



Desenvolvimento Full Stack

Felipe Marques de Almeida - 202208291929

Polo Rodovia Br 407 - Juazeiro - Ba

RPG 0014 - Iniciando o caminho pelo Java – Turma 22.3 – 3º Semestre

GitHub: [Kifflom2108/Missao-Pratica-Nivel-1-Mundo-3 \(github.com\)](https://github.com/Kifflom2108/Missao-Pratica-Nivel-1-Mundo-3)

Objetivo da Prática

Utilizar herança e polimorfismo na definição de entidades.

Utilizar persistência de objetos em arquivos binários.

Implementar uma interface cadastral em modo texto.

Utilizar o controle de exceções da plataforma Java.

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.

Códigos:

```
5 package com.mycompany.cadastropoo.model;
6
7 import java.io.IOException;
8 import java.util.InputMismatchException;
9 import java.util.List;
10 import java.util.Scanner;
11
12 public class Main {
13
14     public static void main(String[] args) {
15         Scanner scanner = new Scanner(System.in);
16         PessoaFisicaRepo repoCpf = new PessoaFisicaRepo();
17         PessoaJuridicaRepo repoCnpj = new PessoaJuridicaRepo();
18         int opcao = getMenuOpcao(scanner, repoCpf, repoCnpj);
19         while (opcao != 0);
20     }
21
22     private static void separador() {
23         System.out.println("-----");
24     }
25
26     private static int getMenuOpcao(Scanner scanner, PessoaFisicaRepo repoCpf, PessoaJuridicaRepo repoCnpj) {
27         separador();
28         System.out.println("1 - Incluir pessoa");
29         System.out.println("2 - Alterar pessoa");
30         System.out.println("3 - Excluir pessoa");
31         System.out.println("4 - Exibir pelo id");
32         System.out.println("5 - Exibir todos");
33         System.out.println("6 - Salvar dados");
34         System.out.println("7 - Carregar Dados");
35         separador();
36         int opcao = 0;
37         try {
38             opcao = scanner.nextInt();
39         } catch (InputMismatchException e) {
40             getLinhaFormatada(scanner, titulo: "Ocorreu um erro, tente novamente.");
41             getMenuOpcao(scanner, repoCpf, repoCnpj);
42         }
43
44         switch (opcao) {
45             case 1:
46                 incluirPessoa(scanner, repoCpf, repoCnpj);
47                 break;
48             case 2:
49                 alterarPessoa(scanner, repoCpf, repoCnpj);
50                 break;
51             case 3:
52                 excluirPessoa(scanner, repoCpf, repoCnpj);
53                 break;
54             case 4:
55                 exibirPeloid(scanner, repoCpf, repoCnpj);
56                 break;
57             case 5:
```

```

58     exibirPessoa(scanner, repoCpf, repoCnpj);
59     break;
60 case 6:
61     salvarDados(scanner, repoCpf, repoCnpj);
62     break;
63 case 7:
64     carregarDados(scanner, repoCpf, repoCnpj);
65     break;
66 case 0:
67     System.out.println("Saindo do programa.");
68     break;
69 default:
70     System.out.println("Opção inválida. Tente novamente.");
71     break;
72 }
73 return getMenuOpcao(scanner, repoCpf, repoCnpj);
74 }
75 }
76
77 private static void incluirPessoa(Scanner scanner, PessoaFisicaRepo repoCpf, PessoaJuridicaRepo repoCnpj) {
78     int tipo = getIntFormatado(scanner, titulo: "Escolha o tipo de pessoa (1 para Física, 2 para Jurídica);");
79     scanner.nextLine();
80
81     if (tipo == 1) {
82         String nome = getLinhaFormatada(scanner, titulo: "Digite o nome da pessoa física: ");
83         String cpf = getLinhaFormatada(scanner, titulo: "Digite o CPF da Pessoa Física:");
84         int idade = getIntFormatado(scanner, titulo: "Digite a idade da Pessoa Física:");
85         PessoaFisica pessoaFisica = new PessoaFisica(repoCpf.obterTodos().size() + 1, nome, cpf, idade);
86         repoCpf.inserir(pessoaFisica);
87         System.out.println("Pessoa física adicionada com sucesso, o id é " + repoCpf.obterTodos().size());
88     } else if (tipo == 2) {
89         String nome = getLinhaFormatada(scanner, titulo: "Digite o nome da Pessoa Jurídica:");
90         String cnpj = getLinhaFormatada(scanner, titulo: "Digite o CNPJ da Pessoa Jurídica:");
91         PessoaJuridica pessoaJuridica = new PessoaJuridica(repoCnpj.obterTodos().size() + 1, nome, cnpj);
92         repoCnpj.inserir(pessoaJuridica);
93     } else {
94         System.out.println("Opção inválida.");
95     }
96 }
97
98 private static void alterarPessoa(Scanner scanner, PessoaFisicaRepo repoCpf, PessoaJuridicaRepo repoCnpj) {
99     int tipo = getIntFormatado(scanner, titulo: "Escolha o tipo de pessoa (1 para Física, 2 para Jurídica);");
