



Campus Aparecida de Goiânia

Dev Full Stack

Iniciando o caminho pelo Java

Italo Augusto Turma: 9001 3º Semestre

Missão Prática | Nível 1 | Mundo 3

Objetivos 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.
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.

Codigos:

Pessoa.java

```
import java.io.Serializable;

public class Pessoa implements Serializable {

    protected String nome;

    protected int id;

    //construtor padrão
    public Pessoa() {}

    //construtor completo
    public Pessoa (int id, String nome) {

        this.id = id;

        this.nome = nome;

    }

    public String getNome() {

        return this.nome;

    }

    public int getId() {

        return this.id;

    }

    public void setNome (String nome) {

        this.nome = nome;

    }

    public void setId (int id) {

        this.id = id;

    }

}
```

```

    }

    public static <T> void exibir(T item) {
        System.out.println("O valor eh: "+item);
    }

    public void exibir() {
        System.out.println("ID: " + id);
        System.out.println("Nome: " + nome);
    }
}

```

PessoaFisica.java

```

import java.io.Serializable;

public class PessoaFisica extends Pessoa implements Serializable {
    protected String cpf;
    protected int idade;

    //construtor padrão
    public PessoaFisica() {
        super();
    }

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

```

```

// metodos getters
public String getCPF() {
    return this.cpf;
}

public int getIdade() {
    return this.idade;
}

// metodos setters
public void setCPF (String cpf) {
    this.cpf = cpf;
}

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

@Override
public void exibir() {
    super.exibir();
    System.out.println("CPF: "+this.cpf);
    System.out.println("Idade: "+this.idade);
}
}

```

PessoaJuridica.java

```
import java.io.Serializable;
```

```

public class PessoaJuridica extends Pessoa implements Serializable {

    protected String cnpj;

    // Construtor padrao
    public PessoaJuridica(){
        super();
    }

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

    public String getCNPJ() {
        return this.cnpj;
    }

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

    @Override
    public void exibir() {
        super.exibir();
        System.out.println("CNPJ: "+this.cnpj);
    }
}

```

PessoaFisicaRepo.java

```
import java.util.ArrayList;

import java.util.List;

import java.io.*;

public class PessoaFisicaRepo {

    private List<PessoaFisica> pessoas;

    public PessoaFisicaRepo() {

        this.pessoas = new ArrayList<>();

    }

    public void inserir (PessoaFisica pessoa) {

        pessoas.add(pessoa);

    }

    public void alterar (PessoaFisica pessoa) {

        for (int i = 0 ; i < pessoas.size() ; i++) {

            if (pessoas.get(i).getId() == pessoa.getId()) {

                pessoas.set(i, pessoa);

                return;

            }

        }

        System.out.println("Pessoa nao encontrada para alteracao!");

    }

    public void excluir (int id) {

        for (int i = 0 ; i < pessoas.size() ; i++) {

            if (pessoas.get(i).getId() == id) {

                pessoas.remove(i);

                return;

            }

        }

    }

}
```

```
    }  
    System.out.println("Pessoa nao encontrada para exclusao!");  
}
```

```
public PessoaFisica obter (int id) {  
    for (PessoaFisica pessoa : pessoas) {  
        if (pessoa.getId() == id) {  
            return pessoa;  
        }  
    }  
    System.out.println("Pessoa nao encontrada.");  
    return null;  
}
```

```
public List<PessoaFisica> obterTodos() {  
    return new ArrayList<>(pessoas);  
}
```

```
public void persistir (String nomeArquivo) throws IOException {  
    try (ObjectOutputStream outputStream = new ObjectOutputStream(new  
        FileOutputStream(nomeArquivo))) {  
        outputStream.writeObject(pessoas);  
        System.out.println("Dados persistidos com sucesso!");  
    } catch (IOException e) {  
        throw new IOException("Erro ao persistir os dados do disco: "+e.getMessage());  
    }  
}
```

```
@SuppressWarnings("unchecked")  
public void recuperar (String nomeArquivo) throws IOException,  
ClassNotFoundException {
```

```

        try (ObjectInputStream inputStream = new ObjectInputStream(new
FileInputStream(nomeArquivo))) {
            pessoas = (List<PessoaFisica>) inputStream.readObject();
            System.out.println("Dados recuperados com sucesso!");
        } catch (IOException | ClassNotFoundException e) {
            throw new IOException("Erro ao recuperar os dados do disco: "+e.getMessage());
        }
    }
}
}

