**Nome do campus:** Polo Pirituba
**Nome do curso:** Desenvolvimento Full Stack
**Nome da disciplina:** Nível 1: Iniciando o caminho pelo java
**Número da turma:** EAD - 9001
**Semestre letivo:** 2025.1 3 semestre
**Nome do integrante da prática:** Guilherme Wissenbach Ferreira
**Repositório Github:**

# **Título da prática:** Missão Prática | Nível 1 | Mundo 3

**Objetivo da prática:**
Implementação de um cadastro de clientes em modo texto, com persistência em arquivos, baseado na tecnologia Java.
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.
5 - 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.

**No pacote model criar as entidades, com as seguintes características:**

**Classe Pessoa**

Pessoa.java

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

package model;

/**
 *
 * @author Guilherme
 */

import java.io.Serializable;

public class Pessoa implements Serializable{
    private int id;
    private String nome;

    public Pessoa(int id, String nome) {
        this.id = id;
        this.nome = nome;
    }

    public int getId() {
        return id;
    }

    public void setId(int id){
        this.id = id;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public void exibir(){
        System.out.print("id: "+this.id + "\n" + "Nome: " + this.nome + "\n");
    }
}
```

## Classe Pessoa Fisica

PessoaFisica.java [-/A]

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

package model;

/**
 *
 * @author Guilherme
 */

import java.io.Serializable;

public class PessoaFisica extends Pessoa implements Serializable {

    private String cpf;
    private int idade;

    public PessoaFisica(int id, String nome, String cpf, int idade){
        super(id, nome);
        this.cpf = cpf;
        this.idade = idade;
    }

    public String getCpf() {
        return cpf;
    }

    public void setCpf(String cpf) {
        this.cpf = cpf;
    }

    public int getIdade(){
        return idade;
    }

    public void setIdade(int idade){
        this.idade = idade;
    }

    public void exibir(){
        System.out.print("\n"+"id: "+getId()+"\nNome: "+getNome()+ "\nCPF: "+this.cpf + "\n" + "Idade: " + this.idade + "\n");
    }
}

```

## Classe Pessoa Juridica

PessoaJuridica.java [-/A]

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

package model;

/**
 *
 * @author Guilherme
 */

import java.io.Serializable;

public class PessoaJuridica extends Pessoa implements Serializable{

    private String cnpj;

    public PessoaJuridica(int id, String nome, String cnpj){
        super(id, nome);
        this.cnpj = cnpj;
    }

    public String getCnpj() {
        return cnpj;
    }

    public void setCnpj(String cnpj) {
        this.cnpj = cnpj;
    }

    public void exibir(){
        System.out.print("\n" + "id: "+getId()+ "\nNome: "+getNome()+"\nCNPJ: "+this.cnpj+"\n");
    }
}

```

## No pacote model criar os gerenciadores, com as seguintes características:

## Classe PessoaFisicaRepo

PessoaFisicaRepo.java [-/A]

Source | History

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

package model;
import java.util.ArrayList;
import java.util.NoSuchElementException;
import java.util.Optional;
import java.io.*;

/**
 *
 * @author Guilherme
 */
public class PessoaFisicaRepo {

    private ArrayList<PessoaFisica> listaPessoasFisicas = new ArrayList<>();

    public void inserir(PessoaFisica pessoaFisica){
        listaPessoasFisicas.add(pessoaFisica);
    }

    public void alterar(PessoaFisica pessoaFisica, String novoNome, String novoCpf, int novaIdade) {
        pessoaFisica.setNome(novoNome);
        pessoaFisica.setCpf(novoCpf);
        pessoaFisica.setIdade(novaIdade);
    }

    public void excluir(int id) {
        try{
        listaPessoasFisicas.remove(obter(id));
        }catch(NoSuchElementException e){
            System.out.println("erro");
        }
    }

    public PessoaFisica obter(int id) {
        Optional<PessoaFisica> pessoaFisicaLocalizada = listaPessoasFisicas.stream().
                filter(pessoaFisica -> pessoaFisica.getId() == id).findFirst();
        if (pessoaFisicaLocalizada.isPresent()) {
           return pessoaFisicaLocalizada.get();
        } else {
           return null;
        }
    }

    public ArrayList<PessoaFisica> obterTodos(){
        return listaPessoasFisicas;
    }

    public void persistir(String arquivo)throws IOException {
        ObjectOutputStream arquivoSaida = new ObjectOutputStream(new FileOutputStream(arquivo));
        arquivoSaida.writeObject(listaPessoasFisicas);
        arquivoSaida.close();
        System.out.println("Dados de pessoas fisicas armazenados.");
    }

    public void recuperar(String arquivo) throws IOException, ClassNotFoundException {
        ObjectInputStream arquivoEntrada = new ObjectInputStream(new FileInputStream(arquivo));
        listaPessoasFisicas = (ArrayList<PessoaFisica>) arquivoEntrada.readObject();
        arquivoEntrada.close();
        System.out.println("Dados de pessoas fisicas recuperados.");
    }
}
```

