**Universidade Estácio de Sá**

**Polo Estácio EAD - Batatais / SP**

**Curso:** Desenvolvimento Full Stack  
**Disciplina:** Iniciando o Caminho pelo Java  
**Missão Prática - Nível 1  
Turma:** RPG0014  
**Semestre Letivo:** 2024.3  
 **Integrante:** Daiana Maira de Oliveira Lascala

**Título**

Implementação de um cadastro de clientes em modo texto, com persistência em arquivos, baseados na tecnologia Java

**Objetivo da Prática**

Criação do Cadastro em Modo Texto

**Descrição dos Códigos:**

O código desenvolvido neste projeto cria um sistema de cadastro de clientes (pessoas físicas e jurídicas) em modo texto com as seguintes funcionalidades:

1. **Menu Interativo:**  
   O sistema apresenta um menu onde o usuário pode selecionar opções como adicionar, alterar, excluir, exibir clientes, salvar e recuperar dados.
2. **Inserção de Dados:**  
   O sistema permite a inclusão de dados de clientes (Pessoa Física ou Jurídica) diretamente a partir do teclado.
3. **Alteração e Exclusão:**  
   Com base no ID do cliente, o sistema permite alterar ou excluir dados de pessoas físicas ou jurídicas.
4. **Persistência e Recuperação de Dados:**  
   Os dados são salvos e recuperados a partir de arquivos binários, garantindo a persistência das informações para futuros acessos.
5. **Tratamento de Exceções:**  
   Exceções são tratadas nas operações de persistência e recuperação, garantindo que o programa não quebre inesperadamente.

**Os resultados da execução dos códigos**

package model;

import java.io.IOException;

import java.util.Scanner;

public class Principal {

public static void main(String[] args) {

try (Scanner scanner = new Scanner(System.in)) {

PessoaFisicaRepo repoFisica = new PessoaFisicaRepo();

PessoaJuridicaRepo repoJuridica = new PessoaJuridicaRepo();

String prefixoArquivo = "";

int opcao = -1;

while (opcao != 0) {

System.out.println("===== Menu =====");

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

System.out.println("2. Alterar Pessoa");

System.out.println("3. Excluir Pessoa");

System.out.println("4. Buscar pelo ID");

System.out.println("5. Exibir todos");

System.out.println("6. Persistir dados");

System.out.println("7. Recuperar dados");

System.out.println("0. Finalizar Programa");

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

opcao = scanner.nextInt();

scanner.nextLine();

switch (opcao) {

case 1:

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

int tipo = scanner.nextInt();

scanner.nextLine();

if (tipo == 1) {

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

int id = scanner.nextInt();

scanner.nextLine();

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

String nome = scanner.nextLine();

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

String cpf = scanner.nextLine();

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

int idade = scanner.nextInt();

PessoaFisica pf = new PessoaFisica(id, nome, cpf, idade);

repoFisica.inserir(pf);

} else if (tipo == 2) {

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

int id = scanner.nextInt();

scanner.nextLine();

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

String nome = scanner.nextLine();

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

String cnpj = scanner.nextLine();

PessoaJuridica pj = new PessoaJuridica(id, nome, cnpj);

repoJuridica.inserir(pj);

}

break;

case 2:

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

tipo = scanner.nextInt();

scanner.nextLine();

if (tipo == 1) {

System.out.print("ID da pessoa a alterar: ");

int id = scanner.nextInt();

scanner.nextLine();

PessoaFisica pf = repoFisica.buscarPorId(id);

if (pf != null) {

System.out.println("Dados atuais: ID = " + pf.getId() + ", Nome = " + pf.getNome() + ", CPF = " + pf.getCpf() + ", Idade = " + pf.getIdade());

System.out.print("Novo nome: ");

String novoNome = scanner.nextLine();

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

String novoCpf = scanner.nextLine();

System.out.print("Nova idade: ");

int novaIdade = scanner.nextInt();

pf.setNome(novoNome);

pf.setCpf(novoCpf);

pf.setIdade(novaIdade);

} else {

System.out.println("Pessoa física não encontrada.");