```

PessoaJuridicaRepo.java

```

import java.util.ArrayList;
import java.util.List;
import java.io.*;

public class PessoaJuridicaRepo {
    private List<PessoaJuridica> pessoas;

    public PessoaJuridicaRepo() {
        this.pessoas = new ArrayList<>();
    }

    public void inserir (PessoaJuridica pessoa) {
        pessoas.add(pessoa);
    }

    public void alterar (PessoaJuridica pessoa) {
        for (int i = 0 ; i < pessoas.size() ; i++) {
            if (pessoas.get(i).getId() == pessoa.getId()) {
                pessoas.set(i, pessoa);
            }
        }
    }
}

```



```

        return;
    }
}

System.out.println("Pessoa nao encontrada para a alteracao!");
}

```

```

public void excluir (int id) {
    for (int i = 0 ; i < pessoas.size() ; i++) {
        if (pessoas.get(i).getId() == id) {
            pessoas.remove(i);
            return;
        }
    }

    System.out.println("Pessoa nao encontrada para exclusao!");
}

```

```

public PessoaJuridica obter (int id) {
    for (PessoaJuridica pessoa : pessoas) {
        if (pessoa.getId() == id) {
            return pessoa;
        }
    }

    System.out.println("Pessoa nao encontrada.");
    return null;
}

```

```

public List<PessoaJuridica> obterTodos() {
    return new ArrayList<>(pessoas);
}

```

```

public void persistir (String nomeArquivo) throws IOException {

```

```

        try (ObjectOutputStream outputStream = new ObjectOutputStream(new
FileOutputStream(nomeArquivo))) {

            outputStream.writeObject(pessoas);

            System.out.println("Dados persistidos com sucesso!");

        } catch (IOException e) {

            throw new IOException("Erro ao persistir os dados do disco: "+e.getMessage());

        }

    }
}

```

```

@SuppressWarnings("unchecked")

public void recuperar (String nomeArquivo) throws IOException,
ClassNotFoundException {

    try (ObjectInputStream inputStream = new ObjectInputStream(new
FileInputStream(nomeArquivo))) {

        pessoas = (List<PessoaJuridica>) inputStream.readObject();

        System.out.println("Dados recuperados com sucesso!");

    } catch (IOException | ClassNotFoundException e) {

        throw new IOException("Erro ao recuperar os dados do disco: "+e.getMessage());

    }

}

}
}

```

Main.java

```

import java.io.*;

import java.util.Scanner;

import java.util.List;

import java.util.InputMismatchException;

public class Main {

    private static Scanner entrada;

```

```

private static PessoaFisicaRepo fisicaRepo = new PessoaFisicaRepo();
private static PessoaJuridicaRepo juridicaRepo = new PessoaJuridicaRepo();
private static int acao;

public static void main(String args[]) {
    entrada = new Scanner(System.in);
    do {
        try {
            System.out.println("Para continuar digite:");
            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 Dados");
            System.out.println("7 - Recuperar Dados");
            System.out.println("0 - Finalizar Programa");
            System.out.println("=====");

            acao = entrada.nextInt();
            entrada.nextLine();

            //incluir
            if (acao == 1) {
                System.out.println("Digite F para incluir uma Pessoa Fisica");
                System.out.println("ou");
                System.out.println("Digite J para incluir uma Pessoa Juridica");
                String tipoPessoa = entrada.next();
                entrada.nextLine();
            }
        } catch (Exception e) {
            System.out.println("Erro: " + e.getMessage());
        }
    } while (acao != 0);
}