## Classe PessoaJuridicaRepo

PessoaJuridicaRepo.java [-/A]

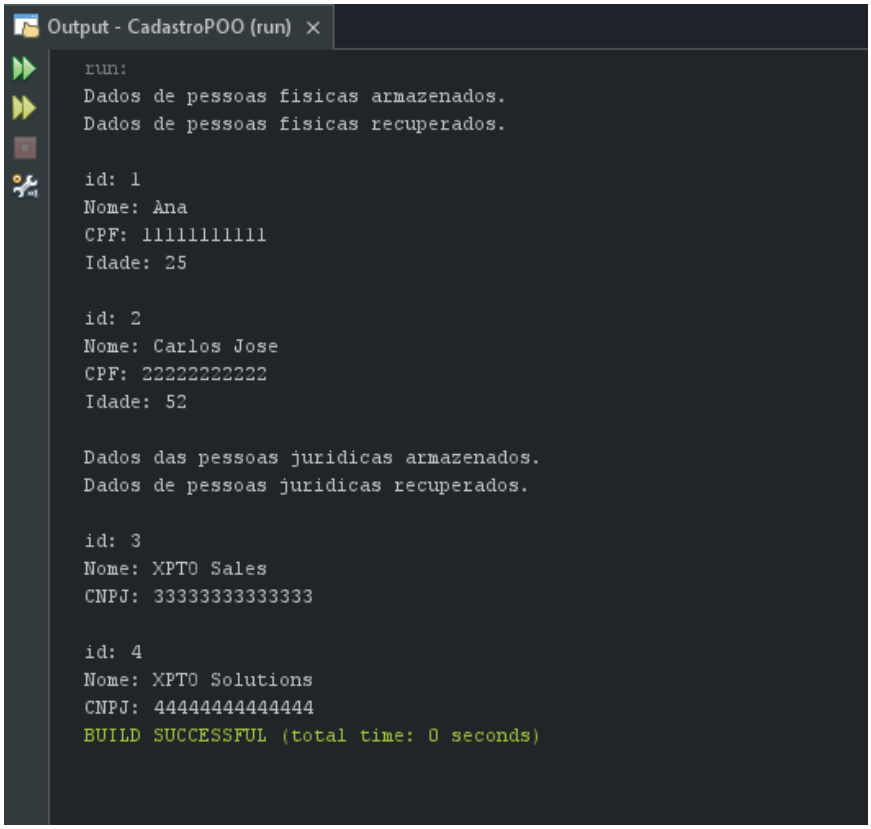
Source | History

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

package model;
import java.util.ArrayList;
import java.util.Optional;
import java.io.*;

