



Resultados da Execução dos Códigos Resultados da Execução dos Códigos

Nome do campus: Salgado Filho

Nome do curso: Full Stack

Nome da Disciplina: Iniciando o caminho pelo Java

número da Turma: 9001

semestre letivo: 2024.3

nome dos integrantes da Prática: Misael Virginio

Trabalhando com Herança, Persistência de Dados e API Stream em Java

Objetivo da Prática

Implementar um sistema que utilize conceitos de herança, interfaces, persistência de dados em arquivos binários e manipulação de coleções com a API Stream do Java.

Resultados da Execução dos Códigos

```
1 package cadastr poo;
2
3 import java.io.IOException;
4 import java.util.Scanner;
5 import model.PessoaFisica;
6 import model.PessoaFisicaRepo;
7 import model.PessoaJuridica;
8 import model.PessoaJuridicaRepo;
9
10 public class CadastroPOO {
11     public static void main(String[] args) {
12         try (Scanner scanner = new Scanner(System.in)) {
```

Notifications

Output X

CadastroPOO (run) X

CadastroPOO (run) #2 X

run:

Menu:

- 1 - Incluir
- 2 - Alterar
- 3 - Excluir
- 4 - Exibir pelo ID
- 5 - Exibir todos
- 6 - Salvar dados
- 7 - Recuperar dados
- 0 - Sair

Selecione uma opç o: |

CadastroPOO (run) #2

running_

X (1 more...)

14:72

INS

```
1 package cadastr poo;
2
3 import java.io.IOException;
4 import java.util.Scanner;
5 import model.PessoaFisica;
6 import model.PessoaFisicaRepo;
7 import model.PessoaJuridica;
8 import model.PessoaJuridicaRepo;
9
10 public class CadastroPOO {
11     public static void main(String[] args) {
12         try (Scanner scanner = new Scanner(System.in)) {
```

Notifications Output X

CadastroPOO (run) X CadastroPOO (run) #2 X

```
run:
Menu:
1 - Incluir
2 - Alterar
3 - Excluir
4 - Exibir pelo ID
5 - Exibir todos
6 - Salvar dados
7 - Recuperar dados
0 - Sair
Selecione uma opçao: 1
Incluir Pessoa Física (1) ou Jurídica (2)? 1
ID: 1
Nome: Jorge
CPF: 11111111111
Idade: 21
```

CadastroPOO (run) #2 running... X (1 more...) 14:72 INS

```
1 package cadastr poo;
2
3 import java.io.IOException;
4 import java.util.Scanner;
5 import model.PessoaFisica;
6 import model.PessoaFisicaRepo;
7 import model.PessoaJuridica;
8 import model.PessoaJuridicaRepo;
9
10 public class CadastroPOO {
11     public static void main(String[] args) {
12         try (Scanner scanner = new Scanner(System.in)) {
```

Notifications Output X

CadastroPOO (run) X CadastroPOO (run) #2 X

run:

Menu:

- 1 - Incluir
- 2 - Alterar
- 3 - Excluir
- 4 - Exibir pelo ID
- 5 - Exibir todos
- 6 - Salvar dados
- 7 - Recuperar dados
- 0 - Sair

Selecione uma opção: 5

Exibir todos (1 - Pessoas Físicas, 2 - Pessoas Jurídicas, 3 - Todos):

1

Pessoas Físicas:

ID: 1, Nome: Jorge

CPF: 11111111111, Idade: 19

CadastroPOO (run) #2 running... X (1 more...) 221:13 INS

Análise e Conclusão

1. Vantagens e desvantagens do uso de herança:

- *Vantagens*: Reutilização de código e hierarquia clara.
- *Desvantagens*: Acoplamento elevado e flexibilidade reduzida.

2. Por que a interface `Serializable` é necessária?

Necessária para converter objetos em bytes, permitindo a gravação e recuperação em arquivos.

3. Como a API `Stream` utiliza o paradigma funcional?

Processa coleções de forma declarativa com operações como `filter`, `map` e `reduce`.

4. Qual padrão de desenvolvimento é adotado para persistência em Java?

O padrão *Repository* isola a lógica de acesso a dados, melhorando a manutenção.

5. O que são elementos estáticos?

Associados à classe, permitem que o método `main` seja chamado sem instância.

6. Para que serve a classe Scanner?

Lê entradas do usuário e arquivos, facilitando a captura de dados.

7. Impacto das classes de repositório na organização do código:

Melhora a separação de responsabilidades e facilita a manutenção.