



Estácio

Campus: Polo Cohama

Curso: Desenvolvimento FullStack

Turma: 9001

Disciplina: Iniciando o Caminho Pelo Java

Nome: João Victor Sá de Araújo

Relatório de Prática: Criação do Cadastro em Modo Texto

2. Objetivo da Prática

O objetivo da prática foi desenvolver um sistema em Java que utiliza conceitos fundamentais de programação orientada a objetos, como herança, encapsulamento, e modularidade, aplicando boas práticas de organização de código. A proposta incluiu a criação de classes representando entidades, repositórios para gerenciamento de dados e métodos para persistência e recuperação de informações em arquivos. Além disso, a prática buscou explorar a interação com o usuário por meio de um menu em modo texto, permitindo a manipulação dinâmica dos dados armazenados. Assim, foi possível consolidar o entendimento sobre manipulação de coleções, tratamento de exceções e persistência de dados.

3. Códigos

CadastroPOO\src\cadastropoo\CadastroPOO.java:

```
/*
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to
edit this template
 */
package cadastropoo;

import cadastropoo.model.PessoaFisica;
import cadastropoo.model.PessoaFisicaRepo;
import cadastropoo.model.PessoaJuridica;
import cadastropoo.model.PessoaJuridicaRepo;
import java.io.IOException;
import java.util.Scanner;

/**
 *
 * @author joao_
 */
public class CadastroPOO {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Repositórios
```

```

PessoaFisicaRepo repoFisica = new PessoaFisicaRepo();
PessoaJuridicaRepo repoJuridica = new PessoaJuridicaRepo();

int opcao;
do {
    // Menu de opções
    System.out.println("\n--- Menu ---");
    System.out.println("1. Incluir");
    System.out.println("2. Alterar");
    System.out.println("3. Excluir");
    System.out.println("4. Exibir pelo ID");
    System.out.println("5. 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(); // Consumir a quebra de linha

    switch (opcao) {
        case 1: // Incluir
            System.out.print("Tipo (1 - Física, 2 - Jurídica):");

            int tipo = scanner.nextInt();
            scanner.nextLine(); // Consumir a quebra de linha

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

            } else if (tipo == 2) {
                System.out.print("ID: ");
                int id = scanner.nextInt();
                scanner.nextLine();

```

```

        System.out.print("Nome: ");
        String nome = scanner.nextLine();
        System.out.print("CNPJ: ");
        String cnpj = scanner.nextLine();
        repoJuridica.inserir(new PessoaJuridica(id,
nome, cnpj));
    }
    break;

    case 2: // Alterar
        System.out.print("Tipo (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.obter(id);
            if (pf != null) {
                System.out.println("Dados atuais: ");
                pf.exibir();
                System.out.print("Novo Nome: ");
                String nome = scanner.nextLine();
                System.out.print("Novo CPF: ");
                String cpf = scanner.nextLine();
                System.out.print("Nova Idade: ");
                int idade = scanner.nextInt();
                scanner.nextLine();
                repoFisica.alterar(new PessoaFisica(id,
nome, cpf, idade));
            } 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.obter(id);
            if (pj != null) {
                System.out.println("Dados atuais: ");

```

```

        pj.exibir();
        System.out.print("Novo Nome: ");
        String nome = scanner.nextLine();
        System.out.print("Novo CNPJ: ");
        String cnpj = scanner.nextLine();
        repoJuridica.alterar(new PessoaJuridica(id,
nome, cnpj));

        } else {
            System.out.println("Pessoa Jurídica não
encontrada!");
        }
    }
    break;

case 3: // Excluir
    System.out.print("Tipo (1 - Física, 2 - Jurídica):
");

    tipo = scanner.nextInt();
    scanner.nextLine();
    System.out.print("ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();

    if (tipo == 1) {
        repoFisica.excluir(id);
        System.out.println("Pessoa Física removida!");
    } else if (tipo == 2) {
        repoJuridica.excluir(id);
        System.out.println("Pessoa Jurídica
removida!");
    }
    break;

case 4: // Exibir pelo ID
    System.out.print("Tipo (1 - Física, 2 - Jurídica):
");

    tipo = scanner.nextInt();
    scanner.nextLine();
    System.out.print("ID: ");
    id = scanner.nextInt();
    scanner.nextLine();

    if (tipo == 1) {

```

```

        PessoaFisica pf = repoFisica.obter(id);
        if (pf != null) {
            pf.exibir();
        } else {
            System.out.println("Pessoa Física não
encontrada!");
        }
    } else if (tipo == 2) {
        PessoaJuridica pj = repoJuridica.obter(id);
        if (pj != null) {
            pj.exibir();
        } else {
            System.out.println("Pessoa Jurídica não
encontrada!");
        }
    }
    break;

case 5: // Exibir todos
    System.out.print("Tipo (1 - Física, 2 - Jurídica):
");

    tipo = scanner.nextInt();
    scanner.nextLine();

    if (tipo == 1) {
        for (PessoaFisica pf : repoFisica.obterTodos())
        {
            pf.exibir();
        }
    } else if (tipo == 2) {
        for (PessoaJuridica pj :
repoJuridica.obterTodos()) {
            pj.exibir();
        }
    }
    break;

case 6: // Salvar dados
    System.out.print("Prefixo do arquivo: ");
    String prefixo = scanner.nextLine();
    try {
        repoFisica.persistir(prefixo + ".fisica.bin");
    }
}