/**
 *
 * @author Guilherme
 */
public class PessoaJuridicaRepo {

    private ArrayList<PessoaJuridica> listaPessoasJuridicas = new ArrayList<>();

    public void inserir(PessoaJuridica pessoaJuridica){
        listaPessoasJuridicas.add(pessoaJuridica);
    }

    public void alterar(PessoaJuridica pessoaJuridica, String novoNome, String novoCnpj) {
        pessoaJuridica.setNome(novoNome);
        pessoaJuridica.setCnpj(novoCnpj);
    }

    public void excluir(int id){
        listaPessoasJuridicas.remove(obter(id));
    }

    public PessoaJuridica obter(int id) {
        Optional<PessoaJuridica> pessoaJuridicaLocalizada = listaPessoasJuridicas.stream().
                filter(pessoaJuridica -> pessoaJuridica.getId() == id).findFirst();
        if (pessoaJuridicaLocalizada.isPresent()) {
           return pessoaJuridicaLocalizada.get();
        } else {
           return null;
        }
    }

    public ArrayList<PessoaJuridica> obterTodos(){
        return listaPessoasJuridicas;
    }

    public void persistir(String arquivo)throws IOException {
        ObjectOutputStream arquivoSaida = new ObjectOutputStream(new FileOutputStream(arquivo));
        arquivoSaida.writeObject(listaPessoasJuridicas);
        arquivoSaida.close();
        System.out.println("\nDados das pessoas juridicas armazenados.");
    }

    public void recuperar(String arquivo)throws IOException, ClassNotFoundException {
        ObjectInputStream arquivoEntrada = new ObjectInputStream(new FileInputStream(arquivo));
        listaPessoasJuridicas = (ArrayList<PessoaJuridica>) arquivoEntrada.readObject();
        arquivoEntrada.close();
        System.out.println("Dados de pessoas juridicas recuperados.");
    }
}
```

## main class CadastroPOO


```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */

package model;
import java.io.IOException;

/**
 *
 * @author Guilherme
 */
public class CadastroPOO {
    public static void main(String[] args) {

        PessoaFisicaRepo repo1 = new PessoaFisicaRepo();
        PessoaFisica pessoaFisica1 = new PessoaFisica(1, "Ana", "11111111111", 25);
        PessoaFisica pessoaFisica2 = new PessoaFisica(2, "Carlos Jose", "22222222222", 52);
        repo1.inserir(pessoaFisica1);
        repo1.inserir(pessoaFisica2);
        try {
            repo1.persistir("listaPessoasFisicas.bin");
        } catch (IOException erro) {
            System.out.println("Erro ao persistir os dados: " + erro.getMessage());
        }

        PessoaFisicaRepo repo2 = new PessoaFisicaRepo();

        try {
            repo2.recuperar("listaPessoasFisicas.bin");
            repo2.obterTodos()
                    .forEach(pessoaFisica -> {
                        pessoaFisica.exibir();
                    });

        } catch (IOException | ClassNotFoundException erro) {
            System.out.println("Erro ao recuperar os dados: " + erro.getMessage());
        }


        PessoaJuridicaRepo repo3 = new PessoaJuridicaRepo();
        PessoaJuridica pessoaJuridica1 = new PessoaJuridica(3, "XPTO Sales", "33333333333333");
        PessoaJuridica pessoaJuridica2 = new PessoaJuridica(4, "XPTO Solutions", "44444444444444");
        repo3.inserir(pessoaJuridica1);
        repo3.inserir(pessoaJuridica2);
        try {
            repo3.persistir("listaPessoasJuridicas.bin");
        } catch (IOException erro) {
            System.out.println("Erro ao persistir os dados: " + erro.getMessage());
        }

