Factory Method Pattern for Document Management System

public class FactoryMethodPatternExample {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordDocumentFactory();

DocumentFactory pdfFactory = new PdfDocumentFactory();

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document wordDoc = wordFactory.createDocument();

Document pdfDoc = pdfFactory.createDocument();

Document excelDoc = excelFactory.createDocument();

wordDoc.open();

wordDoc.save();

wordDoc.close();

pdfDoc.open();

pdfDoc.save();

pdfDoc.close();

excelDoc.open();

excelDoc.save();

excelDoc.close();

System.out.println("\nDemonstrating polymorphic behavior:");

Document[] documents = {wordDoc, pdfDoc, excelDoc};

for (Document doc : documents) {

doc.open();

doc.save();

doc.close();

System.out.println();

}

}

interface Document {

void open();

void save();

void close();

}

static class WordDocument implements Document {

@Override

public void open() {

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

}

@Override

public void save() {

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

}

@Override

public void close() {

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

}

}

static class PdfDocument implements Document {

@Override

public void open() {

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

}

@Override

public void save() {

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

}

@Override

public void close() {

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

}

}

static class ExcelDocument implements Document {

@Override

public void open() {

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

}

@Override

public void save() {

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

}

@Override

public void close() {

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

}

}

abstract static class DocumentFactory {

public abstract Document createDocument();

}

static class WordDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new WordDocument();

}

}

static class PdfDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new PdfDocument();

}

}

static class ExcelDocumentFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new ExcelDocument();

}

}

}

