



Estácio

Aluno: Wallace Felipe Tavares Moreira

Polo: Itaguaí/RJ

Matrícula: 202109237331

Curso: Desenvolvimento Full Stack

1 - Título da Prática;

Iniciando o caminho pelo Java

2 - Objetivo 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.

3 - Todos os códigos solicitados neste roteiro de aula;

CadastroPOO.java: classe principal que contém o método main que interage com o usuário através do console.

Pacote model:

Pessoa.java: classe abstrata que representa uma pessoa, com os atributos id e nome, e métodos para exibição e manipulação desses dados.

PessoaFisica.java: classe que herda de Pessoa e adiciona os atributos cpf e idade, além de métodos específicos.

PessoaJuridica.java: classe que herda de Pessoa e adiciona o atributo cnpj, além de métodos específicos.

PessoaFisicaRepo.java: classe de gerenciamento para entidades de pessoa física, com métodos para inserir, alterar, excluir e dados, além de persistência e recuperação em arquivos.

PessoaJuridicaRepo.java: classe de gerenciamento para entidades de pessoa jurídica, com métodos para inserir, alterar, excluir e dados, além de permanência e recuperação em arquivos.

4 - Os resultados da execução dos códigos também devem ser apresentados;

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

```
Digite o tipo de entidade (Física ou Jurídica):  
F  
ID: 1  
Nome: WALLACE  
CPF: 1233211234  
-----
```

```
Digite o tipo de entidade (Física ou Jurídica):  
J  
ID: 1  
Nome: DEWALLACE  
CNPJ: 123422229999  
-----
```

👉 2º Procedimento | Criação do Cadastro em Modo Texto

```
Output - Run (CadastroPOO) x
--- exec-maven-plugin:3.1.0:exec (default-cli) @ Cadast.
Digite a opção desejada:
1 - Incluir
2 - Alterar
3 - Excluir
4 - Obter
5 - Obter todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar a execução

1
Digite o tipo de entidade (Física ou Jurídica):
F
Digite o ID:
06
Digite o nome:
Wallace Tavares
Digite o CPF:
Pessoa física adicionada com sucesso.
```

5 - Análise e Conclusão:

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

Vantagens da herança:

Reutilização de código, extensibilidade, polimorfismo.

Desvantagens da herança:

Acoplamento, hierarquia complexa, problemas de herança múltipla e diamante, coesão fraca.

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

A interface `Serializable` é necessária ao efetuar persistência em arquivos binários porque indica que a classe pode ser convertida em bytes e armazenada em disco, permitindo a gravação e recuperação de objetos de forma transparente.

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

A API Stream utiliza o paradigma funcional para manipular coleções de dados de forma concisa e expressiva, usando funções de ordem superior, expressões lambda e operações de redução.

Isso permite escrever código mais legível, modular e eficiente.

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

Ao trabalhar com Java, o padrão de desenvolvimento comumente adotado para a persistência de dados em arquivos é o padrão de Serialização.

A serialização permite que objetos Java sejam convertidos em uma sequência de bytes e salvos em arquivos, para posterior recuperação e desserialização dos dados.

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

Elementos estáticos em Java são membros de uma classe que pertencem à própria classe em vez de pertencerem a instâncias individuais dessa classe. Isso significa que eles são compartilhados por todas as instâncias da classe.

O método main é declarado como estático porque ele precisa ser acessado sem a necessidade de criar uma instância da classe. Como o método main é o ponto de entrada do programa Java, ele é invocado diretamente pela JVM (Java Virtual Machine) antes de qualquer objeto ser criado.

Portanto, ele precisa ser estático para poder ser chamado sem a necessidade de criar um objeto da classe que o contém.

f - Para que serve a classe Scanner?

A classe Scanner em Java é usada para ler dados de diferentes fontes de entrada, como o teclado (entrada padrão), arquivos ou strings.

Ela fornece métodos convenientes para ler diferentes tipos de dados, como inteiros, números de ponto flutuante, strings, entre outros.

Com a classe Scanner, você pode ler dados digitados pelo usuário no console, ler dados de um arquivo de texto ou ler dados de uma string, facilitando a interação com o usuário e o processamento de dados em um programa Java.

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

O uso de classes de repositório na organização do código separa a lógica de acesso a dados da lógica de negócios, resultando em um código mais modular e de fácil manutenção.
