**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.

Code –

package Design\_Patterns\_And\_Principles.FactoryMethodPatternExample;

// Interface for Document

interface Document {

    void open();

    void save();

}

// Abstract class for Document Factory

abstract class DocumentFactory {

    public abstract Document createDocument();

}

// Concrete class for Word Document

class WordDocument implements Document {

    @Override

    public void open() {

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

    }

    @Override

    public void save() {

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

    }

}

// Concrete class for PDF Document

class PdfDocument implements Document {

    @Override

    public void open() {

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

    }

    @Override

    public void save() {

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

    }

}

// Concrete class for Excel Document

class ExcelDocument implements Document {

    @Override

    public void open() {

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

    }

    @Override

    public void save() {

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

    }

}

// Concrete Document Factory for Word Document

class WordDocumentFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new WordDocument();

    }

}

// Concrete Document Factory for PDF Document

class PdfDocumentFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new PdfDocument();

    }

}

// Concrete Document Factory for Excel Document

class ExcelDocumentFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new ExcelDocument();

    }

}

// Test class to demonstrate the Factory Method

public class DocumentFactoryTest {

    public static void main(String[] args) {

        DocumentFactory wordFactory = new WordDocumentFactory();

        Document wordDoc = wordFactory.createDocument();

        wordDoc.open();

        wordDoc.save();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        Document pdfDoc = pdfFactory.createDocument();

        pdfDoc.open();

        pdfDoc.save();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        Document excelDoc = excelFactory.createDocument();

        excelDoc.open();

        excelDoc.save();

    }

}