**Strategy**

// Interface Estratégia

interface EstrategiaArmazenamento {

void salvar(String arquivo);

}

// Estratégia concreta - PDF

class ArmazenamentoPDF implements EstrategiaArmazenamento {

@Override

public void salvar(String arquivo) {

System.out.println("Salvando " + arquivo + " em formato PDF...");

}

}

// Estratégia concreta - DOCX

class ArmazenamentoDOCX implements EstrategiaArmazenamento {

@Override

public void salvar(String arquivo) {

System.out.println("Salvando " + arquivo + " em formato DOCX...");

}

}

// Estratégia concreta - TXT

class ArmazenamentoTXT implements EstrategiaArmazenamento {

@Override

public void salvar(String arquivo) {

System.out.println("Salvando " + arquivo + " em formato TXT...");

}

}

// Contexto

class GerenciadorArquivos {

private EstrategiaArmazenamento estrategia;

public void definirEstrategia(EstrategiaArmazenamento estrategia) {

this.estrategia = estrategia;

}

public void salvarArquivo(String arquivo) {

if (estrategia == null) {

System.out.println("Nenhuma estratégia definida!");

return;

}

estrategia.salvar(arquivo);

}

}

// Teste

public class Aplicacao {

public static void main(String[] args) {

GerenciadorArquivos gerenciador = new GerenciadorArquivos();

// TXT

gerenciador.definirEstrategia(new ArmazenamentoTXT());

gerenciador.salvarArquivo("relatorio.txt");

// PDF

gerenciador.definirEstrategia(new ArmazenamentoPDF());

gerenciador.salvarArquivo("contrato.pdf");

// DOCX

gerenciador.definirEstrategia(new ArmazenamentoDOCX());

gerenciador.salvarArquivo("curriculo.docx");

}

}