}

} else if (tipo == 2) {

System.out.print("ID da empresa a alterar: ");

int id = scanner.nextInt();

scanner.nextLine();

PessoaJuridica pj = repoJuridica.buscarPorId(id);

if (pj != null) {

System.out.println("Dados atuais: ID = " + pj.getId() + ", Nome = " + pj.getNome() + ", CNPJ = " + pj.getCnpj());

System.out.print("Novo nome: ");

String novoNome = scanner.nextLine();

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

String novoCnpj = scanner.nextLine();

pj.setNome(novoNome);

pj.setCnpj(novoCnpj);

} else {

System.out.println("Pessoa jurídica não encontrada.");

}

}

break;

case 3:

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

tipo = scanner.nextInt();

scanner.nextLine();

if (tipo == 1) {

System.out.print("ID da pessoa física a excluir: ");

int id = scanner.nextInt();

scanner.nextLine();

repoFisica.remover(id);

} else if (tipo == 2) {

System.out.print("ID da pessoa jurídica a excluir: ");

int id = scanner.nextInt();

scanner.nextLine();

repoJuridica.remover(id);

}

break;

case 4:

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

tipo = scanner.nextInt();

scanner.nextLine();

if (tipo == 1) {

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

int id = scanner.nextInt();

scanner.nextLine();

PessoaFisica pf = repoFisica.buscarPorId(id);

if (pf != null) {

System.out.println(pf);

} else {

System.out.println("Pessoa física não encontrada.");

}

} else if (tipo == 2) {

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

int id = scanner.nextInt();

scanner.nextLine();

PessoaJuridica pj = repoJuridica.buscarPorId(id);

if (pj != null) {

System.out.println(pj);

} else {

System.out.println("Pessoa jurídica não encontrada.");

}

}

break;

case 5:

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

tipo = scanner.nextInt();

scanner.nextLine();

if (tipo == 1) {

for (PessoaFisica pf : repoFisica.exibirTodos()) {

System.out.println(pf);

}

} else if (tipo == 2) {

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

System.out.println(pj);

}

}

break;

case 6:

System.out.print("Digite o prefixo dos arquivos: ");

prefixoArquivo = scanner.nextLine();

try {

repoFisica.persistir(prefixoArquivo + ".fisica.bin");

repoJuridica.persistir(prefixoArquivo + ".juridica.bin");

} catch (IOException e) {

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

}

break;

case 7:

System.out.print("Digite o prefixo dos arquivos: ");

prefixoArquivo = scanner.nextLine();

try {

repoFisica.recuperar(prefixoArquivo + ".fisica.bin");

repoJuridica.recuperar(prefixoArquivo + ".juridica.bin");

} catch (IOException | ClassNotFoundException e) {

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

}

break;

case 0:

System.out.println("Encerrando o programa.");

break;

default:

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

break;

}

}

}

}

}

**Análise e Conclusão:**

1. **O que são elementos estáticos e qual o motivo para o método main adotar esse modificador?**
   * Elementos estáticos são aqueles pertencentes à classe e não às instâncias de objetos. O modificador static define que um método ou variável pode ser acessado sem a necessidade de instanciar a classe.
   * O método main é estático para que o programa possa ser iniciado sem que seja necessário criar um objeto da classe que o contém. Isso é importante para que a JVM possa chamar o método main diretamente ao executar o programa.
2. **Para que serve a classe Scanner?**
   * A classe Scanner é usada para ler a entrada do usuário, como os dados digitados no teclado. No nosso projeto, ela é utilizada para capturar informações como nome, CPF, CNPJ, idade e as opções do menu.
3. **Como o uso de classes de repositório impactou na organização do código?**
   * O uso de classes de repositório (como PessoaFisicaRepo e PessoaJuridicaRepo) tornou o código mais organizado e modular. Cada repositório é responsável pela gestão de um tipo específico de cliente (físico ou jurídico), o que facilita a manutenção e permite um controle mais eficiente sobre os dados, além de separar as responsabilidades de armazenamento do restante da lógica do sistema.

**Repositório Git:**

https://github.com/DaianaLascala/Missao-Nivel-1-Java-