**Design Principles and Patterns**

**Exercise 1: Implementing the Singleton Pattern**

**CODE:**

1. **Logger.java**

**package** com.example.singleton;

**public** **class** Logger {

// Step 1: Create a private static instance of the same class

**private** **static** Logger *instance*;

// Step 2: Make the constructor private so others can't use 'new'

**private** Logger() {

System.***out***.println("Logger Initialized");

}

// Step 3: Provide a public static method to get the instance

**public** **static** Logger getInstance() {

**if** (*instance* == **null**) {

*instance* = **new** Logger(); // Create only if not already created

}

**return** *instance*;

}

// Method to log messages

**public** **void** log(String message) {

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

}

}

1. **Main.java**

**package** com.example.singleton;

**public** **class** Main {

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

// Getting two instances of Logger

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

// Using both instances

logger1.log("First message.");

logger2.log("Second message.");

// Check if both logger instances are the same

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

System.***out***.println("Both loggers are the same instance.");

} **else** {

System.***out***.println("Different instances were created.");

}

}

}

**OUTPUT:**

A screenshot of a computer

AI-generated content may be incorrect.

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

**CODE:**

1. **Document.java**

**package** com.example.factorymethod;

**public** **interface** Document {

**void** open();

}

1. **WordDocument.java**

**package** com.example.factorymethod;

**public** **class** WordDocument **implements** Document {

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

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

}

}

1. **PdfDocument.java**

**package** com.example.factorymethod;

**public** **class** PdfFactory **extends** DocumentFactory {

**public** Document createDocument() {

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

}

}

1. **ExcelDocument.java**

**package** com.example.factorymethod;

**public** **class** ExcelDocument **implements** Document {

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

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

}

}

1. **DocumentFactory.java**

**package** com.example.factorymethod;

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

**public** **abstract** Document createDocument();

}

1. **WordFactory.java**

**package** com.example.factorymethod;

**public** **class** WordFactory **extends** DocumentFactory {

**public** Document createDocument() {

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

}

}

1. **PdfFactoty.java**

**package** com.example.factorymethod;

**public** **class** PdfFactory **extends** DocumentFactory {

**public** Document createDocument() {

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

}

}

1. **ExcelFactory.java**

**package** com.example.factorymethod;

**public** **class** ExcelFactory **extends** DocumentFactory {

**public** Document createDocument() {

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

}

}

1. **Main.java**

**package** com.example.factorymethod;

**public** **class** Main {

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

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

Document word = wordFactory.createDocument();

word.open();

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

Document pdf = pdfFactory.createDocument();

pdf.open();

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

Document excel = excelFactory.createDocument();

excel.open();

}

}

**OUTPUT:**

A screenshot of a computer

AI-generated content may be incorrect.