        PessoaJuridicaRepo repo4 = new PessoaJuridicaRepo();
        try {
            repo4.recuperar("listaPessoasJuridicas.bin");
            repo4.obterTodos()
                    .forEach(pessoaJuridica -> {
                        pessoaJuridica.exibir();
                    });
        } catch (IOException | ClassNotFoundException erro) {
            System.out.println("Erro ao recuperar os dados: " + erro.getMessage());
        }
    }
}
```

## Resultado




## Análise e Conclusão:

### Quais as vantagens e desvantagens do uso de herança?

**Vantagens**

**Reutilização de Código**, pois evita duplicação de código;

**Facilidade de Manutenção**, é fácil as alterações na superclasse que são refletidas automaticamente nas subclasses;

**Organização Estrutural** que proporciona uma estrutura hierárquica lógica, tornando o código mais compreensível e modular.

**Polimorfismo** que permite que objetos da subclasse sejam tratados como objetos da superclasse, facilitando o uso de conceitos como sobreposição de métodos e métodos abstratos;

**Classes podem herdar características**(métodos e atributos) de outras classes situadas acima ou transmitir suas características às classes abaixo. Caso uma alteração seja necessária, ela só precisará ser feita na classe pai, e será automaticamente propagada para as subclasses.

**Desvantagens**

É fraco o encapsulamento entre classes e subclasses, onde ao mudar uma superclasse pode afetar todas as subclasses;

Quando um objeto precisa ser de uma classe diferente em momentos diferentes e não é possível com a herança.

Estácio

**Análise e Conclusão:**

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

Essa interface permite que os objetos sejam serializados(convertidos em uma sequência de bytes) e desserializados com a conversão de volta à um objeto. Esse processo é essencial para armazenar objetos em arquivos, transmiti-los via rede ou gravá-los em banco de dados de forma eficiente.

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

O paradigma funcional é amplamente utilizado na API Stream do Java que é usada para manipular coleções (Collections) de uma maneira mais eficiente, utilizando funções. Ela possibilita uma interação sobre essas coleções de objetos e, a cada elemento, realizar alguma ação, seja ela de filtragem, mapeamento, transformação, etc.

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

Na persistência de dados em Java, o padrão de desenvolvimento mais adotado é o Data Access Object (DAO) que é o padrão que separa a lógica de acesso a dados da lógica de negócios, tornando o código mais organizado e fácil de mante-lo. Nesse nosso projeto foram utilizadas a classe ObjectOutputStream para escrever objetos em um arquivo "[prefixo].fisica.bin e [prefixo].juridica.bin" e a classe ObjectInputStream para ler os objetos dos mesmos arquivos.

Estácio

## main class CadastroPOO2

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */


package model;
import java.io.IOException;
import java.util.Scanner;


/**
 *
 * @author Guilherme
 */
public class CadastroPOO2 {

    public static void main(String[] args) {

        PessoaFisicaRepo pfRepo = new PessoaFisicaRepo();
        PessoaJuridicaRepo pjRepo = new PessoaJuridicaRepo();

        Scanner scan = new Scanner(System.in);
        String escolha;

        do {
            System.out.println("==============================");
            System.out.println("1 - Incluir Pessoa");
            System.out.println("2 - Alterar Pessoa");
            System.out.println("3 - Excluir Pessoa");
            System.out.println("4 - Buscar pelo Id");
            System.out.println("5 - Exibir Todos");
            System.out.println("6 - Persistir/Salvar Dados");
            System.out.println("7 - Recuperar/Carregar Dados");
            System.out.println("0 - Finalizar Programa");
            System.out.println("==============================");

            escolha = scan.next();

            switch (escolha) {

                // Incluir
                case "1":
                    do {
                        System.out.println("==============================");
                        System.out.println("F - Pessoa Fisica | J - Pessoa Juridica | M - Menu");

                        escolha = scan.next();
                        scan.nextLine();

                        switch (escolha.toUpperCase()) {

                            case "F":
                                System.out.print("Digite o id da pessoa: ");
                                int idInformado = scan.nextInt();
                                System.out.println("Insira os dados... ");
                                scan.nextLine();
                                System.out.print("Nome: ");
                                String nome = scan.nextLine();
                                System.out.print("CPF: ");
                                String cpf = scan.nextLine();
                                System.out.print("Idade: ");
                                int idade = scan.nextInt();

                                int pfRepoSize = pfRepo.obterTodos().size();

                                PessoaFisica pessoaFisica = new PessoaFisica(idInformado, nome, cpf, idade);
                                pfRepo.inserir(pessoaFisica);

                                System.out.println("Inclusao realizada com sucesso!");
                                pessoaFisica.exibir();
                                break;

                            case "J":
                                System.out.print("Digite o id da pessoa: ");
                                int idInformado2 = scan.nextInt();
                                scan.nextLine();
                                System.out.print("Nome: ");
                                nome = scan.nextLine();
                                System.out.print("CNPJ: ");
                                String cnpj = scan.nextLine();
```

