



Missão Prática | Nível 1 | Mundo 3

Filipe Alvim Santos - 202208291325

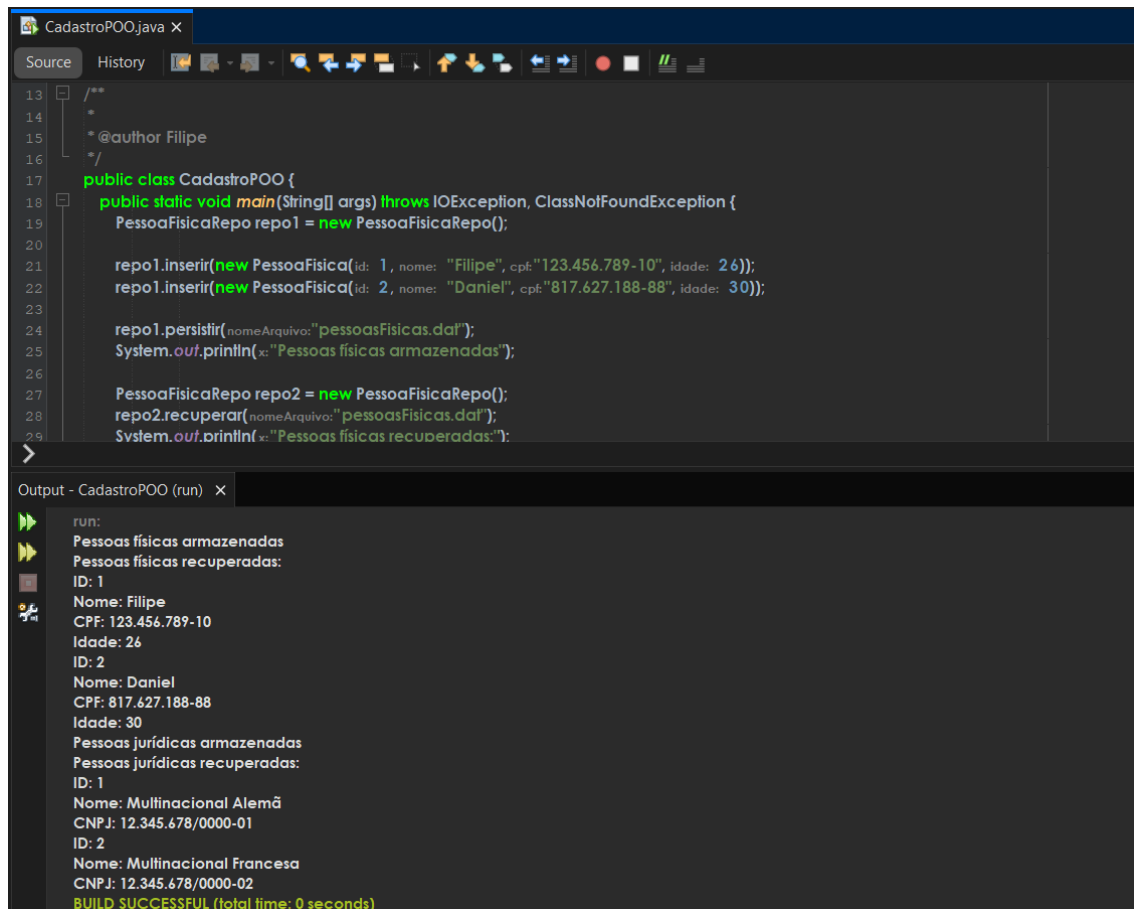
**Campus Polo Sulacap – RJ / Desenvolvimento Full Stack
Nível 1: Iniciando o caminho pelo Java – Turma: 22.3 – 3º Semestre**

Objetivo da Prática

Os objetivos da prática são: Utilizar herança e polimorfismo na definição de entidades, utilizar persistência de objetos em arquivos binários, implementar uma interface cadastral em modo texto e utilizar o controle de exceções da plataforma Java.

No final do projeto, terei implementado um sistema cadastral em Java, utilizando os recursos da programação orientada a objetos e a persistência em arquivos binários.

1º Procedimento– Criação das Entidades e Sistema de Persistência



```
13  /**
14   *
15   * @author Filipe
16   */
17  public class CadastroPOO {
18      public static void main(String[] args) throws IOException, ClassNotFoundException {
19          PessoaFisicaRepo repo1 = new PessoaFisicaRepo();
20
21          repo1.inserir(new PessoaFisica(id: 1, nome: "Filipe", cpf:"123.456.789-10", idade: 26));
22          repo1.inserir(new PessoaFisica(id: 2, nome: "Daniel", cpf:"817.627.188-88", idade: 30));
23
24          repo1.persistir(nomeArquivo:"pessoasFisicas.dat");
25          System.out.println("Pessoas físicas armazenadas");
26
27          PessoaFisicaRepo repo2 = new PessoaFisicaRepo();
28          repo2.recuperar(nomeArquivo:"pessoasFisicas.dat");
29          System.out.println("Pessoas físicas recuperadas");
30      }
31  }
```

```
run:
Pessoas físicas armazenadas
Pessoas físicas recuperadas:
ID: 1
Nome: Filipe
CPF: 123.456.789-10
Idade: 26
ID: 2
Nome: Daniel
CPF: 817.627.188-88
Idade: 30
Pessoas jurídicas armazenadas
Pessoas jurídicas recuperadas:
ID: 1
Nome: Multinacional Alemã
CNPJ: 12.345.678/0000-01
ID: 2
Nome: Multinacional Francesa
CNPJ: 12.345.678/0000-02
BUILD SUCCESSFUL (total time: 0 seconds)
```

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

Resposta: A herança na programação orientada a objetos tem várias vantagens e desvantagens. As vantagens incluem a reutilização de código, pois as subclasses podem herdar campos e métodos da superclasse, a organização do código ao agrupar classes relacionadas e a extensibilidade, pois as subclasses podem adicionar seus próprios campos e métodos. As desvantagens incluem o acoplamento elevado, pois as subclasses estão fortemente ligadas à superclasse, a inflexibilidade, pois uma vez definida uma hierarquia de herança, pode ser difícil alterá-la, e a ocultação de informações, onde uma subclasse pode inadvertidamente afetar o comportamento da superclasse.

b) Porque a interface Serializable é necessária ao efetuar persistência em arquivos binários?

