

Nome: Michel Angelo da Silva Portugal .

Matrícula: 202208826174

Campus: Duque de Caxias

Disciplina: Iniciando o caminho pelo Java / RPG: 0014 / Sem 2/2023.

Objetivo da Prática:

1. Utilizar herança e polimorfismo na definição de entidades.
2. Utilizar persistência de objetos em arquivos binários.
3. Implementar uma interface cadastral em modo texto.
4. Utilizar o controle de exceções da plataforma Java.

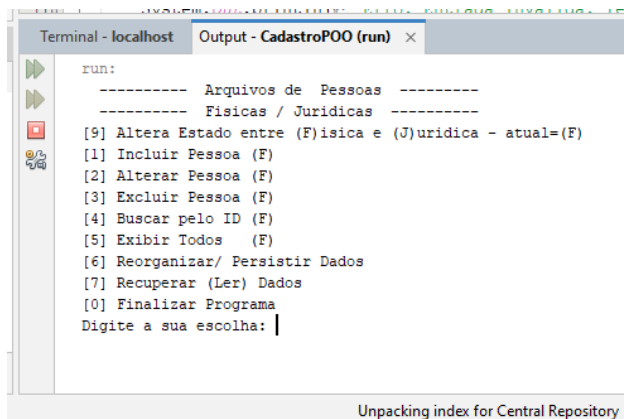
Link do Repositório:

<https://github.com/miaspe/JAVA>

Código na Prática:

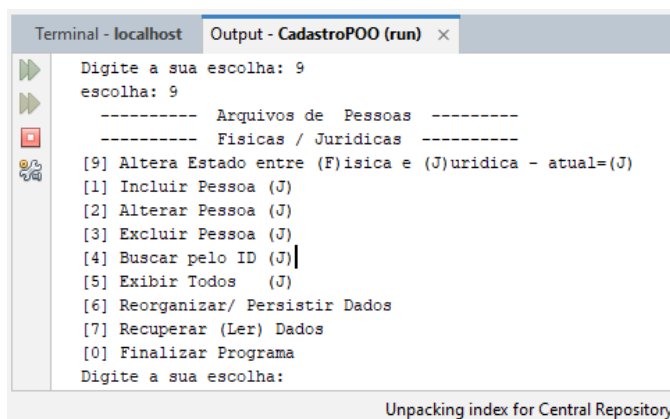
Abaixo apresentamos os códigos pertinentes às funcionalidades de um simples menu de: Cadastro, Alteração, Reinclusão e Listagem de CPFs e CNPJs em cadastro array fazendo persistência em um arquivo (.bin):

Começando pelo menu composto das opções pertinentes as funcionalidades do referido programa:



```
run:
----- Arquivos de Pessoas -----
----- Físicas / Jurídicas -----
[9] Altera Estado entre (F)ísica e (J)urídica - atual=(F)
[1] Incluir Pessoa (F)
[2] Alterar Pessoa (F)
[3] Excluir Pessoa (F)
[4] Buscar pelo ID (F)
[5] Exibir Todos (F)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha: |
```

Fig1:



```
Terminal - localhost Output - CadastroPOO (run) x
Digite a sua escolha: 9
escolha: 9
----- Arquivos de Pessoas -----
----- Físicas / Jurídicas -----
[9] Altera Estado entre (F)ísica e (J)urídica - atual=(J)
[1] Incluir Pessoa (J)
[2] Alterar Pessoa (J)
[3] Excluir Pessoa (J)
[4] Buscar pelo ID (J)|
[5] Exibir Todos (J)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha:
```

Fig:2

Como pode ser notado nas figuras de execução do programa, temos na figura1 o start inicial com a opção de cadastrar pessoas físicas ou seja CPFs e Nomes ; E na fig 2 com a opção (9) selecionada o programa cambeia para pessoa jurídica onde não mais cadastrará CPFs e sim CNPJs.

Nas figuras 1.1 e 2.1 podemos verificar a continuação da execução tanto do módulo pessoa física quanto jurídica:

```
Terminal - localhost  Output - CadastroPOO (run) x
ID: 1
Nome: qqqqqqqqqqqqqqqqqqqqqq
CPF: 12345678900
Idade: 11

ID: 2
Nome: wwwwwwwwwwwwwwwwwwwwwwww
CPF: 12345678922
Idade: 22

ID: 3
Nome: eeeeeeeeeeeeeeeeeeeeee
CPF: 12345678933
Idade: 33

ID: 4
Nome: rrrrrrrrrrrrrrrrrrrrrr
CPF: 12345678944
Idade: 44

Digite o nome da pessoa (ou 'sair' para encerrar): |
```

