



Missão Prática | Nível 1 | Mundo 3

João Rainier de Castro Carvalho (202208942661)

Estácio Distrito Federal (Polo Asa Sul)/DF – Desenvolvimento Full Stack – 2022.3 – Mundo 3

Link GitHub: [Faculdade-Estacio-3-semester/Nível 1 - Iniciando o Caminho Pelo Java/CadastroPOO at main · joaorainier/Faculdade-Estacio-3-semester \(github.com\)](https://github.com/joaorainier/Faculdade-Estacio-3-semester/tree/main/Java/CadastroPOO)

Objetivo da Prática:

RPG0014 - Iniciando o caminho pelo Java

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

Objetivos 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.

Materiais necessários para a prática:

JDK e IDE NetBeans.

Equipamentos: Computador com JDK e NetBeans instalados.

1 - Quais as vantagens e desvantagens do uso de herança?

Das Vantagens: reutilização de código, a herança permite que você reutilize o código de uma classe existente em uma nova classe. Polimorfismo, Classes derivadas podem ser tratadas como instâncias da classe base, o que facilita a criação de código mais genérico e flexível. Estrutura hierárquica, a herança ajuda a organizar as classes em uma hierarquia, facilitando a compreensão e manutenção do código.

Das Desvantagens: acoplamento, criar um acoplamento entre a classe base e suas subclasses, o que pode tornar o código mais difícil de entender e manter. Herança múltipla, em linguagens que suportam herança múltipla, pode haver problemas de ambiguidade e complexidade. Mudanças na classe base podem afetar as subclasses, e vice-versa, tornando o sistema mais frágil.

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

Se faz necessária para indicar que uma classe pode ser serializada, ou seja, pode ser convertida em uma sequência de bytes para ser gravada em arquivos ou transmitida pela rede. Ao efetuar persistência em arquivos binários, a serialização é usada para converter objetos em bytes para que possam ser armazenados e posteriormente restaurados. A interface Serializable é uma marcação que não contém métodos a serem implementados, mas sinaliza ao ambiente de execução Java que a classe pode ser serializada.

3 - Como o paradigma funcional é utilizado pela API stream no Java?

Oferecem operações de processamento de dados de forma mais declarativa e expressiva. Alguns elementos do paradigma funcional na API Stream incluem:

Operações de alto nível: Operações como map, filter, e reduce são oferecidas, permitindo um código mais conciso e legível.

Imutabilidade: Muitas operações da API Stream retornam novos Streams, preservando a imutabilidade dos dados originais.

Expressões lambda: As expressões lambda são frequentemente usadas para definir comportamentos a serem aplicados nos elementos do Stream.

Lazy evaluation: As operações em um Stream são avaliadas sob demanda, o que pode levar a um melhor desempenho em certos casos.

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

Em Java, um padrão comum para persistência de dados em arquivos é o uso de serialização, mencionado anteriormente. Além disso, o Java fornece bibliotecas como o ObjectOutputStream e ObjectInputStream que podem ser utilizadas para gravar e ler objetos em arquivos de forma eficiente. Outras abordagens incluem o uso de APIs de acesso a banco de dados embutidos (como o H2 Database), ou o uso de bibliotecas de persistência como Hibernate para persistência em bancos de dados relacionais. A escolha depende dos requisitos específicos do projeto e das necessidades de persistência de dados.

5 - O que são elementos estáticos e qual o motivo para o método main adotar esse modificador?

É uma classe que pertence à própria classe em vez de depender de instâncias específicas da classe. Existem dois tipos principais de elementos estáticos: métodos estáticos e variáveis estáticas. O método main em Java é estático para permitir que a JVM o chame sem precisar instanciar a classe que o contém.

6 - Para que serve a classe Scanner?

A classe `Scanner` em Java, localizada no pacote `java.util`, é utilizada para obter entrada do usuário a partir do teclado ou para realizar leituras de dados de outras fontes, como arquivos. Ela oferece métodos para analisar e converter diferentes tipos de dados, facilitando a interação com o usuário ou a leitura de dados de entrada.

7 - Como o uso de classes de repositório impactou na organização do código?

Nesta prática, ficou mais fácil de encontrar os objetos para alterar suas funcionalidades.

