Cognizant Deep Nurture 4.0

Design Patterns and Principles

**Candidate Name:** Samarth Chourey  
**Superset ID:** 6394625

**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:**

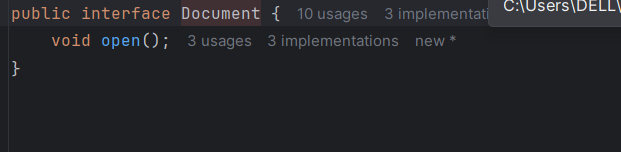
1. **Create a New Java Project:**
   * Create a new Java project named **FactoryMethodPatternExample**.
2. **Define Document Classes:**
   * Create interfaces or abstract classes for different document types such as **WordDocument**, **PdfDocument**, and **ExcelDocument**.
3. **Create Concrete Document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

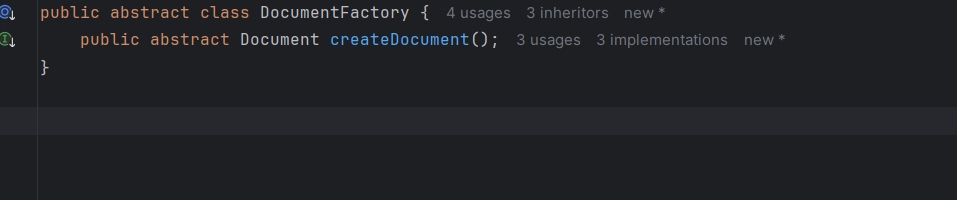
1. Created a new project named FactoryMethodPatternExample
2. Created Interfaces for **WordDocument**, **PdfDocument**, and **ExcelDocument**
3. Implemented Concrete Classes for the interfaces
4. Created DocumentFactory and Implemented Factory class for all Document Types
5. Tested Method Implementation

**Screenshots**

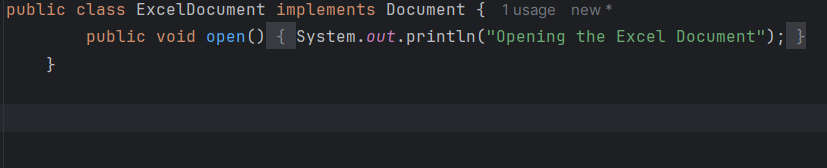
**Document.java**

****

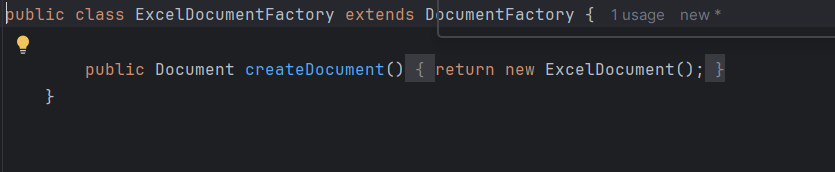
Document**Factory.java**



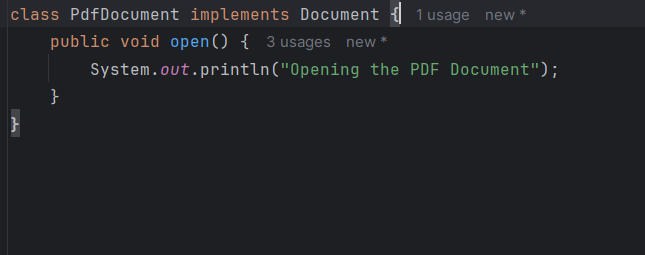
**ExcelDocument.java**



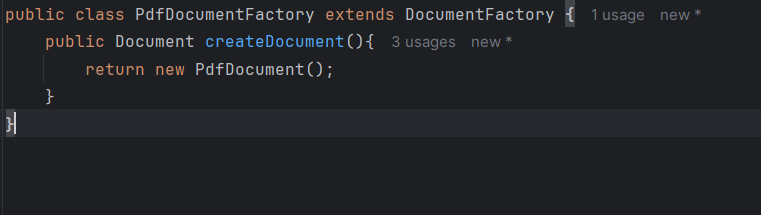
**ExcelDocumentFactory.java**

****

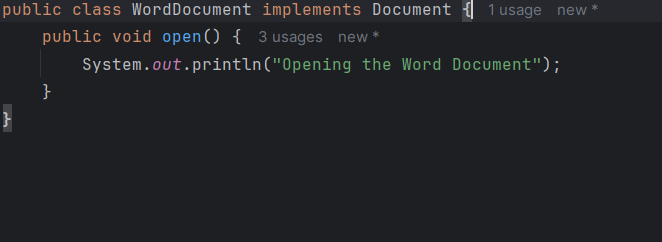
**PdfDocument.java**



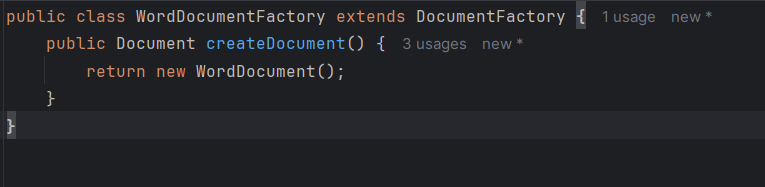
**PdfDocumentFactory.java**



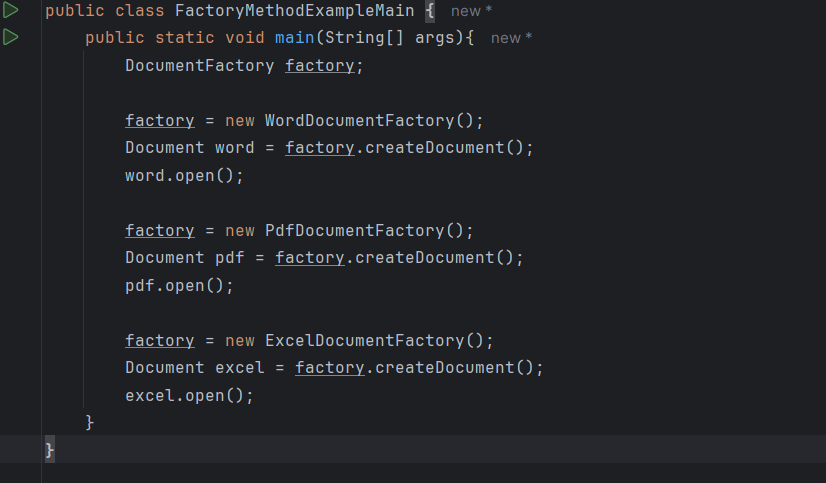
**WordDocument.java**



**WordDocumenFactory.java**



**FactoryMethodExampleMain.java**



**Output**

