LoggerTest.java**

package com.singleton;

public class LoggerTest {

    public static void main(String[] args) {

        Logger loggerA = Logger.getInstance();

        Logger loggerB = Logger.getInstance();

        loggerA.logMessage("Starting the application");

        loggerB.logMessage("Processing request");

        if (loggerA == loggerB) {

            System.out.println("Both references point to the same Logger instance.");

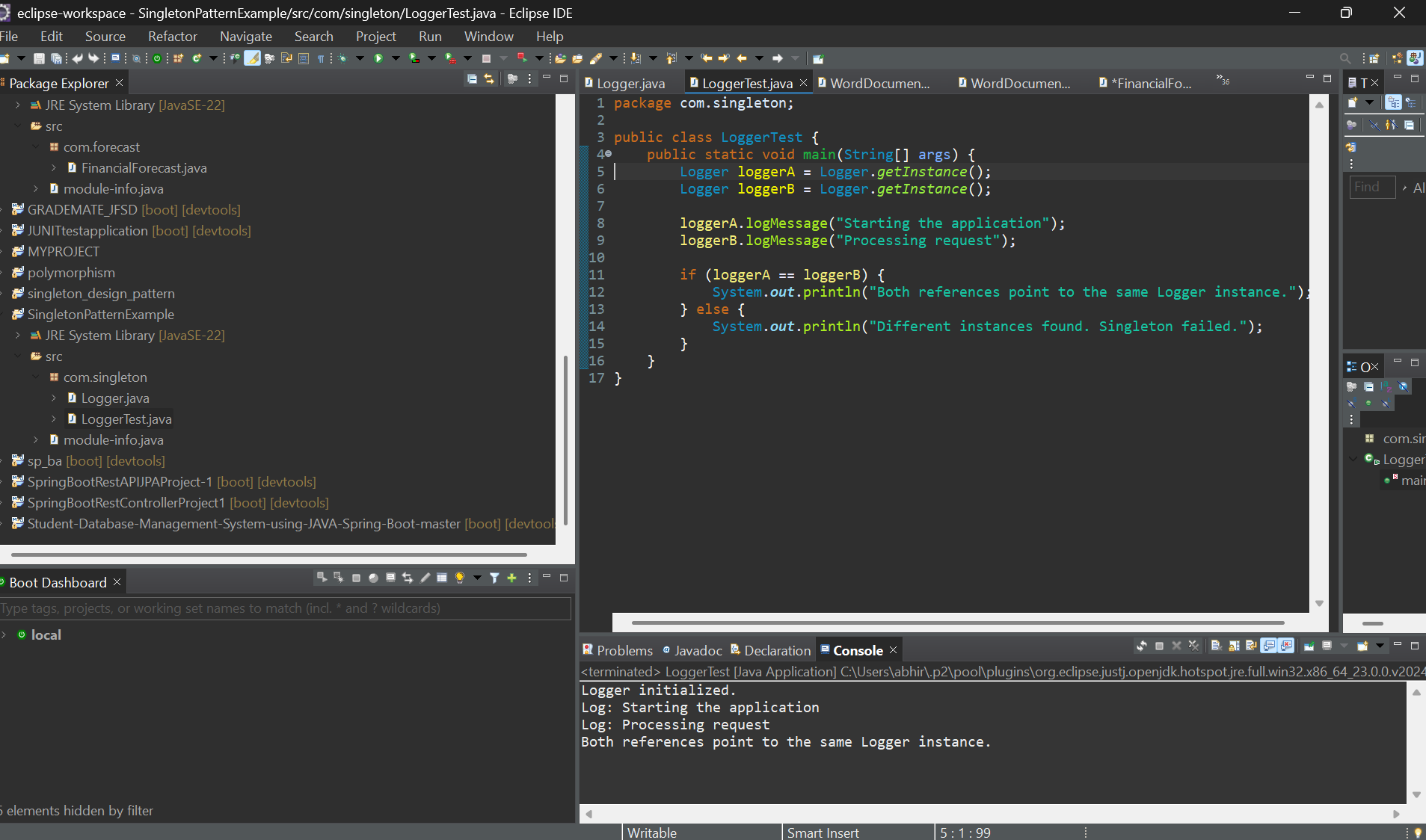
        } else {

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

        }

    }

}

**Output:** 

**2.FactoryMethodPatternExample**

**Document.java**

package com.factorymethodpattern;

public interface Document {

    void open();

}

**DocumentFactory.java**

package com.factorymethodpattern;

public abstract class DocumentFactory {

    public abstract Document createDocument();

}

**ExcelDocument.java**

package com.factorymethodpattern;

public class ExcelDocument implements Document {

    public void open() {

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

    }

}

**ExcelDocumentFactory.java**

**package com.factorymethodpattern;**

**public class ExcelDocumentFactory extends DocumentFactory {**

**public Document createDocument() {**

**return new ExcelDocument(); }**

**PdfDocument.java**

package com.factorymethodpattern;

public class PdfDocument implements Document {

    public void open() {

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

    }

}

**PdfDocumentFactory.java**

package com.factorymethodpattern;

public class PdfDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new PdfDocument();

    }

}

**WordDocument.java**

package com.factorymethodpattern;

public class WordDocument implements Document {

    public void open() {

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

    }

}

**WordDocumentFactory.java**

package com.factorymethodpattern;

public class WordDocumentFactory extends DocumentFactory {

    public Document createDocument() {

return new WordDocument();

    } }

**Test.java**

package com.factorymethodpattern;

public class Test {

    public static void main(String[] args) {

        DocumentFactory wordFactory = new WordDocumentFactory();

        Document word = wordFactory.createDocument();

        word.open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        Document pdf = pdfFactory.createDocument();

        pdf.open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        Document excel = excelFactory.createDocument();

        excel.open();

    } }

**Output:**

A screenshot of a computer program

AI-generated content may be incorrect.