Fig 1.1(pessoa física)

```
Terminal - localhost  Output - CadastroPOO (run) x
Nome: xxxxxxxxxxxxxxxxxxxxxx
CPF: 12345678933333

ID: 3
Nome: jjjjjjjjjjjjjjjjjjjjjjj
CPF: 1234123412341234

ID: 4
Nome: vvvvvvvvvvvvvvvvvvvvvv
CPF: 12345678901234

ID: 5
Nome: bbbbbbbbbbbbbbbbbb
CPF: 98765432177755

ID: 6
Nome: nnnnnnnnnnnnnnnnnnnnnn
CPF: 987654987654987654
```

Fig 2.1(pessoa jurídica)

Agora vamos fazer a operação de (2)alterar pessoa física,repere que digitamos a opção e o programa nos avisa que existem 4 registros no total e o registro 2 que foi o registro escolhido pra alteração está disponível:

```
Terminal - localhost  Output - CadastroPOO (run) x
----- Arquivos de Pessoas -----
----- Físicas / Jurídicas -----

[9] Altera Estado entre (F)ísica e (J)urídica - atual=(F)
[1] Incluir Pessoa (F)
[2] Alterar Pessoa (F)
[3] Excluir Pessoa (F)
[4] Buscar pelo ID (F)
[5] Exibir Todos (F)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha: 2
escolha: 2
Digite o ID da pessoa que deseja ler: 2
maior ID :4
Pessoa encontrada:

ID: 2
Nome: wwwwwwwwwwwwwwwwwwwwwwww
CPF: 12345678922
Idade: 22

Digite o novo Nome: kkkkkkkkkkkk
Digite o novo CPF: 66677788822
Digite a nova Idade: 22
```

Fig 1.3

Foi digitado o novo registro 2:

Como: Nome :kkkkk

CPF:66677788822

Idade:22

E agora vamos relistar os registros pra verificar se foi tudo bem mesmo na fig 1.4 e 1.5 abaixo:

```
ID: 2
Nome: wwwwwwwwwwwwwwwwwwwwwwww
CPF: 12345678922
Idade: 22

Digite o novo Nome: kkkkkkkkkkkk
Digite o novo CPF: 66677788822
Digite a nova Idade: 22
Atualizado com sucesso!
```

Fig 1.4

```
Terminal - localhost  Output - CadastroPOO (run) x
Digite a sua escolha: 5
escolha: 5
Pessoas cadastradas:

ID: 1
Nome: qqqqqqqqqqqqqqqqqqqqqq
CPF: 12345678900
Idade: 11

ID: 2
Nome: kkkkkkkkkkkkkkkkkk
CPF: 66677788822
Idade: 22

ID: 3
Nome: eeeeeeeeeeeeeeeeeeeeee
CPF: 12345678933
Idade: 33

ID: 4
Nome: rrrrrrrrrrrrrrrrrrrrrr
CPF: 12345678944
Idade: 44

----- Arquivos de Pessoas -----
```

Fig 1.5

Na figura ao lado vemos a função listar (5) Exibir todos funcionando perfeitamente verificamos que o registro (wwwwww) foi atualizado pra (kkkkkkkk) conforme o desejado, com um adendo de que no nosso programa a persistência ocorre a cada alteração não sendo necessária que seja pedida pelo usuário mas mesmo assim está disponível na opção (6) que reúne também a função reorganizar, vamos explica-la melhor abaixo:

### Função (6) – Reorganizar / Persistir Dados:

Conforme já foi dito em nosso programa a persistência ocorre a cada alteração não sendo necessária que seja pedida pelo usuário. Assim o risco da necessidade de redigitação por erro do usuário após a inclusão de vários dados sucessivos pra somente no fim pedir ou esquecer de fazer a persistência fica afastado.

Agora pra função Reorganizar demonstraremos a função (3) Excluir, primeiro... conforme abaixo:

```
Terminal - localhost  Output - CadastroPOO (run) x
----- Arquivos de Pessoas -----
----- Físicas / Jurídicas -----
[9] Altera Estado entre (F)ísica e (J)urídica - atual=(F)
[1] Incluir Pessoa (F)
[2] Alterar Pessoa (F)
[3] Excluir Pessoa (F)
[4] Buscar pelo ID (F)
[5] Exibir Todos (F)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha: 3
escolha: 3
Digite o ID da pessoa para excluir (ou 'sair' para encerrar): 2
Pessoa excluída com sucesso!
Digite o ID da pessoa para excluir (ou 'sair' para encerrar): sair
Arquivo salvo com sucesso!
```

