



Estácio
Paty do Alferes

Universidade estácio de sar

Polo = paty do alferes(RJ)

Curso = Desenvolvimento full stack

Disciplina Nível 1 = Iniciando caminho pelo java

Turma = 2023.3

Integrantes = João Victor Machado Santos

Link = <https://github.com/jvmstp/-Iniciando-o-caminho-pelo-Java>

RPG0014 - Iniciando o caminho pelo Java

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

1. Utilizar herança e polimorfismo na definição de entidades.

2. Utilizar persistência de objetos em arquivos binários.
3. Implementar uma interface cadastral em modo texto.
4. Utilizar o controle exceções da plataforma Java.
5. No final do projeto, o aluno terá implementado um sistema cadastral em Java, utilizando os recursos da programação orientada a objetos e a persistência em arquivos binários.

```
package cadastropoo;

import model.PessoaFisica;
import model.PessoaJuridica;
import model.PessoaFisicaRepo;
import model.PessoaJuridicaRepo;

import java.util.InputMismatchException;
import java.util.Scanner;

public class CadastroPOO {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        PessoaFisicaRepo repoFisica = new PessoaFisicaRepo();
        PessoaJuridicaRepo repoJuridica = new PessoaJuridicaRepo();

        int opcao = -1;

        while (opcao != 0) {
            try {
                System.out.println("Selecione uma opção:");
                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");
opcao = scanner.nextInt();
```

```
switch (opcao) {
    case 1: // Incluir
        incluir(scanner, repoFisica, repoJuridica);
        break;

    case 2: // Alterar
        alterar(scanner, repoFisica, repoJuridica);
        break;

    case 3: // Excluir
        excluir(scanner, repoFisica, repoJuridica);
        break;

    case 4: // Exibir pelo ID
        exibirPelold(scanner, repoFisica, repoJuridica);
        break;

    case 5: // Exibir todos
        exibirTodos(scanner, repoFisica, repoJuridica);
        break;

    case 6: // Salvar dados
        salvar(scanner, repoFisica, repoJuridica);
        break;
```

```

        case 7: // Recuperar dados
            recuperar(scanner, repoFisica, repoJuridica);

            break;

        case 0: // Sair
            System.out.println("Encerrando o programa.");

            break;

        default:
            System.out.println("Opção inválida.");

            break;

    }

    } catch (InputMismatchException e) {
        System.out.println("Entrada inválida. Por favor, insira um número.");
        scanner.nextLine(); // Consumir a entrada incorreta
    }

    }

    scanner.close();

    }

```

```

private static void incluir(Scanner scanner, PessoaFisicaRepo repoFisica,
    PessoaJuridicaRepo repoJuridica) {

    try {

        System.out.println("1. Pessoa Física | 2. Pessoa Jurídica");

        int tipo = scanner.nextInt();

        scanner.nextLine(); // Consumir a quebra de linha

        if (tipo == 1) {

            System.out.println("Informe o ID:");

            int id = scanner.nextInt();

            scanner.nextLine();

            System.out.println("Informe o nome:");

            String nome = scanner.nextLine();

            System.out.println("Informe o CPF:");

            String cpf = scanner.nextLine();

```

```

        System.out.println("Informe a idade:");
        int idade = scanner.nextInt();
        PessoaFisica pf = new PessoaFisica(id, nome, cpf, idade);
        repoFisica.inserir(pf);
    } else if (tipo == 2) {
        System.out.println("Informe o ID:");
        int id = scanner.nextInt();
        scanner.nextLine();
        System.out.println("Informe o nome:");
        String nome = scanner.nextLine();
        System.out.println("Informe o CNPJ:");
        String cnpj = scanner.nextLine();
        PessoaJuridica pj = new PessoaJuridica(id, nome, cnpj);
        repoJuridica.inserir(pj);
    }
} catch (InputMismatchException e) {
    System.out.println("Entrada inválida. Tente novamente.");
    scanner.nextLine(); // Limpar a entrada inválida
}
}

private static void alterar(Scanner scanner, PessoaFisicaRepo repoFisica,
    PessoaJuridicaRepo repoJuridica) {
    // Lógica para alterar (similar ao incluir)
}

private static void excluir(Scanner scanner, PessoaFisicaRepo repoFisica,
    PessoaJuridicaRepo repoJuridica) {
    // Lógica para excluir
}

private static void exibirPeloid(Scanner scanner, PessoaFisicaRepo repoFisica,
    PessoaJuridicaRepo repoJuridica) {
    // Lógica para exibir pelo ID
}

```

```

private static void exibirTodos(Scanner scanner, PessoaFisicaRepo repoFisica,
                                PessoaJuridicaRepo repoJuridica) {

    // Lógica para exibir todos

}

private static void salvar(Scanner scanner, PessoaFisicaRepo repoFisica,
                            PessoaJuridicaRepo repoJuridica) {

    System.out.println("Informe o prefixo do arquivo:");

    String prefixo = scanner.next();

    try {

        repoFisica.persistir(prefixo + ".fisica.bin");
        repoJuridica.persistir(prefixo + ".juridica.bin");
        System.out.println("Dados salvos com sucesso.");

    } catch (Exception e) {

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

    }

}

private static void recuperar(Scanner scanner, PessoaFisicaRepo repoFisica,
                               PessoaJuridicaRepo repoJuridica) {

    System.out.println("Informe o prefixo do arquivo:");

    String prefixo = scanner.next();

    try {

        repoFisica.recuperar(prefixo + ".fisica.bin");
        repoJuridica.recuperar(prefixo + ".juridica.bin");
        System.out.println("Dados recuperados com sucesso.");

    } catch (Exception e) {

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

    }

}

```

Resultado da execução

Selecione uma opção:

1. Incluir
2. Alterar
3. Excluir
4. Exibir pelo ID
5. Exibir todos
6. Salvar dados
7. Recuperar dados
0. Sair

1

1. Pessoa Física | 2. Pessoa Jurídica

1

Informe o ID:

09

Informe o nome:

joao

Informe o CPF:

12345678912

Informe a idade:

29

Selecione uma opção:

1. Incluir
2. Alterar
3. Excluir
4. Exibir pelo ID
5. Exibir todos
6. Salvar dados
7. Recuperar dados
0. Sair

Análise e Conclusão:

- Elementos estáticos pertencem à classe, e não a instâncias dela. O método(`main`) precisa ser estático para ser acessado diretamente pela JVM, sem a necessidade de criar uma instância da classe.
- A classe `Scanner` é usada para ler entradas de dados do teclado, permitindo interatividade no programa ao receber dados como inteiros, (strings) e outros tipos de entrada.
- O uso de classes de repositório melhorou a organização do código, separando a lógica de persistência de dados da lógica de interface com o usuário, promovendo a modularidade e facilitando a manutenção e expansão do sistema.