// -🡪Product

package DesignPattern.FactoryMethodPatternExample;

public interface Document {

    void create();

}

public class ExcelDocument implements Document {

    public void create(){

        System.out.println("Creating Excel Document.");

    }

}

public class PdfDocument implements Document {

    public void create() {

        System.out.println("Creating PDF Document.");

    }

}

public class WordDocument implements Document {

    public void create() {

        System.out.println("creating Word Document.");

    }

}

// -🡪Creator (Factory Method)

package DesignPattern.FactoryMethodPatternExample;

public abstract class DocumentFactory {

    public abstract Document createDocument();

}

public class ExcelDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new ExcelDocument();

    }

}

public class PdfDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new PdfDocument();

    }

}

public class WordDocumentFactory extends DocumentFactory{

    public Document createDocument(){

        return new ExcelDocument();

    }

}

//🡪Main Class

package DesignPattern.FactoryMethodPatternExample;

public class Main {

    public static void main(String[] args) {

        System.out.println("-----------------");

        DocumentFactory wordFactory = new WordDocumentFactory();

        Document word = wordFactory.createDocument();

        word.create();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        Document pdf = pdfFactory.createDocument();

        pdf.create();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        Document excel = excelFactory.createDocument();

        excel.create();

        System.out.println("-----------------");

    }

}