Na listagem abaixo vemos que o arquivo já não mais apresenta o ID2 pois este foi excluído conforme figura acima:

```
Terminal - localhost  Output - CadastroPOO (run) x
Digite o ID da pessoa para excluir (ou 'sair' para encerrar): sair
Arquivo salvo com sucesso!
----- Arquivos de Pessoas -----
----- Físicas / Jurídicas -----
[9] Altera Estado entre (F)ísica e (J)urídica - atual=(F)
[1] Incluir Pessoa (F)
[2] Alterar Pessoa (F)
[3] Excluir Pessoa (F)
[4] Buscar pelo ID (F)
[5] Exibir Todos (F)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha: 5
escolha: 5
Pessoas cadastradas:

ID: 1
Nome: qqqqqqqqqqqqqqqqqqqqqqq
CPF: 12345678900
Idade: 11

ID: 3
Nome: eeeeeeeeeeeeeeeeeeeeeee
CPF: 12345678933
Idade: 33

ID: 4
Nome: rrrrrrrrrrrrrrrrrrrrrrrr
CPF: 12345678944
Idade: 44
```

Agora vamos reincluir o ID2 com a opção (2) veja ao lado

```
Terminal - localhost  Output - CadastroPOO (run) x
----- Arquivos de Pessoas -----
----- Físicas / Jurídicas -----
[9] Altera Estado entre (F)ísica e (J)urídica - atual=(F)
[1] Incluir Pessoa (F)
[2] Alterar Pessoa (F)
[3] Excluir Pessoa (F)
[4] Buscar pelo ID (F)
[5] Exibir Todos (F)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha: 2
escolha: 2
Digite o ID da pessoa que deseja ler: 2
maior ID :4
Nenhum arquivo encontrado com o ID informado.
Deseja registrar arquivo pra posição 2 ? (S/N) s
Digite o novo Nome: gggggggggg
Digite o novo CPF: 33344455522
Digite a nova Idade: 21
Atualizado com sucesso!
```

```
Terminal - localhost  Output - CadastroPOO (r
[4] Buscar pelo ID (F)
[5] Exibir Todos (F)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha: 5
escolha: 5
Pessoas cadastradas:

ID: 1
Nome: qqqqqqqqqqqqqqqqqqqqqqq
CPF: 12345678900
Idade: 11

ID: 2
Nome: ggggggggggg
CPF: 33344455522
Idade: 21

ID: 3
Nome: eeeeeeeeeeeeeeeeeeeeeee
CPF: 12345678933
Idade: 33

ID: 4
Nome: rrrrrrrrrrrrrrrrrrrrrrrr
CPF: 12345678944
Idade: 44
```

Agora listamos todos os registros com a opção 5 e verificamos o sucesso da operação: notando que o registro ID2 agora passou a ser (gggggggg , 33344455522, 21)

Opção (7) Recuperar (Ler) Dados:

Essa função não foi implementada pelo simples fato de que, como o salvamento é automático no arquivo (.bin), foi verificado que a função (5) Exibir Todos , acaba por desempenhar a mesma função.

Opção (0) Finalizar Programa. É auto explicativa.

Link do Repositório:

<https://github.com/miaspe/JAVA>

Perguntas a serem respondidas:

Quais as vantagens e desvantagens do uso de herança?

R: Uma das principais vantagens da herança é a de reaproveitar estados e comportamentos (métodos), economizando esforço na reescrita de código.

Uma desvantagem é a dependência das subclasses dos atributos das classes acima o que pode gerar um evento de quebra do código quando a classe pai é manipulada pois haverá como o próprio nome diz herança deixada pra classes filhos, dificultando possivelmente, a manutenção do código.

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

R: A interface Serializable, é tão somente um commando que faz com que o interpretador JAVA enxergue que o programa quer gravar uma determinada informação, assim ele vai organizar (serializar) e “traduzir” a informação desejada pro format binário e salvar o arquivo; O mesmo ocorre no sentido contrário, quando se deseja recuperar a informação salva anteriormente, que será retraduzida desta vez do formato binário ao formato original pra ser apresentada ao usuário ou usada pelo programa.

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

R: A API stream é uma funcionalidade implementada a partir do JAVA 8, permite processamento e manipulação de sequências de dados através de subcomandos como (List, Set, Map) promovendo agilidade no processamento.

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

R: Em JAVA temos os já conhecidos de outras linguagens o JSON e o XML, somados a eles temos ainda GSON e o que usamos em nosso código o Serializable.

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

R: São elementos que atendem a todas as classes sem ficar “preso”(ou pertencer a alguma delas)

Para que serve a classe Scanner?

R: Ler (Varrer) o buffer de teclado ou seja registrar o que é digitado.

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

R: “Facilitou” a organização (separação por categoria) dos Dados.

