# **Assignment: Exercise 2 – Factory Method Pattern: Document Management System**

## **1. Creating a New Java Project**

For this task, I created a new Java project in my IDE and named it **FactoryMethodPatternExample**. The main goal of this project was to learn and implement the **Factory Method Pattern** by building a small document management system that can create different types of documents like Word, PDF, and Excel.

## **2. Setup: Creating a Document Interface**

I started by creating a simple interface called Document. This interface has only one method for now, which is open(). All the different document types will implement this interface.

public interface Document {

void open();

}

## **3. Implementation: Concrete Document Classes**

Next, I created three different classes: WordDocument, PdfDocument, and ExcelDocument. Each class implements the Document interface and gives its own version of the open() method.

public class WordDocument implements Document {

@Override

public void open() {

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

}

}

public class PdfDocument implements Document {

@Override

public void open() {

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

}

}

public class ExcelDocument implements Document {

@Override

public void open() {

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

}

}

## **4. Factory Method: DocumentFactory and Subclasses**

After creating the document types, I created an abstract class called DocumentFactory with an abstract method createDocument(). This method is supposed to return a Document, but the actual object creation is done by its subclasses.

public abstract class DocumentFactory {

public abstract Document createDocument();

}

Then I created three factory classes, one for each document type.

public class WordDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new WordDocument();

}

}

public class PdfDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new PdfDocument();

}

}

public class ExcelDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new ExcelDocument();

}

}

## **5. Testing the Factory Pattern**

To check if the Factory Method pattern is working properly, I created a Main class. In it, I used the factory classes to create each document and call the open() method on them.

public class Main {

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();

}

}

## **6. Output**

Opening Word Document...

Opening PDF Document...

Opening Excel Document...