



Universidade:[Estácio]

Campus: [1685 Polo Anhaia Melo - São Paulo - SP]

Curso: [Desenvolvimento Full Stack]

Disciplina: [Nível 1: Iniciando o Caminho Pelo Java]

Turma: [9001]

Semestre Letivo: [2024.3]

Integrantes da Prática: [Fernando Garcia Farias]

Relatório discente de acompanhamento

1. Iniciando o caminho pelo Java - Implementação de um cadastro de clientes com persistência em arquivos.
2. O objetivo da prática apresentada é permitir que os alunos utilizem herança, polimorfismo e persistência de objetos em arquivos binários na definição de entidades para implementar um sistema de cadastro de clientes em Java.
3. Todos os códigos solicitados neste roteiro de aula:

Desenvolvimento de um sistema de cadastro de clientes em Java, com persistência em arquivos binários; Criação de entidades, utilizando herança e polimorfismo. Os dados são gerenciados por meio de repositórios, que incluem operações de inserção, alteração, exclusão, obtenção e persistência em arquivos. Implemento do método "Main" para testar o funcionamento do sistema, incluindo a adição de clientes, persistência e recuperação de dados.

4. Os resultados da execução dos códigos foram conforme esperado, com o sistema de cadastro de clientes funcionando corretamente. As entidades foram criadas, os dados foram armazenados em arquivos binários e as operações de

inserção, alteração, exclusão, obtenção e recuperação foram realizadas com sucesso.

5. Análise e Conclusão:

Vantagens e desvantagens do uso de herança:

- Vantagens: A herança permite reutilização de código, facilitando a manutenção e evolução do sistema, além de promover a organização das classes, tornando o código mais compreensível.
- Desvantagens: Herança pode levar a um acoplamento excessivo entre classes, tornando o sistema mais complexo. Além disso, mudanças na classe pai podem afetar suas subclasses, o que pode causar problemas de compatibilidade.

Necessidade da interface Serializable na persistência em arquivos binários:

- A interface Serializable é necessária para que os objetos Java possam ser convertidos em bytes para serem armazenados em arquivos ou transmitidos pela rede. Ao implementar a interface Serializable, uma classe indica que seus objetos podem ser convertidos em uma sequência de bytes, o que é essencial para a persistência em arquivos binários.

Uso do paradigma funcional pela API stream no Java:

- A API Stream no Java permite operações de processamento de coleções de forma mais expressiva, utilizando o paradigma funcional, que permite operações como filtragem, mapeamento, redução e ordenação de elementos de uma coleção de forma eficiente. O paradigma funcional é utilizado para passar funções como argumentos para métodos da API Stream, permitindo operações complexas de forma simples e elegante.

Padrão de desenvolvimento adotado na persistência de dados em arquivos no Java:

- No Java, o padrão de desenvolvimento mais comum para persistência de dados em arquivos é o uso de classes de entrada e saída (I/O) fornecidas pela API Java, como FileInputStream e FileOutputStream para leitura e escrita de arquivos binários. A interface Serializable é frequentemente utilizada para serializar objetos antes de serem gravados em arquivos e desserializar objetos após a leitura dos arquivos.

<https://github.com/Futureforworld/1-Procedimento-Cria-o-das-Entidades-e-Sistema-de-Persist-ncia>