**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**.

1. **Define Document Classes:**

* Create interfaces or abstract classes for different document types such as **WordDocument**, **PdfDocument**, and **ExcelDocument**.

1. **Create Concrete Document Classes:**

* Implement concrete classes for each document type that implements or extends the above interfaces or abstract classes.

1. **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.

1. **Test the Factory Method Implementation:**

* Create a test class to demonstrate the creation of different document types using the factory method.

**SOLUTION:**

**Step 1:** Create a New Java Project

Create a new Java project named FactoryMethodPatternExample.

**Step 2:** Define Document Classes:

abstract class Document

{

public abstract void open();

public abstract void close();

public abstract void save();

}

**Step 3:** Create Concrete Document Classes

class WordDocument extends Document {

@Override

public void open() {

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

}

@Override

public void close() {

System.out.println("Closing Word document...");

}

@Override

public void save() {

System.out.println("Saving Word document...");

}

}

class PdfDocument extends Document

{

@Override

public void open()

{

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

}

@Override

public void close()

{

System.out.println("Closing PDF document...");

}

@Override

public void save()

{

System.out.println("Saving PDF document...");

}

}

class ExcelDocument extends Document {

@Override

public void open() {

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

}

@Override

public void close() {

System.out.println("Closing Excel document...");

}

@Override

public void save() {

System.out.println("Saving Excel document...");

}

}

**Step 4:** Implement the Factory Method

abstract class DocumentFactory

{

public abstract Document createDocument();

}

class WordDocumentFactory extends DocumentFactory

{

@Override

public Document createDocument()

{

return new WordDocument();

}

}

class PdfDocumentFactory extends DocumentFactory

{

@Override

public Document createDocument()

{

return new PdfDocument();

}

}

class ExcelDocumentFactory extends DocumentFactory

{

@Override

public Document createDocument()

{

return new ExcelDocument();

}

}

**Step 5:** Test the Factory Method Implementation:

public class FactoryMethodPatternExample

{

public static void main(String[] args)

{

DocumentFactory wordFactory = new WordDocumentFactory();

Document wordDoc = wordFactory.createDocument();

wordDoc.open();

wordDoc.save();

wordDoc.close();

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

pdfDoc.save();

pdfDoc.close();

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excelDoc = excelFactory.createDocument();

excelDoc.open();

excelDoc.save();

excelDoc.close();

}

}

**SAMPLE OUTPUT:**

