



Universidade Estácio de Sá
Paulo amancio de souza

Desenvolvimento Full Stack

Iniciando o Caminho Pelo Java
Mundo 3
Parte 2

Palhoça
2024

Acesso Github:<https://github.com/willsouzaa/Cadastro-de-clientes-java.git>

Título da Prática

Cadastro de Pessoas Físicas e Jurídicas em Modo Texto com Persistência de Dados

Objetivo da Prática

Implementar um sistema de cadastro para pessoas físicas e jurídicas, permitindo inclusão, alteração, exclusão, exibição e persistência de dados usando arquivos binários.

Códigos Implementados

A prática incluiu a implementação de classes para representar pessoas físicas e jurídicas, repositórios para gerenciar essas entidades, e um programa principal que oferece uma interface de texto para interação com o usuário.

Resultados da Execução dos Códigos

- **Inclusão:** Adição de novas pessoas físicas e jurídicas foi realizada com sucesso.
- **Alteração:** Modificação dos dados existentes foi feita corretamente.
- **Exclusão:** Remoção de registros funcionou conforme esperado.
- **Exibição:** Dados de pessoas individuais e listas completas foram exibidos corretamente.
- **Persistência:** Dados foram salvos e recuperados de arquivos binários sem problemas.

Análise e Conclusão

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

Elementos estáticos pertencem à classe em vez de instâncias individuais. O método `main` é estático porque ele precisa ser chamado pela JVM sem criar uma instância da classe, permitindo que o programa seja executado como ponto de entrada.

Para que serve a classe Scanner?

A classe `Scanner` é utilizada para ler a entrada do usuário de várias fontes, como o teclado. No programa, `Scanner` captura dados inseridos pelo usuário no console, facilitando a interação com o sistema.

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

O uso de classes de repositório melhorou a organização e modularidade do código. Separando a lógica de manipulação de dados das operações de entrada e saída, as classes de repositório facilitaram a manutenção e a escalabilidade do sistema. Isso resultou em um código mais limpo, com responsabilidades bem definidas para gerenciamento de dados.

