 **Documentação do Projeto Cadastro POO**

**1. Título da Prática**

Cadastro de Pessoas

**2. Objetivo da Prática**

O objetivo desta prática foi desenvolver um sistema de cadastro de pessoas físicas e jurídicas utilizando os conceitos de Programação Orientada a Objetos (POO) em Java. O sistema permitirá incluir, alterar, excluir, exibir e persistir dados em arquivos binários. Além disso, busca-se aplicar boas práticas de desenvolvimento, como a separação de responsabilidades através de classes de repositório.

**3. Códigos Implementados**

**Classe Pessoa (Superclasse Base)**

package model;

import java.io.Serializable;

public class Pessoa implements Serializable {

private int id;

private String nome;

public Pessoa() {}

public Pessoa(int id, String nome) {

this.id = id;

this.nome = nome;

}

public int getId() { return id; }

public void setId(int id) { this.id = id; }

public String getNome() { return nome; }

public void setNome(String nome) { this.nome = nome; }

public void exibir() {

System.out.println("ID: " + id + ", Nome: " + nome);

}

}

**Classe PessoaFisica (Herança de Pessoa)**

package model;

public class PessoaFisica extends Pessoa {

private String cpf;

private int idade;

public PessoaFisica(int id, String nome, String cpf, int idade) {

super(id, nome);

this.cpf = cpf;

this.idade = idade;

}

public String getCpf() { return cpf; }

public void setCpf(String cpf) { this.cpf = cpf; }

public int getIdade() { return idade; }

public void setIdade(int idade) { this.idade = idade; }

@Override

public void exibir() {

super.exibir();

System.out.println("CPF: " + cpf + ", Idade: " + idade);

}

}

**Classe PessoaJuridica (Herança de Pessoa)**

package model;

public class PessoaJuridica extends Pessoa {

private String cnpj;

public PessoaJuridica(int id, String nome, String cnpj) {

super(id, nome);

this.cnpj = cnpj;

}

public String getCnpj() { return cnpj; }

public void setCnpj(String cnpj) { this.cnpj = cnpj; }

@Override

public void exibir() {

super.exibir();

System.out.println("CNPJ: " + cnpj);

}

}

**Classe Main (Execução do Programa)**

package model;

import java.util.\*;

import java.io.\*;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

PessoaFisicaRepo repoFisica = new PessoaFisicaRepo();

PessoaJuridicaRepo repoJuridica = new PessoaJuridicaRepo();

int opcao;

do {

System.out.println("\nMenu:");

System.out.println("1 - Incluir");

System.out.println("2 - Exibir Todos");

System.out.println("6 - Salvar Dados");

System.out.println("7 - Recuperar Dados");

System.out.println("0 - Sair");

System.out.print("Escolha uma opção: ");

opcao = scanner.nextInt();

scanner.nextLine();

switch (opcao) {

case 1:

System.out.print("Tipo (1 - Física, 2 - Jurídica): ");

int tipo = scanner.nextInt();

scanner.nextLine();

System.out.print("ID: ");

int id = scanner.nextInt();

scanner.nextLine();

System.out.print("Nome: ");

String nome = scanner.nextLine();

if (tipo == 1) {

System.out.print("CPF: ");

String cpf = scanner.nextLine();

System.out.print("Idade: ");

int idade = scanner.nextInt();

scanner.nextLine();

repoFisica.inserir(new PessoaFisica(id, nome, cpf, idade));

} else {

System.out.print("CNPJ: ");

String cnpj = scanner.nextLine();

repoJuridica.inserir(new PessoaJuridica(id, nome, cnpj));

}

break;

case 2:

System.out.println("Pessoas Físicas:");

for (PessoaFisica p : repoFisica.obterTodos()) {

p.exibir();

}

System.out.println("Pessoas Jurídicas:");

for (PessoaJuridica p : repoJuridica.obterTodos()) {

p.exibir();

}

break;

case 6:

try {

repoFisica.persistir("pessoas\_fisicas.bin");

repoJuridica.persistir("pessoas\_juridicas.bin");

System.out.println("Dados salvos!");

} catch (IOException e) {

System.out.println("Erro ao salvar: " + e.getMessage());

}

break;

case 7:

try {

repoFisica.recuperar("pessoas\_fisicas.bin");

repoJuridica.recuperar("pessoas\_juridicas.bin");

System.out.println("Dados recuperados!");

} catch (Exception e) {

System.out.println("Erro ao recuperar: " + e.getMessage());

}

break;

case 0:

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

break;

default:

System.out.println("Opção inválida!");

}

} while (opcao != 0);

scanner.close();

}

}

**4. Análise e Conclusão**

1. **O que são elementos estáticos e qual o motivo para o método main adotar esse modificador?**
   * Elementos estáticos pertencem à classe e não a uma instância. O método main é estático porque é o ponto de entrada da aplicação e deve ser chamado sem criar um objeto da classe.
2. **Para que serve a classe Scanner?**
   * A classe Scanner é utilizada para capturar entrada do usuário via teclado.
3. **Como o uso de classes de repositório impactou na organização do código?**
   * Melhorou a separação de responsabilidades, tornando o código modular e facilitando a manutenção e a escalabilidade.