**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:**

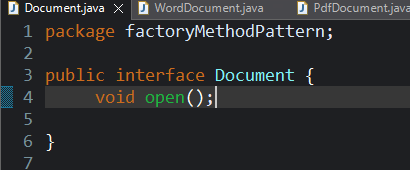
1. **Create a New Java Project:**
   * Create a new Java project named **FactoryMethodPatternExample**.
2. **Define Document Classes:**
   * Create interfaces or abstract classes for different document types such as **WordDocument**, **PdfDocument**, and **ExcelDocument**.
3. **Create Concrete Document 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:**

**1. Document Interface**

* A common interface Document defines the method open().
* All document types will implement this interface.

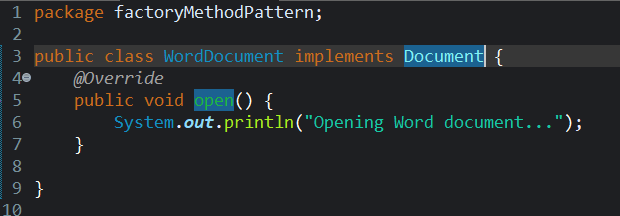
**Document.java**

****

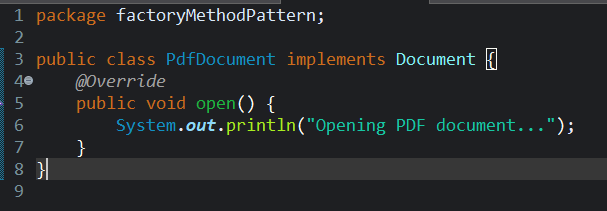
**2. Concrete Document Classes**

* Classes like WordDocument, PdfDocument, and ExcelDocument implement the Document interface.
* Each class provides its own logic for opening the document.

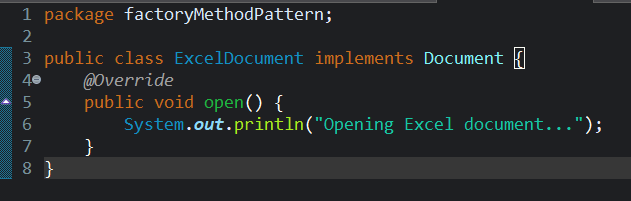
**WordDocument.java**

****

**PdfDocument.java**

****

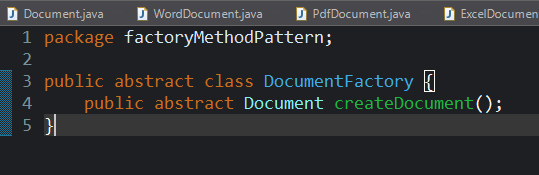
**ExcelDocument.java**

****

**3. Abstract Factory Class**

* DocumentFactory is an abstract class that declares the method createDocument().
* It doesn’t define how the document is created — it leaves that to subclasses.

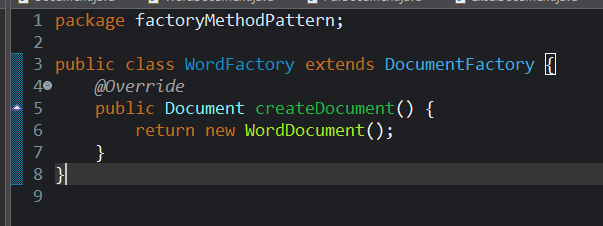
**DocumentFactory.java**

****

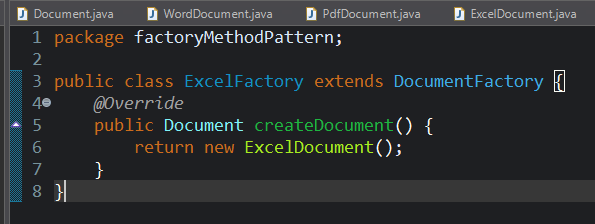
**4. Concrete Factories**

* WordFactory, PdfFactory, and ExcelFactory are subclasses of DocumentFactory.
* Each factory overrides createDocument() to return its specific type of document.

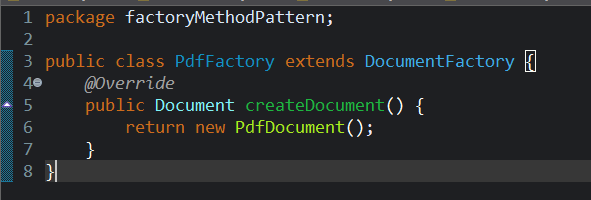
**WordFactory.java**

****

**ExcelFactory.java**

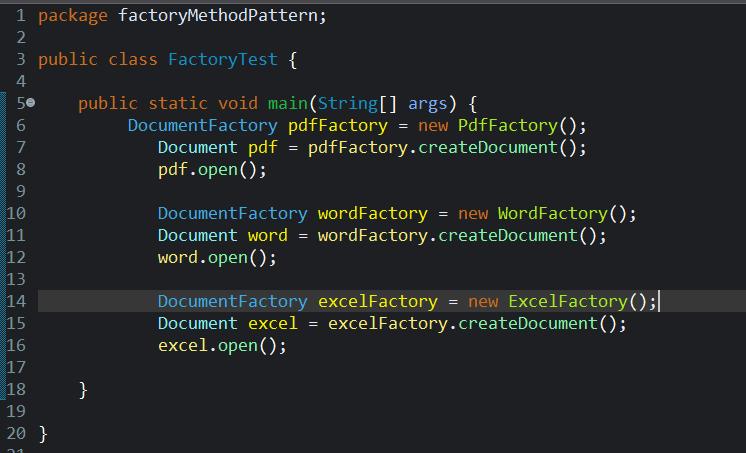
****

**PdfFactory.java**

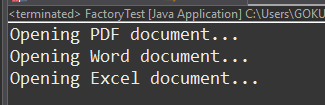
****

**5. Client/Test Code**

* The client/test class (FactoryTest) uses the specific factory (e.g., PdfFactory) to create the document.
* The client doesn't know or care how the document is created — only that it conforms to the Document interface



**Output:**

****