Estácio

## main class CadastroPOO2

```
                        int pjRepoSize = pjRepo.obterTodos().size();

                        PessoaJuridica pessoaJuridica = new PessoaJuridica(idInformado2, nome, cnpj);
                        pjRepo.inserir(pessoaJuridica);

                        System.out.println("Inclusao realizada com sucesso!");
                        pessoaJuridica.exibir();
                        break;

                    case "M":
                        break;

                    default:
                        System.out.println("Opcao invalida.");
                        break;
                }
            } while (!escolha.equalsIgnoreCase("M"));
            break;

        // Alterar
        case "2":
            do {
                System.out.println("============================");
                System.out.println("F - Pessoa Fisica | J - Pessoa Juridica | M - Menu");

                escolha = scan.next();
                scan.nextLine();

                switch (escolha.toUpperCase()) {

                    case "F":
                        System.out.println("Digite o ID da pessoa: ");
                        int idPessoaFisica = scan.nextInt();
                        scan.nextLine();

                        PessoaFisica pessoaFisicaLocalizada = pfRepo.obter(idPessoaFisica);

                        if (pessoaFisicaLocalizada != null) {
                            pessoaFisicaLocalizada.exibir();

                            System.out.println("Nome atual: " + pessoaFisicaLocalizada.getNome());
                            System.out.print("Novo nome: ");
                            String novoNome = scan.nextLine();

                            System.out.println("CPF atual: " + pessoaFisicaLocalizada.getCpf());
                            System.out.print("Novo CPF: ");
                            String novoCPF = scan.nextLine();

                            System.out.println("Idade atual: " + pessoaFisicaLocalizada.getCpf());
                            System.out.print("Nova Idade: ");
                            int novaIdade = scan.nextInt();

                            pfRepo.alterar(pessoaFisicaLocalizada, novoNome, novoCPF, novaIdade);

                            System.out.println("Pessoa alterada com sucesso!");
                        } else
                            System.out.println("Pessoa nao localizada! ");
                        break;

                    case "J":
                        System.out.println("Digite o ID da pessoa: ");
                        int idPessoaJuridica = scan.nextInt();
                        scan.nextLine();

                        PessoaJuridica pessoaJuridicaLocalizada = pjRepo.obter(idPessoaJuridica);

                        if (pessoaJuridicaLocalizada != null) {
                            pessoaJuridicaLocalizada.exibir();

                            System.out.println("Nome atual: " + pessoaJuridicaLocalizada.getNome());
                            System.out.println("Novo nome: ");
                            String novoNome = scan.nextLine();

                            System.out.println("CNPJ atual: " + pessoaJuridicaLocalizada.getCnpj());
                            System.out.println("Novo CNPJ: ");
                            String novoCNPJ = scan.nextLine();

                            pjRepo.alterar(pessoaJuridicaLocalizada, novoNome, novoCNPJ);
```

Estácio

## main class CadastroPOO2

```
                                System.out.println("Pessoa alterada com sucesso!");
                            } else
                                System.out.println("Pessoa nao localizada!");
                            break;

                        case "M":
                            break;

                        default:
                            System.out.println("Opcao invalida.");
                            break;
                    }
                } while (!escolha.equalsIgnoreCase("M"));
                break;

            // EXCLUIR
            case "3":
                do {
                    System.out.println("============================");
                    System.out.println("F - Pessoa Fisica | J - Pessoa Juridica | M - Menu");

                    escolha = scan.next();
                    scan.nextLine();

                    switch (escolha.toUpperCase()) {

                        case "F":
                            System.out.println("Digite o ID da pessoa: ");
                            int idPessoaFisica = scan.nextInt();

                            PessoaFisica pessoaFisicaLocalizada = pfRepo.obter(idPessoaFisica);

                            if (pessoaFisicaLocalizada != null) {
                                pessoaFisicaLocalizada.exibir();
                                pfRepo.excluir(idPessoaFisica);

                                System.out.println("Pessoa excluida com sucesso!");
                            } else
                                System.out.println("Pessoa nao localizada!");
                            break;


                        case "J":
                            System.out.println("Digite o ID da pessoa: ");
                            int idPessoaJuridica = scan.nextInt();

                            PessoaJuridica pessoaJuridicaLocalizada = pjRepo.obter(idPessoaJuridica);

                            if (pessoaJuridicaLocalizada != null) {
                                pessoaJuridicaLocalizada.exibir();

                                pjRepo.excluir(idPessoaJuridica);

                                System.out.println("Pessoa excluida com sucesso!");
                            } else
                                System.out.println("Pessoa nao localizada!");
                            break;

                        case "M":
                            break;

                        default:
                            System.out.println("Opcao invalida.");
                            break;
                    }

                } while (!escolha.equalsIgnoreCase("M"));
                break;

            // obterId
            case "4":
                do {
                    System.out.println("============================");
                    System.out.println("F - Pessoa Fisica | J - Pessoa Juridica | M - Menu");

                    escolha = scan.next();
                    scan.nextLine();

                    switch (escolha.toUpperCase()) {