100    if (tipo == 1) {
101        int idPessoaFisicaAlterar = getIntFormatado(scanner, titulo: "Digite o ID da Pessoa Física que deseja alterar: ");
102        PessoaFisica pessoaFisicaExistente = repoCpf.obter(id: idPessoaFisicaAlterar);
103        if (pessoaFisicaExistente != null) {
104            System.out.println("Dados da Pessoa Física a ser alterada:");
105            pessoaFisicaExistente.exibir();
106            String novoNomePF = getLinhaFormatada(scanner, titulo: "Digite o novo nome: ");
107            String novoCpf = getLinhaFormatada(scanner, titulo: "Digite o novo CPF: ");
108            int novaldade = getIntFormatado(scanner, titulo: "Digite a nova idade: ");
109            PessoaFisica pessoaFisicaAtualizada = new PessoaFisica(id: idPessoaFisicaAlterar, nome: novoNomePF, cpf: novoCpf, idade: novaldade);
110            repoCpf.alterar(pessoaFisicaExistente, novaPessoaFisica: pessoaFisicaAtualizada);
111            System.out.println("Pessoa Física alterada com sucesso! \n");
112        } else {
113            System.out.println("Pessoa Física com o ID " + idPessoaFisicaAlterar + " não encontrado. \n");
114        }
115    } else if (tipo == 2) {
116        int pessoaJuridicaAlterar = getIntFormatado(scanner, titulo: "Digite o ID da Pessoa Jurídica que deseja alterar: ");
117        PessoaJuridica pessoaJuridicaExistente = repoCnpj.obter(id: pessoaJuridicaAlterar);
118        if (pessoaJuridicaExistente != null) {
119            System.out.println("Dados da Pessoa Jurídica a ser alterada");
120            pessoaJuridicaExistente.exibir();
121            String novoNomePJ = getLinhaFormatada(scanner, titulo: "Digite o novo nome da Empresa: ");
122            String novoCnpj = getLinhaFormatada(scanner, titulo: "Digite o novo CNPJ da Empresa: ");
123            PessoaJuridica pessoaJuridicaAtualizada = new PessoaJuridica(id: pessoaJuridicaAlterar, nome: novoNomePJ, cnpj: novoCnpj);
124            repoCnpj.alterar(pessoaJuridicaExistente, novaPessoaJuridica: pessoaJuridicaAtualizada);
125        } else {
126            System.out.println("Pessoa Jurídica com o ID " + pessoaJuridicaAlterar + " não encontrado. \n");
127        }
128    } else {
129        System.out.println("Opção inválida. \n");
130    }
131 }
132
133 private static void excluirPessoa(Scanner scanner, PessoaFisicaRepo repoCpf, PessoaJuridicaRepo repoCnpj) {
134
135     int excluir = getIntFormatado(scanner, titulo: "Escolha (1 - Pessoa Física, 2 - Pessoa Jurídica) ");
136     if (excluir == 1) {
137         int pessoaFisicaExcluir = getIntFormatado(scanner, titulo: "Digite o ID da Pessoa Física que deseja excluir: ");
138         PessoaFisica pessoaFisicaExistente = repoCpf.obter(id: pessoaFisicaExcluir);
139         if (pessoaFisicaExistente != null) {
140             System.out.println("Dados da Pessoa Física a ser excluída");
141             pessoaFisicaExistente.exibir();
142             String confirmacaoPessoaFisica = getLinhaFormatada(scanner, titulo: "Tem certeza de que deseja excluir esta Pessoa Física? (S/N): ");
143             if (confirmacaoPessoaFisica.equalsIgnoreCase("S")) {
144                 repoCpf.excluir(id: pessoaFisicaExcluir);
145                 System.out.println("Pessoa Física excluída com sucesso! \n");
146             } else {
147                 System.out.println("Exclusão cancelada. \n");
148             }
149         } else {
150             System.out.println("Pessoa Física com o ID " + pessoaFisicaExcluir + " não encontrado. \n");
151         }
152     } else if (excluir == 2) {
153         int pessoaJuridicaExcluir = getIntFormatado(scanner, titulo: "Digite o ID da Pessoa Jurídica que deseja excluir: ");
154         PessoaJuridica pessoaJuridicaExistente = repoCnpj.obter(id: pessoaJuridicaExcluir);
155         if (pessoaJuridicaExistente != null) {
156             System.out.println("Dados da Pessoa Jurídica a ser excluída");
157             pessoaJuridicaExistente.exibir();
158             String confirmacaoPessoaJuridica = getLinhaFormatada(scanner, titulo: "Tem certeza de que deseja excluir esta Pessoa Jurídica? (S/N): ");
159             if (confirmacaoPessoaJuridica.equalsIgnoreCase("S")) {
160                 repoCnpj.excluir(id: pessoaJuridicaExcluir);
161                 System.out.println("Pessoa Jurídica excluída com sucesso! \n");
162             } else {
163                 System.out.println("Exclusão cancelada. \n");
164             }
165         }
166     }
167 }

