**Factory Method Pattern**

**Cognizant Deepskilling**

By Vaishnavi K

21/6/2025

Project Name : FactoryMethodPattern

**Define Document Interface**

public interface Document {

void open();

}

**Create Concrete Document Classes**

**WordDocument.java**

public class WordDocument implements Document {

@Override

public void open() {

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

}

}

**PdfDocument.java**

public class PdfDocument implements Document {

@Override

public void open() {

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

}

}

**ExcelDocument.java**

public class ExcelDocument implements Document {

@Override

public void open() {

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

}

}

**Implement the Factory Method**

**Abstract Factory**

public abstract class DocumentFactory {

public abstract Document createDocument();

}

**Concrete Factories**

**WordDocumentFactory.java**

public class WordDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new WordDocument();

}

}

**PdfDocumentFactory.java**

public class PdfDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new PdfDocument();

}

}

**ExcelDocumentFactory.java**

public class ExcelDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new ExcelDocument();

}

}

**TestFactoryMethod.java**

public class TestFactoryMethod {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordDocumentFactory();

DocumentFactory pdfFactory = new PdfDocumentFactory();

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document word = wordFactory.createDocument();

Document pdf = pdfFactory.createDocument();

Document excel = excelFactory.createDocument();

word.open();

pdf.open();

excel.open();

}

}

Output:



