



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

**André Luis Gonçalves Carvalhal - Matrícula: 20220318540**

**Campus Barra World**

**Iniciando o Caminho pelo Java – 9001 – 2023.3**

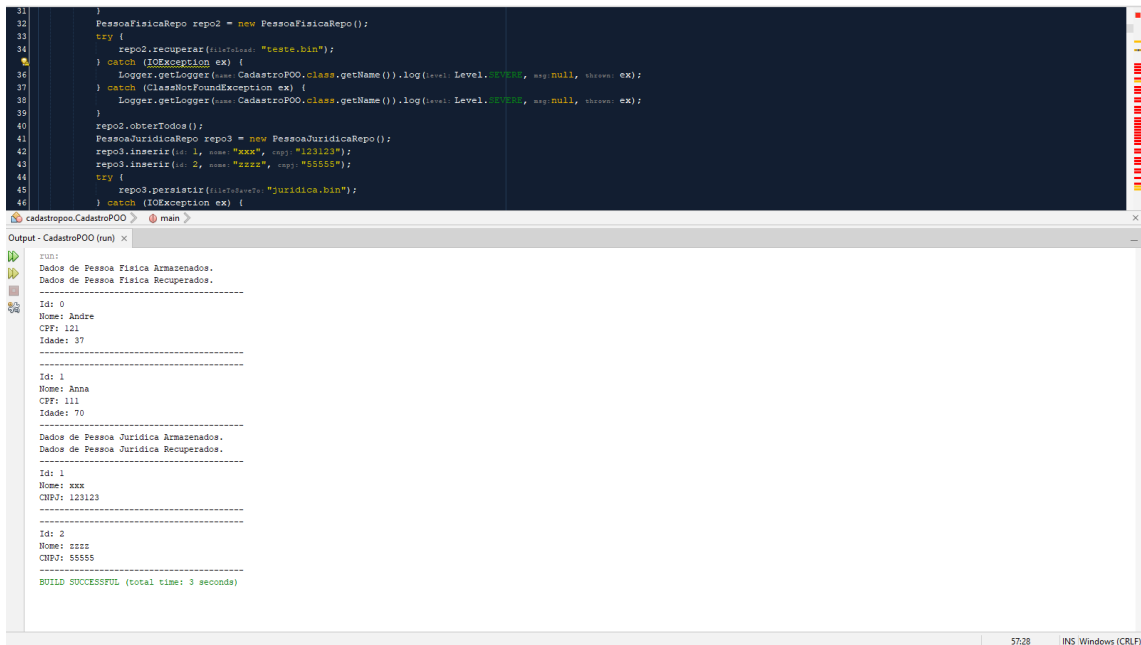
### **Objetivo da Prática**

- 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 de 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.**

### **Link GitHub**

**<https://github.com/ANDREC1986/RPG0014-202203185403>**

## 1º Procedimento | Criação das Entidades e Sistema de Persistência



```
31 }
32 PessoaFisicaRepo repo2 = new PessoaFisicaRepo();
33 try {
34     repo2.recuperar("teste.bin");
35 } catch (IOException ex) {
36     Logger.getLogger(CadastroPOO.class.getName()).log(Level.SEVERE, null, ex);
37 } catch (ClassNotFoundException ex) {
38     Logger.getLogger(CadastroPOO.class.getName()).log(Level.SEVERE, null, ex);
39 }
40 repo3.obterTodos();
41 PessoaJuridicaRepo repo3 = new PessoaJuridicaRepo();
42 repo3.inserir(id: 1, nome: "xxx", cnpj: "123123");
43 repo3.inserir(id: 2, nome: "zzzz", cnpj: "55555");
44 try {
45     repo3.persistir("juridica.bin");
46 } catch (IOException ex) {
47     //
48 }
49 }
50 }
51 }
52 }
53 }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
```

Output - CadastroPOO (run) x

```
run:
Dados de Pessoa Fisica Armazenados.
Dados de Pessoa Fisica Recuperados.
-----
Id: 0
Nome: Andre
CPF: 121
Idade: 37
-----
Id: 1
Nome: Anna
CPF: 111
Idade: 70
-----
Dados de Pessoa Juridica Armazenados.
Dados de Pessoa Juridica Recuperados.
-----
Id: 1
Nome: xxx
CNPJ: 123123
-----
Id: 2
Nome: zzzz
CNPJ: 55555
-----
BUILD SUCCESSFUL (total time: 3 seconds)
```

### Conclusão:

#### a) Quais as vantagens e desvantagens do uso de herança?

As vantagens estão na reutilização do código, podendo desenvolver em uma superclasse e reutilizá-lo nas subclasses, você obtém extensibilidade ao poder estender a classe pai, criando novas subclasses, o código fica organizado de forma mais de compreender, pois estabelece uma hierarquia no código.

As desvantagens são, o forte acoplamento entre a superclasse e as subclasses, tornando o sistema mais frágil, já que a alteração na superclasse afetará todos as subclasses relacionadas. As classes podem se tornar muito complexas, atrapalhando a reutilização do código.

#### b) Por que a interface Serializable é necessária ao efetuar persistência em arquivos binários?

A interface “Serializable” permite ao mecanismo de serialização do java saber que determinada classe pode ser convertida em um forma binária, para posteriormente pode ser transmitida ou persistida em arquivo.

