



Campus: Santa Cruz da Serra - Duque de Caxias/RJ  
Curso: Desenvolvimento full stack  
Disciplina: RPG0014 - Iniciando o caminho pelo Java  
Número da Turma: 9001  
Semestre letivo: Mundo 3  
Integrantes: Felipe Freaza Fidalgo

## Iniciando o caminho pelo Java (CadastroPOO)

```
Output - CadastroPOO (run)

run:
Dados de Pessoas Físicas Armazenados.
Dados de Pessoas Físicas Recuperadas:
ID: 327
Nome: Rodrigo Oliveira
CPF: 63338168330
Idade: 42

ID: 582
Nome: Marina Santos
CPF: 74089572355
Idade: 39

Dados de Pessoas Jurídicas Armazenados.
Dados de Pessoas Jurídicas Recuperadas:
ID: 192
Nome: InovaSoft Tecnologia
CNPJ: 67551068000130

ID: 739
Nome: Soluções NovaTech
CNPJ: 17601791000108

BUILD SUCCESSFUL (total time: 9 seconds)
```

```
Saida - CadastroPOO (run)

run:

MENU PRINCIPAL

Selecione uma opção:
1 - Incluir
2 - Alterar
3 - Excluir
4 - Exibir pelo ID
5 - Exibir todos
6 - Salvar dados
7 - Recuperar dados
0 - Finalizar

1
Selecione o tipo (F - Física, J - Jurídica):
F
Digite o ID:
689
Digite o nome:
```

## Análise e Conclusão:

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

## 1.Quais as vantagens e desvantagens do uso de herança?

A herança é um conceito fundamental na programação orientada a objetos (POO) e pode trazer tanto vantagens quanto desvantagens.

Algumas das vantagens são a reutilização de código, abstração e generalização, facilidade de manutenção e polimorfismo.

Algumas das desvantagens são o acoplamento, hierarquia complexa, fragilidade e aumento do acoplamento temporal.

Em outras palavras, a herança pode ser uma ferramenta muito útil na programação orientada a objetos, mas é necessário um bom planejamento e consideração dos possíveis impactos. É importante equilibrar os benefícios da reutilização de código com as possíveis desvantagens da complexidade e fragilidade do código.

## 2.Por que a interface Serializable é necessária ao efetuar persistência em arquivos binários?

A interface Serializable é necessária para persistência em arquivos binários porque permite que um objeto seja convertido em bytes e, assim, seja salvo em um arquivo. Essa conversão preserva o estado do objeto. Ao ler o arquivo, os bytes são interpretados e o objeto é reconstruído, mantendo seus atributos e valores originais. A interface Serializable facilita a gravação e a recuperação de objetos em arquivos binários, possibilitando sua persistência.

### **3.Como o paradigma funcional é utilizado pela API stream no Java?**

A API Stream do Java utiliza o paradigma funcional para manipular coleções de dados de forma concisa e eficiente. Ela utiliza pipelines de operações, compostas por operações intermediárias e terminais, para processar os elementos da coleção. As operações intermediárias transformam e filtram os elementos, enquanto as operações terminais produzem um resultado final. Expressões lambda e interfaces funcionais são usadas para definir as operações a serem executadas nos elementos da coleção. A API Stream permite realizar operações complexas em coleções de maneira mais legível e eficiente.

### **4.Quando trabalhamos com Java, qual padrão de desenvolvimento é adotado na persistência de dados em arquivos?**

No desenvolvimento em Java, é comum adotar o padrão DAO (Data Access Object) para a persistência de dados em arquivos. Esse padrão separa a lógica de acesso aos dados do restante do código, utilizando interfaces ou classes abstratas para definir os métodos de acesso aos dados. As implementações concretas lidam com a manipulação dos dados em arquivos. O padrão DAO promove a modularidade, a separação de preocupações e a flexibilidade no código, permitindo a troca de implementações de persistência sem afetar o sistema.

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

### **1.O que são elementos estáticos e qual o motivo para o método main adotar esse modificador?**

Elementos estáticos são membros da classe que são compartilhados por todas as instâncias. Isso inclui variáveis estáticas e métodos estáticos. O método "main" é adotado como estático porque é o ponto de entrada do programa e precisa ser acessível diretamente pela classe, sem a necessidade de criar uma instância. Além disso, ele é executado no contexto estático da classe. Portanto, o modificador "static"

permite que o método "main" seja chamado sem a criação de um objeto da classe e seja o ponto de partida do programa.

## **2. Para que serve a classe Scanner?**

A classe Scanner em Java é usada para obter entrada do usuário, tanto a partir do teclado como de arquivos. Ela fornece métodos para ler diferentes tipos de dados, como inteiros, números de ponto flutuante e strings. A classe Scanner simplifica a interação com o usuário, permitindo que os programas recebam e processem informações inseridas durante a execução.

## **3. Como o uso de classes de repositório impactou na organização do código?**

O uso de classes de repositório impacta positivamente na organização do código ao separar a camada de persistência da lógica de negócio. Isso resulta em um código mais limpo, coeso e reutilizável, permitindo concentrar as operações de acesso e manipulação de dados em um único lugar. Essa abordagem facilita a troca de mecanismos de armazenamento e auxilia na implementação de boas práticas de programação, como o princípio da responsabilidade única. É importante manter uma estrutura coerente e consistente ao utilizar as classes de repositório.

## **Conclusão:**

A abordagem do projeto em Java utilizando o padrão Ant e o NetBeans é adequada para criação de um sistema de cadastro com persistência. A estruturação das entidades e gerenciadores é bem definida, facilitando a manutenção. No entanto, o uso extensivo de herança para diferenciar PessoaFisica e PessoaJuridica pode complicar a evolução do sistema. Além disso, a persistência em arquivos binários pode ser limitada em cenários mais complexos devido à falta de flexibilidade e escalabilidade. A ausência de validações de entrada e tratamento mais robusto de exceções pode impactar a robustez do sistema.

