**CTS DIGITAL NURTURE - 4.0 JAVA FSE**

**WEEK 1 - DESIGN PATTERNS AND PRINCIPLES**

MANDATORY HANDS-ON 2:

QUESTION:

**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.

PROGRAM:

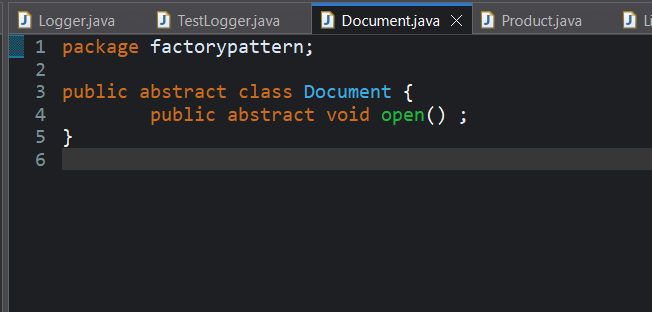
**Document.java**

package factorypattern;

public abstract class Document {

public abstract void open() ;

}

****

**ExcelDocument.java**

package factorypattern;

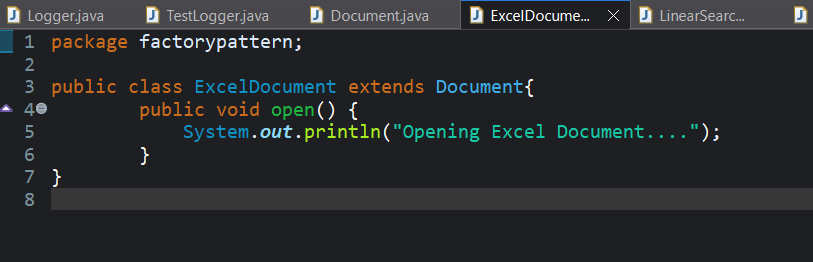
public class ExcelDocument extends Document{

public void open() {

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

}

}

****

**WordDocument.java**

package factorypattern;

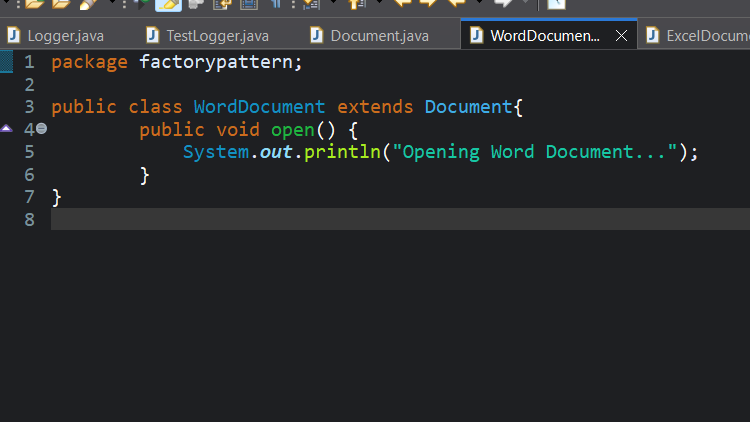
public class WordDocument extends Document{

public void open() {

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

}

}



**PdfDocument.java**

package factorypattern;

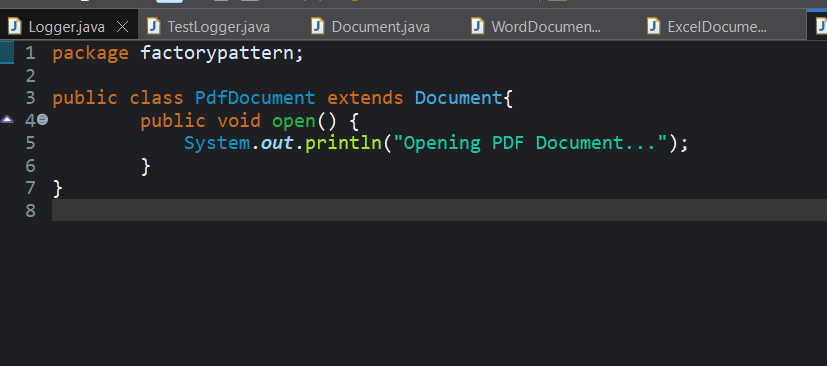
public class PdfDocument extends Document{

public void open() {

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

}

}

****

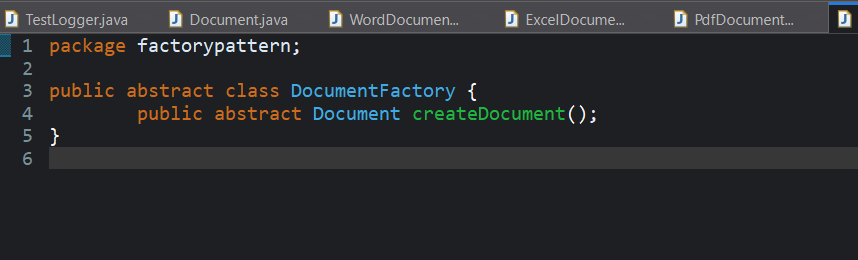
**DocumentFactory.java**

package factorypattern;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

****

**ExcelDocumentFactory.java**

package factorypattern;

