**QUESTION :**

**Exercise 1: Implementing the Singleton Pattern**

**CODE:**

**APP.JAVA:**

public class App {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        logger1.log("First message");

        Logger logger2 = Logger.getInstance();

        logger2.log("Second message");

        if (logger1 == logger2) {

            System.out.println("Same Logger instance used.");

        } else {

            System.out.println("Different Logger instances!");

        }

    }

}

public class App {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        logger1.log("First message");

        Logger logger2 = Logger.getInstance();

        logger2.log("Second message");

        if (logger1 == logger2) {

            System.out.println("Same Logger instance used.");

        } else {

            System.out.println("Different Logger instances!");

        }

    }

}

LOGGER:

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;

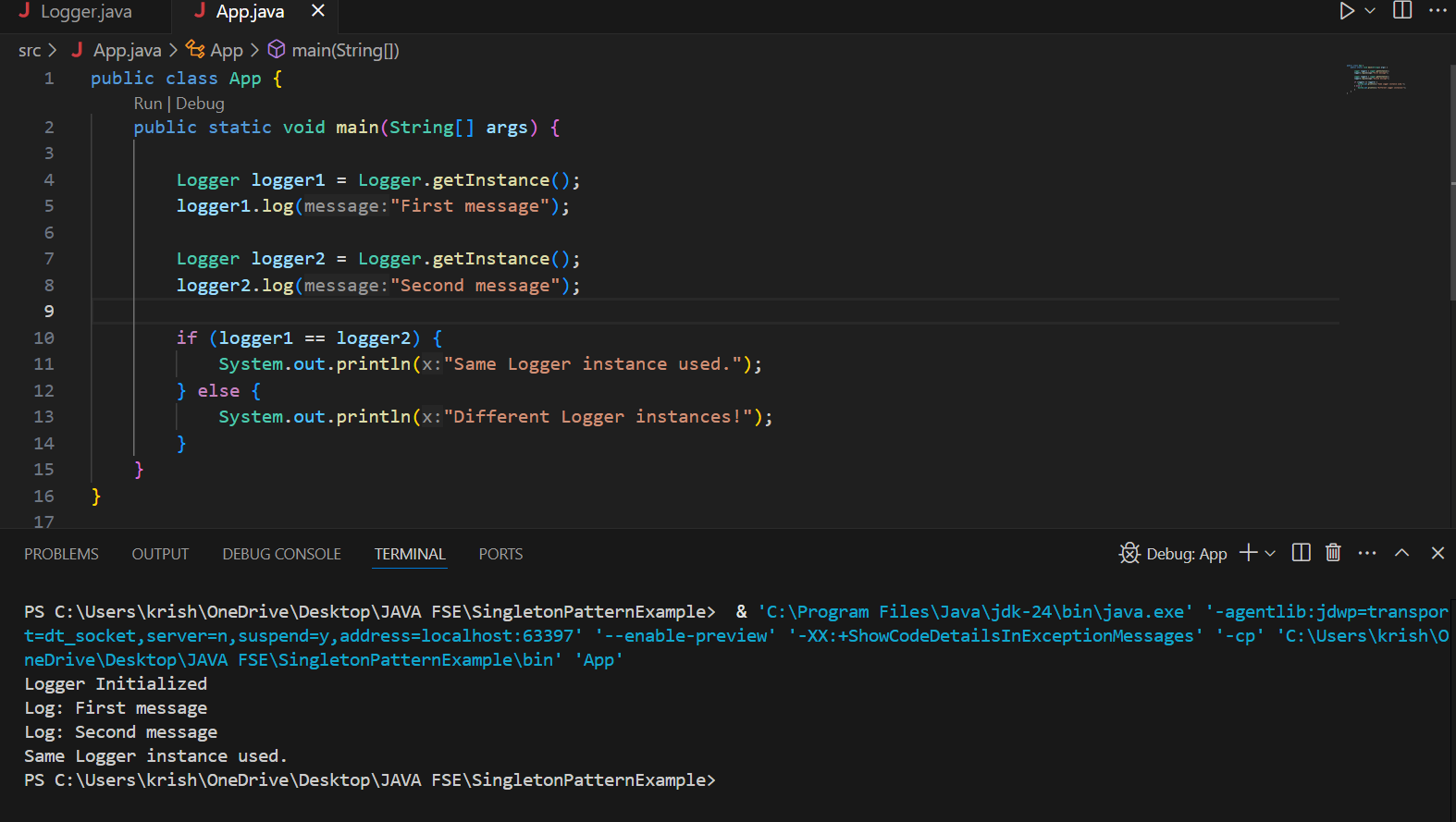
    }

    public void log(String message) {

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

    }

}



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

CODE:

Document:

public interface Document {

void open();

}

Word document:

public class WordDocument implements Document {

public void open() {

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

}

}

Pdfdocument:

public class PdfDocument implements Document {

public void open() {

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

}

}

Excel document:

public class ExcelDocument implements Document {

public void open() {

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

}

}

Document factory:

public abstract class DocumentFactory {

public abstract Document createDocument();

}

Wordfactory:

public class WordFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

Pdffactory:

public class PdfFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

Excel factory:

public class ExcelFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

Main:

public class Main {

    public static void main(String[] args) {

        DocumentFactory wordFactory = new WordFactory();

        Document wordDoc = wordFactory.createDocument();

        wordDoc.open();

        DocumentFactory pdfFactory = new PdfFactory();

        Document pdfDoc = pdfFactory.createDocument();

        pdfDoc.open();

        DocumentFactory excelFactory = new ExcelFactory();

        Document excelDoc = excelFactory.createDocument();

        excelDoc.open();

    }

}

A screenshot of a computer program

AI-generated content may be incorrect.