```

## main class CadastroPOO2

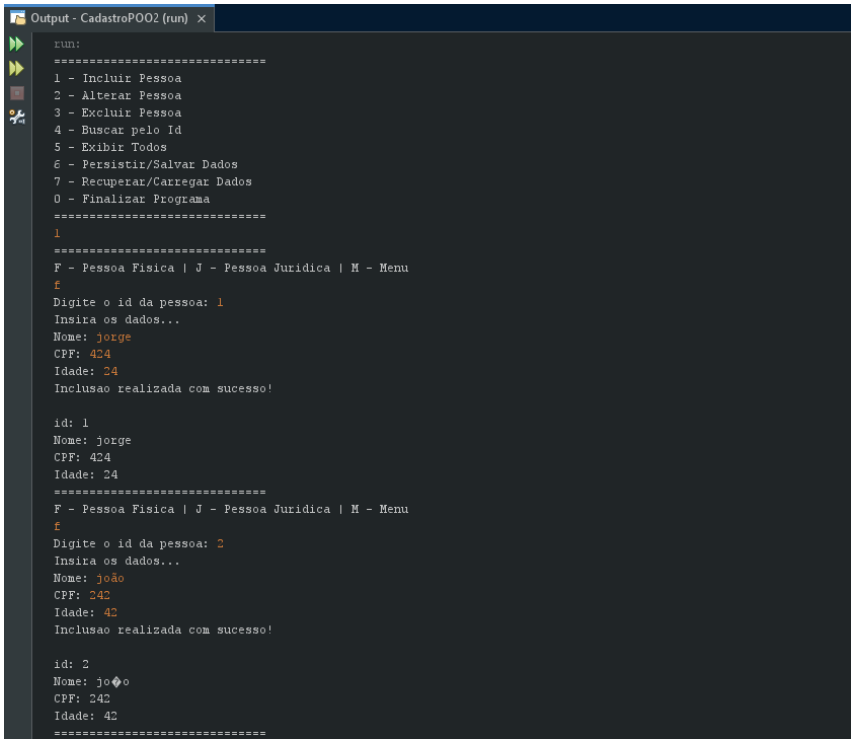
```
                            case "F":
                                System.out.println("Digite o ID da pessoa: ");
                                int idPessoaFisica = scan.nextInt();

                                PessoaFisica pessoaFisicaLocalizada = pfRepo.obter(idPessoaFisica);

                                if (pessoaFisicaLocalizada != null) {
                                    System.out.println("Pessoa localizada!");
                                    pessoaFisicaLocalizada.exibir();
                                } else
                                    System.out.println("Pessoa nao localizada!");
                                break;

                            case "J":
                                System.out.println("Digite o ID da pessoa: ");
                                int idPessoaJuridica = scan.nextInt();

                                PessoaJuridica pessoaJuridicaLocalizada = pjRepo.obter(idPessoaJuridica);

                                if (pessoaJuridicaLocalizada != null) {
                                    System.out.println("Pessoa localizada!");

                                    pessoaJuridicaLocalizada.exibir();
                                } else
                                    System.out.println("Pessoa nao localizada!");
                                break;

                            case "M":
                                break;

                            default:
                                System.out.println("Opcao invalida.");
                                break;
                        }

                    } while (!escolha.equalsIgnoreCase("M"));
                    break;

                //obterTodos
                case "5":
                    do {
                        System.out.println("============================");
                        System.out.println("F - Pessoa Fisica | J - Pessoa Juridica | M - Menu");

                        escolha = scan.next();
                        scan.nextLine();

                        switch (escolha.toUpperCase()) {

                            case "F":
                                System.out.println("Lista de pessoas Fisicas: ");
                                pfRepo.obterTodos()
                                        .forEach(pessoaFisica -> {
                                            pessoaFisica.exibir();
                                            System.out.println();
                                        });
                                break;

                            case "J":
                                System.out.println("Lista de pessoas juridicas: ");
                                pjRepo.obterTodos()
                                        .forEach(pessoaJuridica -> {
                                            pessoaJuridica.exibir();
                                            System.out.println();
                                        });
                                break;

                            case "M":
                                break;

                            default:
                                System.out.println("Opcao invalida");
                                break;
                        }

                    } while (!escolha.equalsIgnoreCase("M"));
                    break;