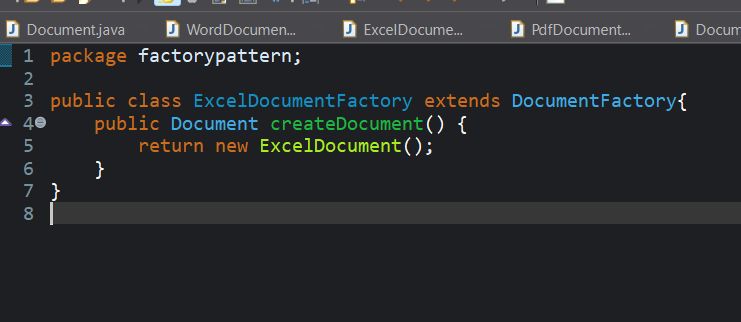
public class ExcelDocumentFactory extends DocumentFactory{

public Document createDocument() {

return new ExcelDocument();

}

}

****

**PdfDocumentFactory.java**

package factorypattern;

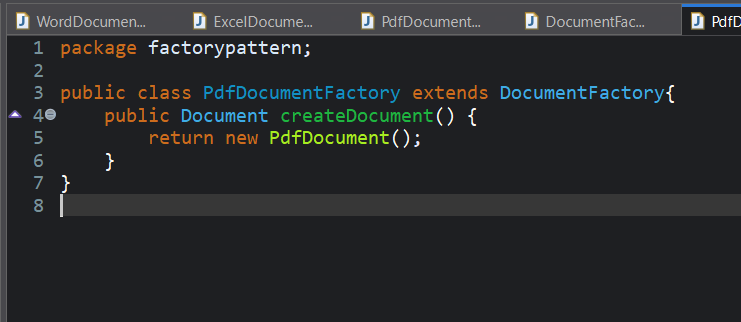
public class PdfDocumentFactory extends DocumentFactory{

public Document createDocument() {

return new PdfDocument();

}

}

****

**WordDocumentFactory.java**

package factorypattern;

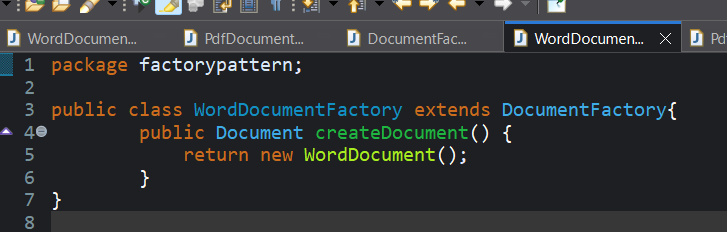
public class WordDocumentFactory extends DocumentFactory{

public Document createDocument() {

return new WordDocument();

}

}



**TestFactory.java**

package factorypattern;

public class TestFactory {

public static void main(String[] args) {

DocumentFactory documentFactory=null;

String str="pdf";

switch(str) {

case "word":{

documentFactory=new WordDocumentFactory();

break;

}

case "pdf":{

documentFactory=new PdfDocumentFactory();

break;

}

case "excel":{

documentFactory=new ExcelDocumentFactory();

break;

}

default:{

System.*out*.println("Invalid input");

}

}

if(documentFactory!=null) {

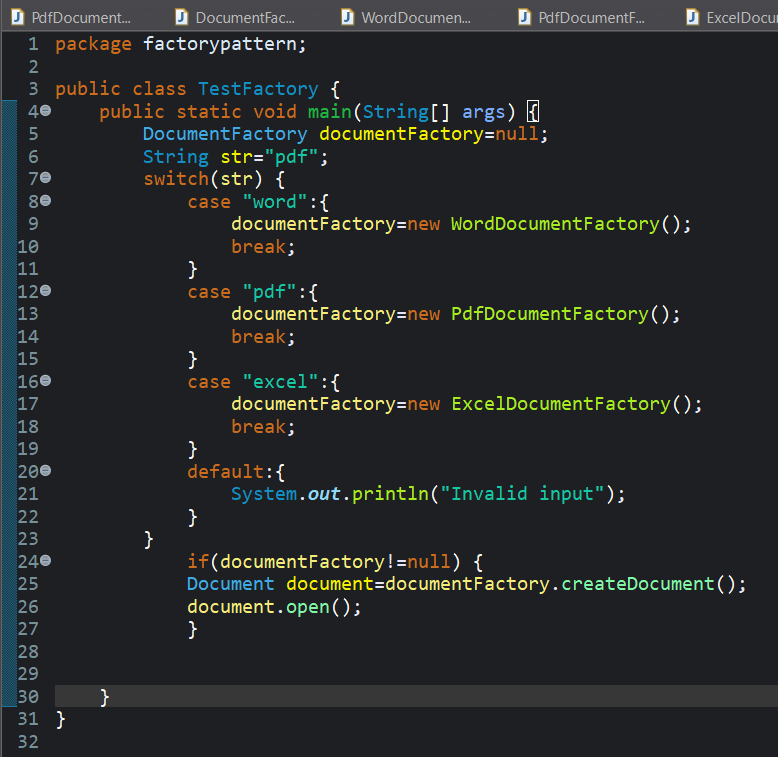
Document document=documentFactory.createDocument();

document.open();

}

}

}

****

OUTPUT:

Opening PDF Document...

