WEEK – 1

IMPLEMENTING THE FACTORY METHOD PATTERN

Nipuna Amanapu

[namanapu@gitam.in](mailto:namanapu@gitam.in)

superset id: 6432842

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

**Scenario:**

You are developing a document management system that needs to create different types of documents (e.g., Word, PDF, Excel). Use the Factory Method Pattern to achieve this.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **FactoryMethodPatternExample**.
2. **Define Document Classes:**
   * Create interfaces or abstract classes for different document types such as **WordDocument**, **PdfDocument**, and **ExcelDocument**.
3. **Create Concrete Document Classes:**
   * Implement concrete classes for each document type that implements or extends the above interfaces or abstract classes.
4. **Implement the Factory Method:**
   * Create an abstract class **DocumentFactory** with a method **createDocument()**.
   * Create concrete factory classes for each document type that extends DocumentFactory and implements the **createDocument()** method.
5. **Test the Factory Method Implementation:**
   * Create a test class to demonstrate the creation of different document types using the factory method.

Solution:

Project Name: FactoryMethodPatternExample

In FactoryMethodPatternExample I have created a sub folder src, under which I created another two sub folders Document and Factory , under which I have created files.

DOCUMENT

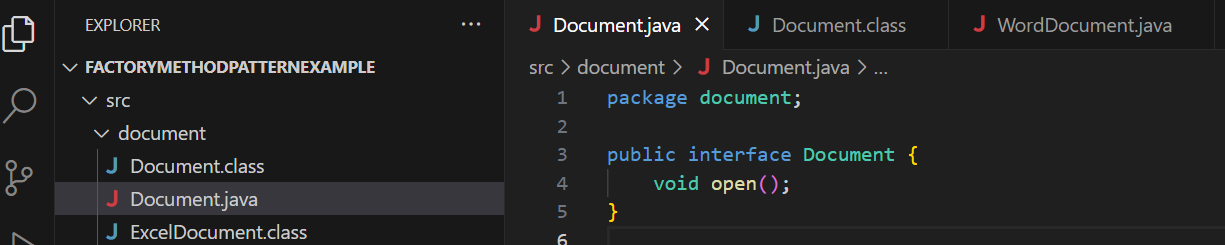
1. Document.java

package document;

public interface Document {

void open();

}



1. WordDocument.java

package document;

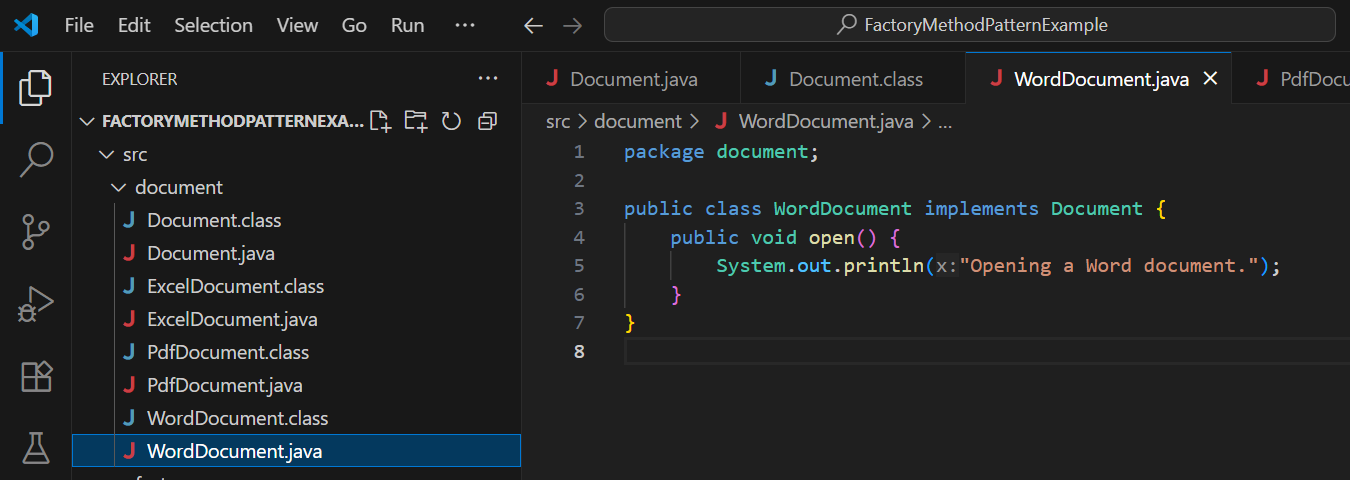
public class WordDocument implements Document {

public void open() {

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

}

}



1. PdfDocument.java

package document;