```

Estácio

## main class CadastroPOO2

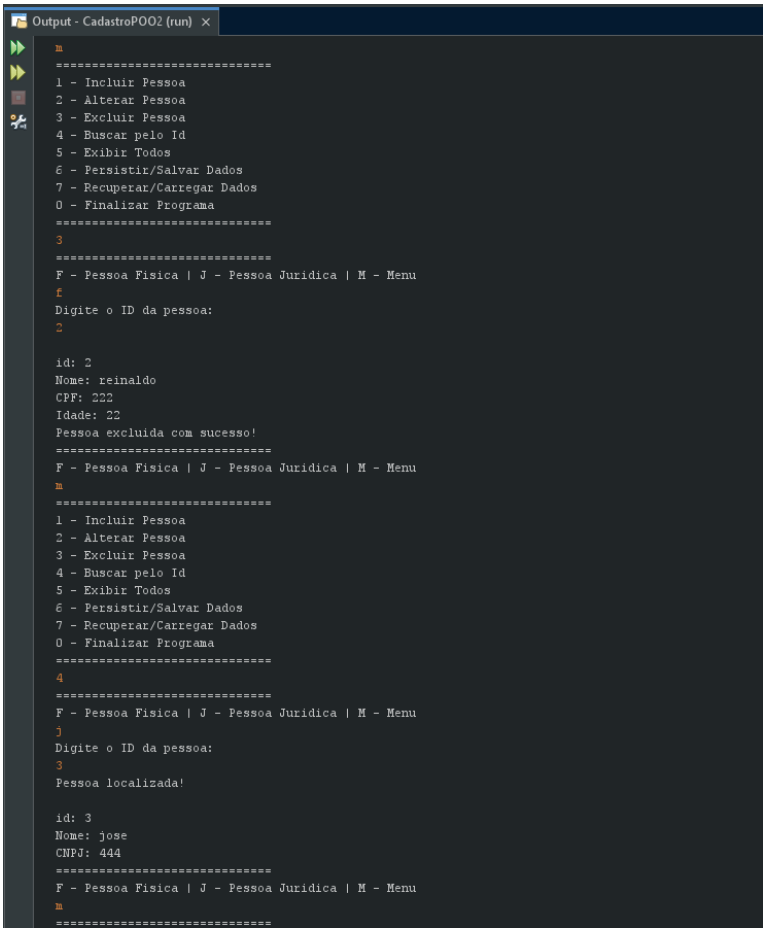
```
                    // Persistir/Salvar
                    case "6":
                        System.out.println("Escolha o nome do arquivo");
                        escolha = scan.next();
                        scan.nextLine();
                        try {
                            pfRepo.persistir(escolha+".fisica.bin");
                            pjRepo.persistir(escolha+".juridica.bin");
                        } catch (IOException erro) {
                            System.out.println("Erro ao persistir/salvar os dados: " + erro.getMessage());
                        }
                        break;

                    //Recuperar/Carregar
                    case "7":
                        System.out.println("Informe o nome do arquivo salvo");
                        escolha = scan.next();
                        scan.nextLine();
                        try {
                            pfRepo.recuperar(escolha+".fisica.bin");
                            pjRepo.recuperar(escolha+".juridica.bin");
                        } catch (ClassNotFoundException | IOException erro) {
                            System.out.println("Erro ao recuperar os dados: " + erro.getMessage());
                        }
                        break;

                    case "0":
                        System.out.println("Sistema Finalizado com sucesso.");
                        break;

                    default:
                        System.out.println("Opcao invalida");
                        break;
                }
            } while (!escolha.equals("0"));
            scan.close();
        }
    }