```

```

166     } else {
167         System.out.println("Pessoa Juridica com o ID " + pessoaJuridicaExcluir + " não encontrado. \n");
168     }
169     } else {
170         System.out.println("Opção inválida. \n");
171     }
172 }
173
174 private static void exibirPeloid(Scanner scanner, PessoaFisicaRepo repoCpf, PessoaJuridicaRepo repoCnpj) {
175     int obter = getIntFormatado(scanner, titulo: "Escolha (1 - Pessoa Física, 2 - Pessoa Jurídica)");
176     if (obter == 1) {
177         int obterPessoaFisica = getIntFormatado(scanner, titulo: "Digite o ID da Pessoa Física que deseja obter: ");
178         PessoaFisica pessoaFisicaExistente = repoCpf.obter(id: obterPessoaFisica);
179         if (pessoaFisicaExistente != null) {
180             System.out.println("Dados da Pessoa Física obtida");
181             pessoaFisicaExistente.exibir();
182         } else {
183             System.out.println("Pessoa Física com o ID " + obterPessoaFisica + " não encontrado. \n");
184         }
185     } else if (obter == 2) {
186         int obterPessoaJuridica = getIntFormatado(scanner, titulo: "Digite o ID da Pessoa Jurídica que deseja obter: ");
187         PessoaJuridica pessoaJuridicaExiste = repoCnpj.obter(id: obterPessoaJuridica);
188         if (pessoaJuridicaExiste != null) {
189             System.out.println("Dados da Pessoa Jurídica obtida");
190             pessoaJuridicaExiste.exibir();
191         } else {
192             System.out.println("Pessoa Jurídica com o ID " + obterPessoaJuridica + " não encontrado. \n");
193         }
194     } else {
195         System.out.println("Opção inválida. \n");
196     }
197 }
198
199 private static void exibirPessoa(Scanner scanner, PessoaFisicaRepo repoCpf, PessoaJuridicaRepo repoCnpj) {
200     int tipo = getIntFormatado(scanner, titulo: "Escolha o tipo de pessoa (1 para Física, 2 para Jurídica)");
201     scanner.nextLine();
202     separador();
203
204     if (tipo == 1) {
205         List<PessoaFisica> obterTodosCpf = repoCpf.obterTodos();
