**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.

**Document Interface**: The Document interface defines the common operations (open, save, close) that all document types must implement.

**Concrete Document Classes**: Each concrete class (WordDocument, PdfDocument, ExcelDocument) implements the Document interface.

**DocumentFactory Abstract Class**: The DocumentFactory abstract class defines the createDocument method that must be implemented by its subclasses.

**Concrete Factory Classes**: Each concrete factory class (WordDocumentFactory, PdfDocumentFactory, ExcelDocumentFactory) extends DocumentFactory and implements the createDocument method to return an instance of the respective document type.

**Testing the Factory Method**: In the FactoryMethodTest class, we demonstrate the creation of different document types using their respective factory classes and the subsequent operations on these documents.