```

## Resultado


Output - CadastroPOO2 (run)
run:
==========
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir/Salvar Dados
7 - Recuperar/Carregar Dados
0 - Finalizar Programa
==========
1
==========
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
f
Digite o id da pessoa: 1
Insira os dados...
Nome: jorge
CPF: 424
Idade: 24
Inclusao realizada com sucesso!
id: 1
Nome: jorge
CPF: 424
Idade: 24
==========
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
f
Digite o id da pessoa: 2
Insira os dados...
Nome: joão
CPF: 242
Idade: 42
Inclusao realizada com sucesso!
id: 2
Nome: jo?o
CPF: 242
Idade: 42
==========


Output - CadastroPOO2 (run)
Idade: 42
==========
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
j
Digite o id da pessoa: 3
Nome: jose
CNPJ: 444
Inclusao realizada com sucesso!
id: 3
Nome: jose
CNPJ: 444
==========
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
m
==========
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir/Salvar Dados
7 - Recuperar/Carregar Dados
0 - Finalizar Programa
==========
2
==========
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
f
Digite o ID da pessoa:
2
id: 2
Nome: jo?o
CPF: 242
Idade: 42
Nome atual: jo?o
Novo nome: reinaldo
CPF atual: 242
Novo CPF: 222
Idade atual: 242
Nova Idade: 22
Pessoa alterada com sucesso!
==========
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
m
==========

## Resultado


Output - CadastroPOO2 (run)
m
==========
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir/Salvar Dados
7 - Recuperar/Carregar Dados
0 - Finalizar Programa
==========
3
==========
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
f
Digite o ID da pessoa:
2
id: 2
Nome: reinaldo
CPF: 222
Idade: 22
Pessoa excluida com sucesso!
==========
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
m
==========
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir/Salvar Dados
7 - Recuperar/Carregar Dados
0 - Finalizar Programa
==========
4
==========
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
j
Digite o ID da pessoa:
3
Pessoa localizada!
id: 3
Nome: jose
CNPJ: 444
==========
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
m
==========


Output - CadastroPOO2 (run)
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
m
==========
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir/Salvar Dados
7 - Recuperar/Carregar Dados
0 - Finalizar Programa
==========
5
==========
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
f
Lista de pessoas Fisicas:
id: 1
Nome: jorge
CPF: 424
Idade: 24
==========
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
j
Lista de pessoas juridicas:
id: 3
Nome: jose
CNPJ: 444
==========
F - Pessoa Fisica | J - Pessoa Juridica | M - Menu
m
==========
1 - Incluir Pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo Id
5 - Exibir Todos
6 - Persistir/Salvar Dados
7 - Recuperar/Carregar Dados
0 - Finalizar Programa
==========
7
Informe o nome do arquivo salvo
jose
Erro ao recuperar os dados: jose.fisica.bin (O sistema n?o pode encontrar o arquivo especificado)
==========

## Resultado


## Análise e Conclusão:

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

Os elementos estáticos são elementos que "existem", ou seja, estão disponíveis para uso, sem a necessidade de serem instânciados. Podem ser utilizados em código sem a necessidade de existirem objetos produzidos (sem a necessidade de um comando "new Classe()").

### Para que serve a classe Scanner?

A classe Scanner em Java é usada para ler entradas do usuário ou de outras fontes de dados, como arquivos e fluxos de entrada inseridos pelo ususario atraves do teclado.

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

As classes de repositório serviram para organizar as atividades de excluir, inserir, alterar, localizar e recuperar e também serviram para salvar os dados de pessoas físicas e jurídicas.