**DESIGN PRINCIPLES & PATTERNS**

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

## **Code**:

### ***FactoryMethod.java:***

| public class FactoryMethod{   interface Document {  void open();  void close();  }   static class WordDocument implements Document {  public void open() {  System.out.println("Opening Word document.");  }  public void close() {  System.out.println("Closing Word document.");  }  }   static class PdfDocument implements Document {  public void open() {  System.out.println("Opening PDF document.");  }  public void close() {  System.out.println("Closing PDF document.");  }  }   static class ExcelDocument implements Document {  public void open() {  System.out.println("Opening Excel document.");  }  public void close() {  System.out.println("Closing Excel document.");  }  }   *//Abstract Factory*  static abstract class DocumentFactory {  public abstract Document createDocument();  }  *//Concrete Factories*  static class WordDocumentFactory extends DocumentFactory {  public Document createDocument() {  return new WordDocument();  }  }   static class PdfDocumentFactory extends DocumentFactory {  public Document createDocument() {  return new PdfDocument();  }  }   static class ExcelDocumentFactory extends DocumentFactory {  public Document createDocument() {  return new ExcelDocument();  }  }   *// Test Class*  public static void main(String[] args) {  *//Worddocument creation*  DocumentFactory wordFactory = new WordDocumentFactory();  Document wordDoc = wordFactory.createDocument();  wordDoc.open();  wordDoc.close();   *// PDF document creation*  DocumentFactory pdfFactory = new PdfDocumentFactory();  Document pdfDoc = pdfFactory.createDocument();  pdfDoc.open();  pdfDoc.close();   *//Excel document Creation*  DocumentFactory excelFactory = new ExcelDocumentFactory();  Document excelDoc = excelFactory.createDocument();  excelDoc.open();  excelDoc.close();  } } |
| --- |

## **Output Screenshot**: