Exercise 2: Implementing the Factory Method Pattern

Public interface Document {

void open();

}

Public class WordDocument implements Document {

@Override

public void open() {

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

}

}

Public class PdfDocument implements Document {

@Override

public void open() {

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

}

}

Public class ExcelDocument implements Document {

@Override

public void open() {

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

}

}

Public abstract class DocumentFactory {

public abstract Document createDocument();

}

Public class WordDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new WordDocument();

}

}

Public class PdfDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new PdfDocument();

}

}

Public class ExcelDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new ExcelDocument();

}

}

public class Main {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordDocumentFactory();

Document word = wordFactory.createDocument();

word.open();

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdf = pdfFactory.createDocument();

pdf.open();

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excel = excelFactory.createDocument();

excel.open();

}

}

Output:

A white background with black dots

AI-generated content may be incorrect.