```

```

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

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

        } catch (IOException e) {
            System.err.println("Erro ao salvar os dados: "
+ e.getMessage());
        }
        break;

    case 7: // Recuperar dados
        System.out.print("Prefixo do arquivo: ");
        prefixo = scanner.nextLine();
        try {
            repoFisica.recuperar(prefixo + ".fisica.bin");
            repoJuridica.recuperar(prefixo +
".juridica.bin");

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

        } catch (IOException | ClassNotFoundException e) {
            System.err.println("Erro ao recuperar os dados:
" + e.getMessage());
        }
        break;

    case 0: // Sair
        System.out.println("Finalizando o sistema...");
        break;

    default:
        System.out.println("Opção inválida! Tente
novamente.");
    }
    } while (opcao != 0);

    scanner.close();
}
}

```

4. Resultados Obtidos

```
--- Menu ---
1. Incluir
2. Alterar
3. Excluir
4. Exibir pelo ID
5. Exibir todos
6. Salvar dados
7. Recuperar dados
0. Sair
Escolha uma opção: 5
Tipo (1 - Física, 2 - Jurídica): 2
ID: 1 , Nome: Avatar
CNPJ: 124135451542
ID: 2 , Nome: Kabib
CNPJ: 884777474
```

5. Análise e Conclusão

5.1. Elementos Estáticos e o Método `main`

Os elementos estáticos em Java são aqueles associados diretamente à classe e não a uma instância específica. Um elemento estático, seja ele um campo, método ou bloco, pertence à classe em si e é compartilhado por todas as instâncias dessa classe.

O método `main` adota o modificador `static` porque ele é o ponto de entrada de qualquer aplicação Java. Esse método é chamado pela JVM (Java Virtual Machine) antes que qualquer instância da classe seja criada. Sendo estático, o método `main` pode ser executado diretamente a partir do contexto da classe, sem a necessidade de instanciar um objeto. Esse comportamento é essencial, pois permite que o programa seja iniciado de forma independente de objetos.

A natureza estática do método `main` também simplifica o processo de inicialização do programa e reduz a sobrecarga de criação de objetos desnecessários durante o início da execução.

5.2. A Classe `Scanner`

A classe `Scanner`, pertencente ao pacote `java.util`, é utilizada para obter entrada de dados do usuário ou de outras fontes, como arquivos. Sua principal funcionalidade é facilitar a leitura de dados em diferentes formatos, como inteiros, textos, números decimais, entre outros.

No contexto de uma aplicação de cadastro em modo texto, a classe `Scanner` desempenha um papel crucial ao permitir a interação dinâmica entre o usuário e o programa. Por exemplo:

- **Leitura de opções do menu:** O usuário pode selecionar operações como incluir, alterar ou excluir registros.
- **Entrada de dados:** Permite que o usuário forneça informações específicas, como o nome, CPF ou ID de uma entidade.
- **Flexibilidade:** Oferece suporte à leitura de diferentes tipos de dados, como `int`, `double` e `String`, tornando-a versátil e prática.

Essa classe simplifica a interação com o terminal, permitindo que desenvolvedores criem interfaces baseadas em texto de forma eficiente e organizada.

5.3. Impacto do Uso de Classes de Repositório na Organização do Código

A introdução de classes de repositório, como `PessoaFisicaRepo` e `PessoaJuridicaRepo`, trouxe significativa melhoria na organização e estruturação do código. Esses repositórios encapsulam a lógica de gerenciamento de entidades (inserção, alteração, exclusão, recuperação, e listagem) em um único local, promovendo os princípios de **modularidade** e **responsabilidade única**.

Benefícios:

1. Clareza e Manutenção:

- As operações relacionadas a entidades específicas são centralizadas no repositório correspondente, tornando o código mais legível e fácil de manter.
- Alterações ou adições na lógica de persistência ou manipulação de dados podem ser feitas diretamente nas classes de repositório, sem impactar o restante do sistema.

2. Reutilização:

- Os métodos implementados nos repositórios podem ser reutilizados em diferentes partes do programa, evitando duplicação de código e promovendo o reaproveitamento.

3. Separação de Responsabilidades:

- A lógica de negócio (manipulação das entidades) é separada da interface do usuário (menu e interação via console). Isso facilita a evolução do sistema, por exemplo, ao trocar a interface de texto por uma interface gráfica.

4. Facilidade de Testes:

- As classes de repositório podem ser testadas isoladamente, garantindo que os métodos de gerenciamento funcionem conforme esperado, independentemente do restante do sistema.

5.5. Conclusão

O uso de classes de repositório não apenas organiza o código, mas também aumenta sua flexibilidade e robustez. Essa abordagem facilita a manutenção, escalabilidade e compreensão do sistema, especialmente em projetos mais complexos que envolvem múltiplas entidades e operações.