```

```
if (tipoPessoa.equalsIgnoreCase("F") || tipoPessoa.equalsIgnoreCase("f")) {  
    System.out.println("Insira uma ID: ");  
    int id;  
    boolean idExistente;  
    do {  
        idExistente = false;  
        id = entrada.nextInt();  
        entrada.nextLine();  
  
        List<PessoaFisica> pessoas = fisicaRepo.obterTodos();  
        for (PessoaFisica pessoa : pessoas) {  
            if (pessoa.getId() == id) {  
                System.out.println("Essa ID ja existe!");  
                System.out.println("Por favor insira uma nova ID: ");  
                idExistente = true;  
                break;  
            }  
        }  
    } while (idExistente);  
  
    System.out.println("ID aceitavel.");  
    System.out.println("Insira o Nome: ");  
    String nome = entrada.nextLine();  
  
    System.out.println("Insira o CPF: ");  
    String cpf = entrada.nextLine();  
  
    System.out.println("Insira a Idade: ");  
    int idade = entrada.nextInt();  
    entrada.nextLine();
```

```

PessoaFisica novaPessoa = new PessoaFisica(id, nome, cpf, idade);

fisicaRepo.inserir(novaPessoa);

System.out.println("Pessoa fisica inserida com sucesso!");
} else if (tipoPessoa.equalsIgnoreCase("J") || tipoPessoa.equalsIgnoreCase("j")) {

    System.out.println("Insira uma ID: ");

    int id;

    boolean idExistente;

    do {

        idExistente = false;

        id = entrada.nextInt();

        entrada.nextLine();

        List<PessoaJuridica> pessoas = juridicaRepo.obterTodos();

        for (PessoaJuridica pessoa : pessoas) {

            if (pessoa.getId() == id) {

                System.out.println("Essa ID ja existe!");

                System.out.println("Por favor insira uma nova ID: ");

                idExistente = true;

                break;

            }

        }

    } while (idExistente);

    System.out.println("ID aceitavel.");

    System.out.println("Insira o nome da empresa: ");

    String nome = entrada.nextLine();

    System.out.println("Insira o CNPJ: ");

    String cnpj = entrada.nextLine();

    PessoaJuridica novaPessoa = new PessoaJuridica(id, nome, cnpj);

```

```

        juridicaRepo.inserir(novaPessoa);

        System.out.println("Pessoa juridica inserida com sucesso!");
    } else {
        System.out.println("Erro: valor diferente do solicitado.");
        System.out.println("Por favor insira somente F ou J.");
    }
}

//alterar
if (acao == 2) {
    System.out.println("Digite F para alterar uma Pessoa Fisica");
    System.out.println("ou");
    System.out.println("Digite J para alterar uma Pessoa Juridica");
    String tipoPessoa = entrada.nextLine();

    if (tipoPessoa.equalsIgnoreCase("F") || tipoPessoa.equalsIgnoreCase("f")) {
        System.out.println("Digite a ID da pessoa que deseja alterar: ");
        int id = entrada.nextInt();
        entrada.nextLine();

        List<PessoaFisica> pessoas = fisicaRepo.obterTodos();
        for (PessoaFisica pessoa : pessoas) {
            if (pessoa.getId() == id) {
                System.out.println("Pessoa Fisica encontrada.");
                System.out.println("Estes sao os dados atuais: ");
                pessoa.exibir();

                System.out.println("-----");

                System.out.println("Agora insira as alteracoes.");
                System.out.println("Insira o nome: ");
            }
        }
    }
}

```

```

String nome = entrada.nextLine();

System.out.println("Insira o CPF: ");
String cpf = entrada.nextLine();

System.out.println("Insira a idade: ");
int idade = entrada.nextInt();
entrada.nextLine();

PessoaFisica pessoaAlterada = new PessoaFisica(id, nome, cpf, idade);
fisicaRepo.alterar(pessoaAlterada);
System.out.println("Alteracao realizada com sucesso.");
} else {
    System.out.println("Essa ID não existe.");
}
}
} else if (tipoPessoa.equalsIgnoreCase("J") || tipoPessoa.equalsIgnoreCase("j")) {
    System.out.println("Digite a ID da pessoa que deseja alterar");
    int id = entrada.nextInt();
    entrada.nextLine();

    List<PessoaJuridica> pessoas = juridicaRepo.obterTodos();
    for (PessoaJuridica pessoa : pessoas) {
        if (pessoa.getId() == id) {
            System.out.println("Pessoa juridica encontrada.");
            System.out.println("Estes sao os dados atuais");
            pessoa.exibir();

            System.out.println("-----");

            System.out.println("Agora insira as alteracoes");