206         for (PessoaFisica pessoaFisica : obterTodosCpf) {
207             pessoaFisica.exibir();
208             separador();
209         }
210     } else if (tipo == 2) {
211         List<PessoaJuridica> obterTodosCnpj = repoCnpj.obterTodos();
212         for (PessoaJuridica pessoaJuridica : obterTodosCnpj) {
213             pessoaJuridica.exibir();
214         }
215     } else {
216         System.out.println("Opção inválida.");
217     }
218 }
219 private static void salvarDados(Scanner scanner, PessoaFisicaRepo repoCpf, PessoaJuridicaRepo repoCnpj){

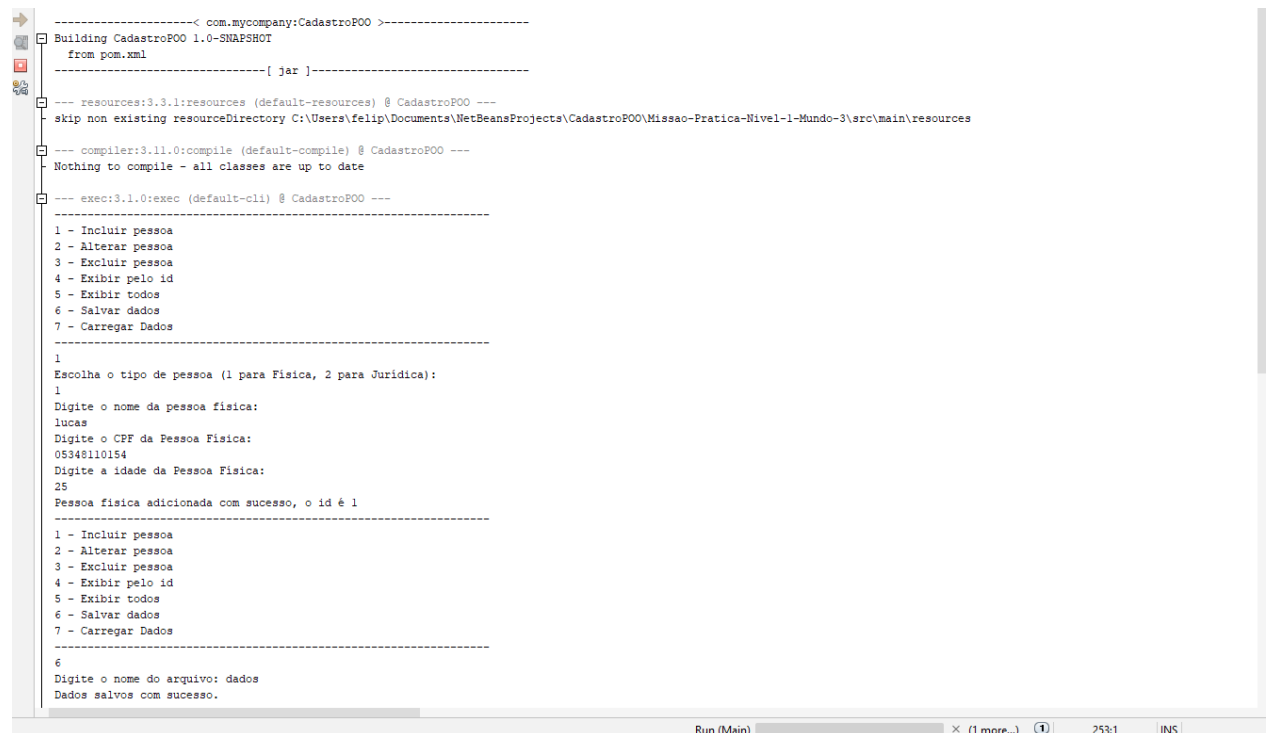
```

```

220     System.out.print("Digite o nome do arquivo: ");
221     String prefixoSalvar = scanner.next();
222     try {
223         repoCpf.persistir(prefixoSalvar + ".fisica.bin");
224         repoCnpj.persistir(prefixoSalvar + ".juridica.bin");
225         System.out.println("Dados salvos com sucesso.");
226     } catch (IOException e) {
227         System.out.println("Erro ao salvar dados: " + e.getMessage());
228     }
229 }
230
231 private static void carregarDados(Scanner scanner, PessoaFisicaRepo repoCpf, PessoaJuridicaRepo repoCnpj){
232     System.out.print("Digite o nome do arquivo: ");
233     String prefixoRecuperar = scanner.next();
234     try {
235         repoCpf.recuperar(prefixoRecuperar + ".fisica.bin");
236         repoCnpj.recuperar(prefixoRecuperar + ".juridica.bin");
237         System.out.println("Dados recuperados com sucesso. \n");
238     } catch (IOException | ClassNotFoundException e) {
239         System.out.println("Erro ao recuperar dados: " + e.getMessage());
240     }
241 }
242
243 private static String getLinhaFormatada(Scanner scanner, String titulo) {
244     System.out.println(":" + titulo);
245     return scanner.next();
246 }
247
248 private static int getIntFormatado(Scanner scanner, String titulo) {
249     System.out.println(":" + titulo);
250     return scanner.nextInt();
251 }
252
253

