**DESIGN PATTERN AND PRINCIPLES:**

**Exercise 1: Implementing the Singleton Pattern Example**

**Code Implementation**:

* **Logger .java -**

**package** com.singleton;

**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 –**

**package** com.singleton;

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

**public** **static** **void** main(String[] args) {

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

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

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

**if** (logger1 == logger2) {

System.***out***.println("Both logger instances are the same (Singleton verified).");

} **else** {

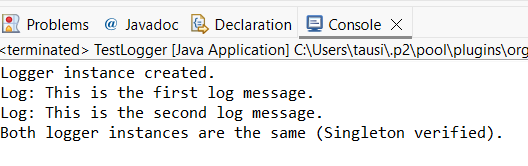
System.***out***.println("Different logger instances exist (Singleton failed).");

}

}

}

Output :



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

**Code Implementation:**

* **IDocument.java –**

**package** com.Factory;

**public** **interface** IDocument {

**void** open();

}

* **WordDocument.java –**

**package** com.Factory;

**public** **class** WordDocument **implements** IDocument {

@Override

**public** **void** open() {

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

}

}

* **PdfDocument.java –**

**package** com.Factory;

**public** **class** PdfDocument **implements** IDocument {

@Override

**public** **void** open() {

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

}

}

* **ExcelDocument.java –**

**package** com.Factory;

**public** **class** ExcelDocument **implements** IDocument {

@Override

**public** **void** open() {

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

}

}

* **DocumentFactory.java –**

**package** com.Factory;

**public** **abstract** **class** DocumentFactory {

**public** **abstract** IDocument createDocument();

}

* **WordDocumentFactory.java –**

**package** com.Factory;

**public** **class** WordDocumentFactory **extends** DocumentFactory {

@Override

**public** IDocument createDocument() {

**return** **new** WordDocument();

}

}

* **PdfDocumentFactory.java –**

**package** com.Factory;

**public** **class** PdfDocumentFactory **extends** DocumentFactory {

@Override

**public** IDocument createDocument() {

**return** **new** PdfDocument();

}

}

* **ExcelDocumentFactory.java –**

**package** com.Factory;

**public** **class** ExcelDocumentFactory **extends** DocumentFactory {

@Override

**public** IDocument createDocument() {

**return** **new** ExcelDocument();

}

}

* **DocumentFactoryTest.java –**

**package** com.Factory;

**public** **class** DocumentFactoryTest {

**public** **static** **void** main(String[] args) {

// Word document

DocumentFactory wordFactory = **new** WordDocumentFactory();

IDocument wordDoc = wordFactory.createDocument();

wordDoc.open();

// PDF document

DocumentFactory pdfFactory = **new** PdfDocumentFactory();

IDocument pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

// Excel document

DocumentFactory excelFactory = **new** ExcelDocumentFactory();

IDocument excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

Output:

A screenshot of a computer

AI-generated content may be incorrect.