```

```

        System.out.println("Insira o nome da empresa: ");
        String nome = entrada.nextLine();

        System.out.println("Insira o CNPJ: ");
        String cnpj = entrada.nextLine();

        PessoaJuridica pessoaAlterada = new PessoaJuridica(id, nome, cnpj);
        juridicaRepo.alterar(pessoaAlterada);
        System.out.println("Pessoa alterada com sucesso.");
    } else {
        System.out.println("Essa ID não existe.");
    }
}
} else {
    System.out.println("Erro: Valor inserido diferente do solicitado.");
    System.out.println("Por favor digite somente F ou J.");
}
}

//excluir
if (acao == 3) {
    System.out.println("Digite F para excluir uma Pessoa Fisica");
    System.out.println("ou");
    System.out.println("Digite J para excluir uma Pessoa Juridica");
    String tipoPessoa = entrada.nextLine();

    if (tipoPessoa.equalsIgnoreCase("F") || tipoPessoa.equalsIgnoreCase("f")) {
        System.out.println("Qual a ID da pessoa que deseja excluir?");
        int id = entrada.nextInt();
        entrada.nextLine();
    }
}

```



```

List<PessoaFisica> pessoas = fisicaRepo.obterTodos();
for (PessoaFisica pessoa : pessoas) {
    if (pessoa.getId() == id) {
        System.out.println("Pessoa Fisica encontrada");
        System.out.println("Esses sao os dados: ");
        pessoa.exibir();
        System.out.println("-----");
        System.out.println("Para excluir digite 'confirmar' dessa mesma forma.");
        System.out.println("Caso contrario, digite qualquer outra coisa.");
        String confirma = entrada.nextLine();

        if (confirma.equalsIgnoreCase("confirmar")) {
            fisicaRepo.excluir(id);
            System.out.println("Pessoa excluida com sucesso.");
        } else {
            System.out.println("Pessoa nao excluida.");
        }
    } else {
        System.out.println("Essa ID nao existe.");
    }
}

} else if (tipoPessoa.equalsIgnoreCase("J") || tipoPessoa.equalsIgnoreCase("j")) {
    System.out.println("Qual a ID da pessoa que deseja excluir");
    int id = entrada.nextInt();
    entrada.nextLine();

    List<PessoaJuridica> pessoas = juridicaRepo.obterTodos();
    for (PessoaJuridica pessoa : pessoas) {
        if (pessoa.getId() == id) {
            System.out.println("Pessoa Juridica encontrada");
            System.out.println("Esses sao os dados: ");

```

```

        pessoa.exibir();

        System.out.println("-----");

        System.out.println("Para excluir digite 'confirmar' dessa mesma forma.");

        System.out.println("Caso contrario, digite qualquer outra coisa");

        String confirma = entrada.nextLine();

        if (confirma.equalsIgnoreCase("confirmar")) {

            juridicaRepo.excluir(id);

            System.out.println("Pessoa excluida com sucesso.");

        } else {

            System.out.println("Pessoa nao excluida.");

        }

    } else {

        System.out.println("Essa ID nao existe.");

    }

}

} else {

    System.out.println("Erro: entrada diferente da solicitada.");

    System.out.println("Por favor insira somente F ou J.");

}

}

```

//Obter/Buscar

```

if (acao == 4) {

    System.out.println("Digite F para buscar uma Pessoa Fisica");

    System.out.println("ou");

    System.out.println("Digite J para buscar uma Pessoa Juridica");

    String tipoPessoa = entrada.nextLine();

    if (tipoPessoa.equalsIgnoreCase("F") || tipoPessoa.equalsIgnoreCase("f")) {

        System.out.println("Digite a ID da pessoa que deseja ver os dados.");
    }
}

```

```

int id = entrada.nextInt();

entrada.nextLine();


List<PessoaFisica> pessoas = fisicaRepo.obterTodos();
for (PessoaFisica pessoa : pessoas) {
    if (pessoa.getId() == id) {
        System.out.println("Pessoa Fisica encontrada.");
        System.out.println("Estes sao os dados:");
        pessoa.exibir();
        System.out.println("-----");
    } else {
        System.out.println("Essa ID nao existe");
    }
}

} else if (tipoPessoa.equalsIgnoreCase("J") || tipoPessoa.equalsIgnoreCase("j")) {
    System.out.println("Digite a ID da pessoa que deseja ver os dados.");
    int id = entrada.nextInt();
    entrada.nextLine();


    List<PessoaJuridica> pessoas = juridicaRepo.obterTodos();
    for (PessoaJuridica pessoa : pessoas) {
        if (pessoa.getId() == id) {
            System.out.println("Pessoa Juridica encontrada.");
            System.out.println("Estes sao os dados:");
            pessoa.exibir();
            System.out.println("-----");
        } else {
            System.out.println("Essa ID nao existe.");
        }
    }

} else {