Resposta: A interface Serializable é necessária para a persistência em arquivos binários porque ela permite que os objetos sejam convertidos em um formato que pode ser gravado em disco ou transmitido através de uma rede. Isso é conhecido como serialização. Sem a implementação dessa interface, o Java não saberia como transformar o estado do objeto em uma representação que pode ser salva em um arquivo binário ou enviada através de uma rede. Portanto, qualquer objeto que você deseja persistir em um arquivo binário deve implementar a interface Serializable.

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

Resposta: A API Stream no Java utiliza o paradigma funcional ao permitir operações em sequências de elementos, como map, filter e reduce, de uma maneira declarativa. Essas operações podem aproveitar os recursos de programação funcional do Java, como expressões lambda e métodos de referência, para fornecer um alto nível de abstração e expressividade. Além disso, a API Stream suporta processamento paralelo, o que pode melhorar significativamente o desempenho para grandes conjuntos de dados.

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

Resposta: Ao trabalhar com Java, o padrão de desenvolvimento comumente adotado para a persistência de dados em arquivos é o padrão Data Access Object (DAO). Este padrão abstrai e encapsula todas as operações de acesso a dados em um objeto DAO, permitindo que os detalhes de acesso a dados sejam alterados sem impactar o código que usa o DAO.

2º Procedimento – Criação do Cadastro em Modo Texto

```
run:
---- Menu ----
1. Incluir
2. Alterar
3. Excluir
4. Exibir pelo id
5. Exibir todos
6. Salvar dados
7. Recuperar dados
0. Sair
Escolha uma opção:
1
Escolha o tipo (Física - 1 / Jurídica - 2):
1
Digite o id:
1
Digite o nome:
Filipe
Digite o CPF:
123.124.125-12
Digite a idade:
27
Pessoa física adicionada com sucesso.
```

```
---- Menu ----
1. Incluir
2. Alterar
3. Excluir
4. Exibir pelo id
5. Exibir todos
6. Salvar dados
7. Recuperar dados
0. Sair
Escolha uma opção:
4
Escolha o tipo (Física - 1 / Jurídica - 2):
1
Digite o id da pessoa que deseja exibir:
1
ID: 1
Nome: Filipe
CPF: 123.124.125-12
Idade: 27
```

```
---- Menu ----
1. Incluir
2. Alterar
3. Excluir
4. Exibir pelo id
5. Exibir todos
6. Salvar dados
7. Recuperar dados
0. Sair
Escolha uma opção:
5
Escolha o tipo (Física - 1 / Jurídica - 2):
1
Pessoas físicas:
ID: 1
Nome: Filipe
CPF: 123.124.125-12
Idade: 27
```

```
---- Menu ----
1. Incluir
2. Alterar
3. Excluir
4. Exibir pelo id
5. Exibir todos
6. Salvar dados
7. Recuperar dados
0. Sair
Escolha uma opção:
2
Escolha o tipo (Física - 1 / Jurídica - 2):
1
Digite o id da pessoa que deseja alterar:
1
Dados atuais da pessoa física:
ID: 1
Nome: Filipe
CPF: 123.124.125-12
Idade: 27
Digite o novo nome:
Daniel
Digite o novo CPF:
111.111.111-11
Digite a nova idade:
30
Pessoa física alterada com sucesso.
```

```
---- Menu ----
1. Incluir
2. Alterar
3. Excluir
4. Exibir pelo id
5. Exibir todos
6. Salvar dados
7. Recuperar dados
0. Sair
Escolha uma opção:
3
Escolha o tipo (Física - 1 / Jurídica - 2):
1
Digite o id da pessoa que deseja excluir:
1
Pessoa física removida com sucesso.
```

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

Resposta: Elementos estáticos em Java, como campos e métodos, pertencem à classe em si, em vez de a uma instância da classe. Isso significa que eles podem ser acessados sem a necessidade de criar um objeto da classe. O método main em Java é estático porque ele precisa ser acessado pelo ambiente de execução Java quando o programa é iniciado, antes de qualquer objeto ser criado. Portanto, ele deve ser estático para que possa ser chamado sem a necessidade de instanciar a classe.

b) Para que serve a classe Scanner?

Resposta: A classe Scanner em Java é usada para ler a entrada do usuário. Ela pode ler vários tipos de dados, como inteiros, strings, doubles, etc. A classe Scanner é geralmente usada para ler a entrada do teclado, mas também pode ser usada para ler de outras fontes, como arquivos.

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

Resposta: O uso de classes de repositório teve um impacto significativo na organização do código. Elas fornecem uma abstração do armazenamento de dados, permitindo que o código que usa os repositórios não precise se preocupar com os detalhes de como os dados são armazenados e recuperados. Isso leva a um código mais limpo e mais fácil de manter. Além disso, as classes de repositório promovem a separação de responsabilidades,

pois cada repositório tem uma responsabilidade clara e bem definida. Isso torna o código mais modular e mais fácil de testar.