```

Resultado:



```
-----< com.mycompany:CadastroPOO >-----
Building CadastroPOO 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ CadastroPOO ---
skip non existing resourceDirectory C:\Users\felip\Documents\NetBeansProjects\CadastroPOO\Missao-Pratica-Nivel-1-Mundo-3\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ CadastroPOO ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ CadastroPOO ---
-----
1 - Incluir pessoa
2 - Alterar pessoa
3 - Excluir pessoa
4 - Exibir pelo id
5 - Exibir todos
6 - Salvar dados
7 - Carregar Dados
-----
1
Escolha o tipo de pessoa (1 para Fisica, 2 para Juridica):
1
Digite o nome da pessoa fisica:
lucas
Digite o CPF da Pessoa Fisica:
05348110154
Digite a idade da Pessoa Fisica:
25
Pessoa fisica adicionada com sucesso, o id é 1
-----
1 - Incluir pessoa
2 - Alterar pessoa
3 - Excluir pessoa
4 - Exibir pelo id
5 - Exibir todos
6 - Salvar dados
7 - Carregar Dados
-----
6
Digite o nome do arquivo: dados
Dados salvos com sucesso.
```

Análise e Conclusão:

Elementos estáticos em Java, como o método main, pertencem à classe e não a instâncias específicas. O método main é declarado como estático para ser o ponto de entrada do programa, podendo ser chamado sem a necessidade de criar uma instância da classe.

A classe Scanner é usada para receber dados de entrada do usuário, facilitando a interação com o programa ao ler dados do teclado ou de outras fontes de entrada.

O uso de classes de repositório, como PessoaFisicaRepo e PessoaJuridicaRepo, melhora a organização do código ao encapsular a lógica de persistência, facilitando a manutenção e o desacoplamento das operações de armazenamento e recuperação de dados. Isso resulta em código mais modular e legível.