SingletonPatternExample

TestLogger.java

public class TestLogger {

    public static void main(String[] args) {

        // Attempt to get Logger instance multiple times

        Logger logger1 = Logger.getInstance();

        Logger logger2 = Logger.getInstance();

        // Use the logger

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

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

        // Check if both references point to the same object

        if (logger1 == logger2) {

            System.out.println("Only one Logger instance is used across the application.");

        } else {

            System.out.println("Multiple instances of Logger exist.");

        }

    }

}

Logger.Java

public class Logger {

    // Step 1: Create a private static instance

    private static Logger instance;

    // Step 2: Private constructor to prevent instantiation

    private Logger() {

        System.out.println("Logger instance created");

    }

    // Step 3: Public static method to provide access to the instance

    public static Logger getInstance() {

        if (instance == null) {

            instance = new Logger();

        }

        return instance;

    }

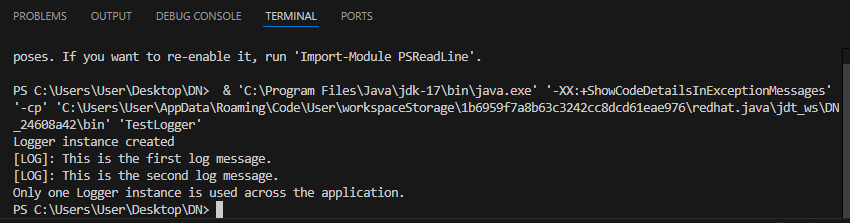
    // Example logging method

    public void log(String message) {

        System.out.println("[LOG]: " + message);

    }

}



FactoryMethodPatternExample

Document.java

package FactoryMethodPatternExample;

public interface Document {

    void open();

}

DocumentFactory.java

package FactoryMethodPatternExample;

public abstract class DocumentFactory {

    public abstract Document createDocument();

}

ExcelDocument.java

package FactoryMethodPatternExample;

public class ExcelDocument implements Document {

    public void open() {

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

    }

}

ExcelDocumentFactory.java

package FactoryMethodPatternExample;

public class ExcelDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new ExcelDocument();

    }

}

FactoryPatternTest.java

package FactoryMethodPatternExample;

public class FactoryPatternTest {

    public static void main(String[] args) {

        DocumentFactory wordFactory = new WordDocumentFactory();

        Document wordDoc = wordFactory.createDocument();

        wordDoc.open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        Document pdfDoc = pdfFactory.createDocument();

        pdfDoc.open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        Document excelDoc = excelFactory.createDocument();

        excelDoc.open();

    }

}

PdfDocument.java

package FactoryMethodPatternExample;

public class PdfDocument implements Document {

    public void open() {

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

    }

}

PdfDocumentFactory.java

package FactoryMethodPatternExample;

public class PdfDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new PdfDocument();

    }

}

WordDocument.java

package FactoryMethodPatternExample;

public class WordDocument implements Document {

    public void open() {

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

    }

}

WordDocumentFactory.java

package FactoryMethodPatternExample;

public class WordDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new WordDocument();

    }

}