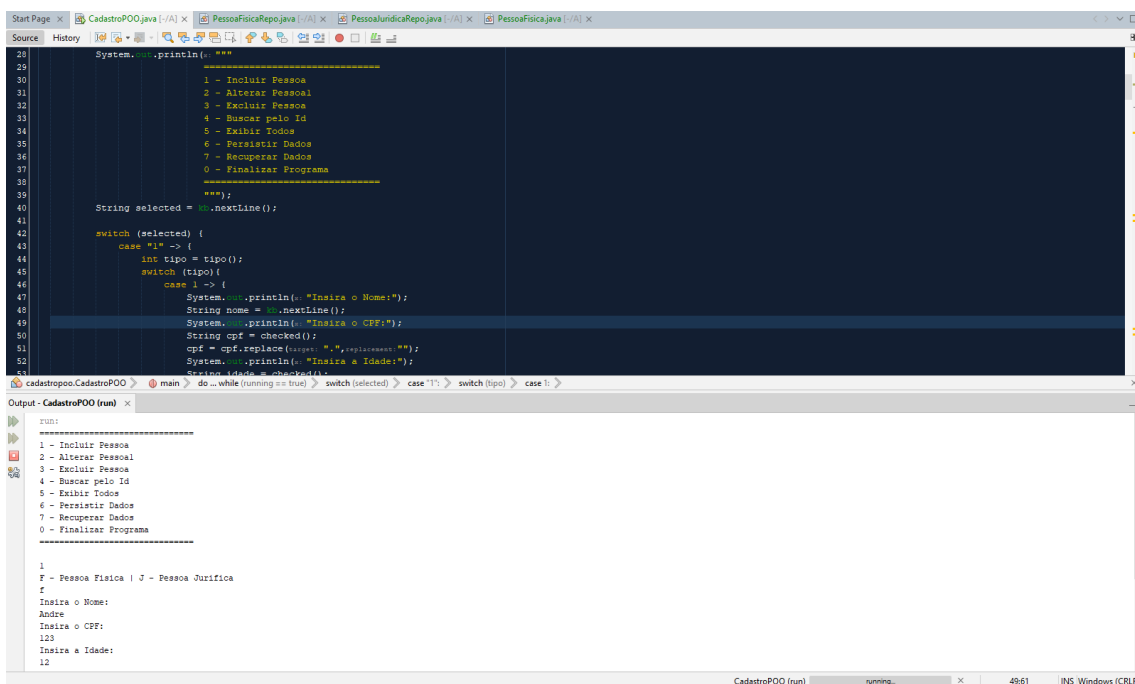
**c) Como o paradigma funcional é utilizado pela API stream no Java?**

A API Stream no Java utilizando o paradigma funcional, assim como no Javascript que tivemos no mundo 2, ela permite a redução, ordenação, agrupamento, filtragem e mapeamento de dados em coleções de forma concisa.

**d) Quando trabalhamos com Java, qual padrão de desenvolvimento é adotado na persistência de dados em arquivos?**

A Serialização, método que permite que classes sejam transformadas em bytes e operadas utilizando os ObjectOutputStream e ObjectInputStream para gravar e carregar os dados respectivamente.

## 2º Procedimento | Criação do Cadastro em Modo Texto



```
28 System.out.println("=====");
29
30 1 - Incluir Pessoa
31 2 - Alterar Pessoa
32 3 - Excluir Pessoa
33 4 - Buscar pelo Id
34 5 - Exibir Todos
35 6 - Persistir Dados
36 7 - Recuperar Dados
37 0 - Finalizar Programa
38
39 =====
40
41 String selected = scanner.nextLine();
42
43 switch (selected) {
44     case "1" -> {
45         int tipo = tipo();
46         switch (tipo) {
47             case 1 -> {
48                 System.out.println("Insira o Nome:");
49                 String nome = scanner.nextLine();
50                 System.out.println("Insira o CPF:");
51                 String cpf = checked();
52                 cpf = cpf.replace(".", "");
53                 System.out.println("Insira a Idade:");
54                 String idade = checked();
55             }
56         }
57     }
58 }
59
60 main
61 do...while (running == true)
62 switch (selected)
63 case "1":
64 switch (tipo)
65 case 1:
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
Output - CadastroPOO (run)
RUN:
=====
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====

1
? - Pessoa Fisica | J - Pessoa Juridica
?
Insira o Nome:
Andre
Insira o CPF:
123
Insira a Idade:
12
```

**Conclusão:**

**a) O que são elementos estáticos e qual o motivo para o método main adotar esse modificador?**

Métodos estáticos são os métodos que não dependem do estado de um objeto, podendo ser invocados diretamente da classe, sem a necessidade de

se instanciar um objeto. O Main utiliza pois o mesmo será acessado direto pela JVM, não tendo a necessidade de ser instanciado como um objeto.

**b) Para que serve a classe Scanner?**

A classe Scanner é uma classe usada para ler dados, podendo ler dados de diversas fontes, como teclado, arquivos, streams, strings.

**c) Como o uso de classes de repositório impactou na organização do código?**

Permitiu a melhor leitura do código, separando o modelo do controle, permitindo a implementação mais rápida do método main.

**Conclusão:**

Foi um trabalho interessante, não tomou muito tempo, acredito que a separação poderia ter sido mais elaborado, separando os controladores da pasta model e colocados em uma pasta controller, para trazer maior clareza para aqueles que ainda não estão acostumados com o modelo MVC.