Conclusão:

Elabore uma análise crítica da sua Missão Prática.

R: Achei muito interessante o JAVA, na minha humilde análise fica entre o python e o JS, falta ainda a prática do dia – dia, pra uma análise mais acurada sobre a real preferência, frente às necessidades da resolução de um problema que venha a ser apresentado.

Como legado a frase do meu avô que dizia...\_ Ferramenta não é gasto, é investimento..., ou seja, quanto mais ferramentas (linguagens de programação) você conhecer, mais apto e capaz a solucionar a questão proposta você se torna.

Foi pedido listar o código: porém verifiquei o tamanho de todos os submódulos (com estimativa de mais de 30 folhas) e como é de fácil verificação aqui mesmo no GIT listei apenas o início que é o menu principal da aplicação, os demais módulos que são as escolhas do menu podem ser visualizadas nessa pasta:

Anexo 1: menu principal em java:

```
package cadastropoo;

// * @author miaspe

import java.io.*;
import java.io.BufferedReader;
import java.io.FileWriter;
import java.util.Scanner;
import java.util.InputMismatchException;
import java.io.IOException;

public class CadastroPOO {
    static String E_Pessoa = "F";
    static String MarcaMenu = "";
    static Scanner scanner = new Scanner(System.in);

    //-----

    public static void limparTela() {
        try {
            if (System.getProperty("os.name").contains("Windows")) {
                new ProcessBuilder("cmd", "/c", "cls").inheritIO().start().waitFor();
            } else {
                System.out.flush();
            }
        } catch (Exception e) {
            // Lidar com exceções
            e.printStackTrace();
        }
    }

    public static void main(String[] args) throws Exception {
        int loop = 1;
        while (loop == 1){// TODO code application logic here
            limparTela();
            System.out.println(" ----- Arquivos de Pessoas -----");
            System.out.println(" ----- Fisicas / Juridicas -----");
            System.out.println("[9] Altera Estado entre (F)isica e (J)uridica - atual=(" + E_Pessoa + ")");
            System.out.println("[1] Incluir Pessoa (" + E_Pessoa + ")");
            System.out.println("[2] Alterar Pessoa (" + E_Pessoa + ")");
            System.out.println("[3] Excluir Pessoa (" + E_Pessoa + ")");
            System.out.println("[4] Buscar pelo ID (" + E_Pessoa + ")");
```

```
System.out.println("[5] Exibir Todos  (" + E_Pessoa + ")");
System.out.println("[6] Reorganizar/ Persistir Dados");
System.out.println("[7] Recuperar (Ler) Dados");
System.out.println("[0] Finalizar Programa");
```

```
System.out.print("Digite a sua escolha: ");
String escolhaMenu = scanner.nextLine();
MarcaMenu = escolhaMenu;
System.out.println("escolha: " + escolhaMenu);
```

```
try{
switch (escolhaMenu) {
    case "1"://System.out.println("1-");
        if(E_Pessoa.equals("F")){
            CadastroPessoaFisica.main(args);
            break;
        }else{
            CadastroPessoaJuridica.main(args);
            break;
        }
    case "2"://System.out.println("2-");
        if(E_Pessoa.equals("F")){
            LerPessoaFisicald.main(args);
            break;
        }else{
            LerPessoaJuridicald.main(args);
            break;
        }
    case "3"://System.out.println("3-");
        if(E_Pessoa.equals("F")){
            ExcluirRegistro.main(args);
            break;
        }else{
            ExcluirRegistroJ.main(args);
            break;
        }
    case "4"://System.out.println("4-");
        if(E_Pessoa.equals("F")){
            LerPessoaFisicald.main(args);
```

```

        break;
    }else{
        LerPessoaJuridicald.main(args);
        break;
    }
    case "5"://System.out.println("5-");
    if(E_Pessoa.equals("F")){
        LerPessoaFisica.main(args);
        break;
    }else{
        LerPessoaJuridica.main(args);
        break;
    }
    case "6"://System.out.println("6-");
    if(E_Pessoa.equals("F")){
        ReclassificarIds.main(args);
        break;
    }else{
        ReclassificarIdsJ.main(args);
        break;
    }
    case "7":System.out.println("7-");break;
    case "9"://System.out.println("9-");
    if(E_Pessoa.equals("F")){E_Pessoa = "J";}else{E_Pessoa = "F";}break;
    case "0":System.out.println("Encerrando o programa:...");loop=0;break;
    default:
        System.out.println("Escolha de novo");
    }}catch(InputMismatchException e){
    System.out.println("Erro: Entrada inválida. Tente de novo.");
    scanner.next();}
}

}

}

```