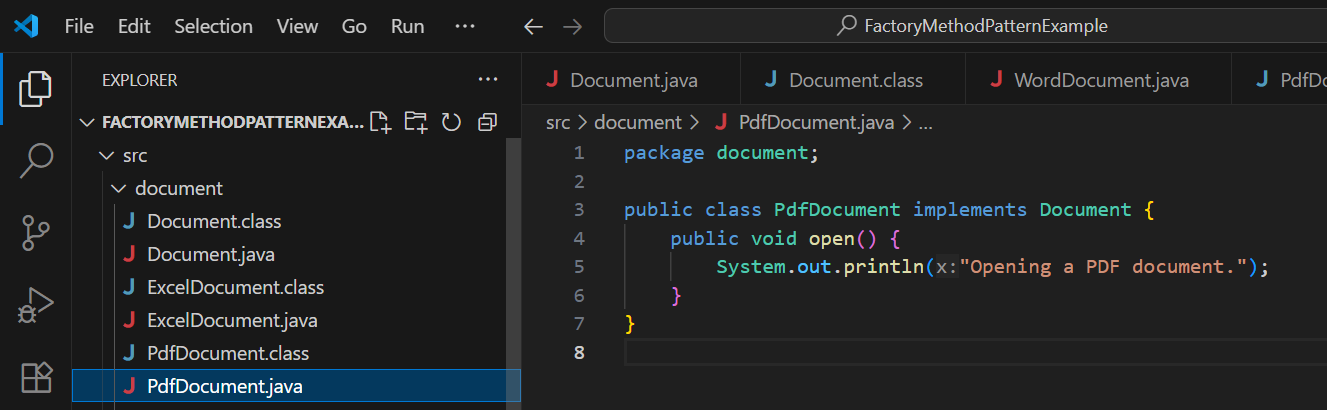
public class PdfDocument implements Document {

public void open() {

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

}

}



1. ExcelDocument.java

package document;

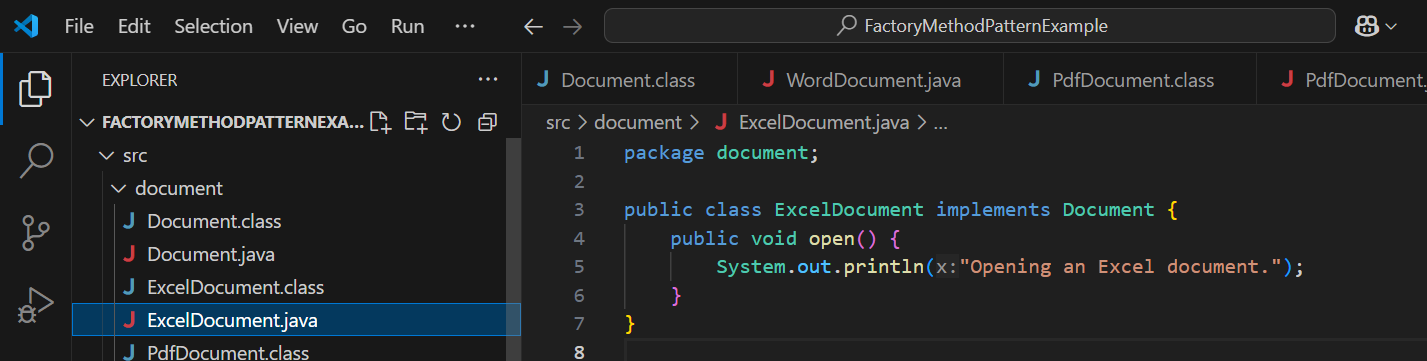
public class ExcelDocument implements Document {

public void open() {

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

}

}



FACTORY :

1)DocumentFactory.java

package factory;

import document.Document;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

2) WordDocumentFactory.java

package factory;

import document.Document;

import document.WordDocument;

public class WordDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

3) PdfDocumentFactory.java

package factory;

import document.Document;

import document.PdfDocument;

public class PdfDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

1. ExcelDocumentFactory.java

package factory;

import document.Document;

import document.ExcelDocument;

public class ExcelDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

Main.java (to test the factory pattern)

import factory.\*;

import document.Document;

public class Main {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordDocumentFactory();

Document word = wordFactory.createDocument();

word.open();

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdf = pdfFactory.createDocument();

pdf.open();

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excel = excelFactory.createDocument();

excel.open();

}

}

OUTPUT :

