**Exercise 2: Implementing the Factory Method Pattern**

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

Create a new Java project named FactoryMethodPatternExample using your preferred IDE.

**Step 2: Define Document Classes**

Define an interface or abstract class for different document types such as WordDocument, PdfDocument, and ExcelDocument.

// Document.java

public interface Document {

void open();

void save();

void close();

}

**Step 3: Create Concrete Document Classes**

Implement concrete classes for each document type that implements the Document interface.

// WordDocument.java

public class WordDocument implements Document {

@Override

public void open() {

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

}

@Override

public void save() {

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

}

@Override

public void close() {

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

}

}

// PdfDocument.java

public class PdfDocument implements Document {

@Override

public void open() {

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

}

@Override

public void save() {

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

}

@Override

public void close() {

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

}

}

// ExcelDocument.java

public class ExcelDocument implements Document {

@Override

public void open() {

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

}

@Override

public void save() {

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

}

@Override

public void close() {

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

}

}

**Step 4: Implement the Factory Method**

Create an abstract class DocumentFactory with a method createDocument(). Then, create concrete factory classes for each document type that extend DocumentFactory and implement the createDocument() method.

// DocumentFactory.java

public abstract class DocumentFactory {

public abstract Document createDocument();}

// WordDocumentFactory.java

public class WordDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new WordDocument(); }}

// PdfDocumentFactory.java

public class PdfDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new PdfDocument();}}

// ExcelDocumentFactory.java

public class ExcelDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new ExcelDocument(); }}

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

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

// FactoryMethodPatternExample.java

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();

}

}