```

```

        System.out.println("Erro: valor inserido diferente do solicitado.");

        System.out.println("Por favor digite somente F ou J.");
    }
}

//Obter Todos
if (acao == 5) {
    System.out.println("Digite F para visualizar todas as Pessoas Fisicas");
    System.out.println("ou");
    System.out.println("Digite J para visualizar todas as Pessoas Juridicas");
    String tipoPessoa = entrada.nextLine();

    if (tipoPessoa.equalsIgnoreCase("F") || tipoPessoa.equalsIgnoreCase("f")) {
        System.out.println("Aqui estao todos os dados de Pessoas Fisicas");
        System.out.println("-----");

        List<PessoaFisica> pessoas = fisicaRepo.obterTodos();
        for (PessoaFisica pessoa : pessoas) {
            pessoa.exibir();
            System.out.println("-----");
        }
    } else if (tipoPessoa.equalsIgnoreCase("J") || tipoPessoa.equalsIgnoreCase("j")) {
        System.out.println("Aqui entao todos os dados de Pessoas Juridicas");
        System.out.println("-----");

        List<PessoaJuridica> pessoas = juridicaRepo.obterTodos();
        for (PessoaJuridica pessoa : pessoas) {
            pessoa.exibir();
            System.out.println("-----");
        }
    } else {

```

```

        System.out.println("Erro: valor inserido diferente do solicitado.");

        System.out.println("Por favor insira somente F ou J.");
    }
}

//Salvar/Persistir
if (acao == 6) {
    System.out.println("Salvando os dados...");

    System.out.println("Qual sera o nome do arquivo?");

    //o leitor não difere letra maiuscula e minuscula
    String nomeArquivo = entrada.nextLine();

    try {
        fisicaRepo.persistir(nomeArquivo + ".fisica.bin");
        juridicaRepo.persistir(nomeArquivo + ".juridica.bin");

        System.out.println("Lembre-se do nome desse arquivo");

        System.out.println("voce pode precisar dele mais tarde.");
    } catch (IOException e) {
        System.out.println("Erro ao persistir os dados no disco: " + e.getMessage());
    }
}

//recuperar
if (acao == 7) {
    System.out.println("Recuperando os dados..");

    System.out.println("Qual o nome do arquivo?");

    String nomeArquivo = entrada.nextLine();

    try {
        fisicaRepo.recuperar(nomeArquivo + ".fisica.bin");
        juridicaRepo.recuperar(nomeArquivo + ".juridica.bin");
    }
}

```

```

        } catch (IOException | ClassNotFoundException e) {

            System.out.println("Erro ao recuperar os dados: " + e.getMessage());

        }

    }

    } catch (InputMismatchException e) {

        System.out.println("Erro: Entrada nao permitida: " + e.getMessage());

        break;

    }

    } while (acao != 0);

}

}

```

Resultado da execu  o:

Para continuar digite:

=====

- 1 - Incluir Pessoa
- 2 - Alterar Pessoa
- 3 - Excluir Pessoa
- 4 - Buscar pelo ID
- 5 - Exibir Todos
- 6 - Persistir Dados
- 7 - Recuperar Dados
- 0 - Finalizar Programa

=====

1

Digite F para incluir uma Pessoa Fisica

ou

Digite J para incluir uma Pessoa Juridica

f

Insira uma ID:

1

ID aceitavel.

Insira o Nome:

Italo Augusto

Insira o CPF:

123.456.789-00

Insira a Idade:

18

Pessoa fisica inserida com sucesso!

Para continuar digite:

=====

1 - Incluir Pessoa

2 - Alterar Pessoa

3 - Excluir Pessoa

4 - Buscar pelo ID

5 - Exibir Todos

6 - Persistir Dados

7 - Recuperar Dados

0 - Finalizar Programa

=====

1

Digite F para incluir uma Pessoa Fisica

ou

Digite J para incluir uma Pessoa Juridica

f

Insira uma ID:

1

Essa ID ja existe!

Por favor insira uma nova ID:

2

ID aceitavel.

Insira o Nome:

Nathalia

Insira o CPF:

987.654.321-00

Insira a Idade:

18

Pessoa fisica inserida com sucesso!

Para continuar digite:

=====

1 - Incluir Pessoa

2 - Alterar Pessoa

3 - Excluir Pessoa

4 - Buscar pelo ID

5 - Exibir Todos

6 - Persistir Dados

7 - Recuperar Dados

0 - Finalizar Programa

=====

1

Digite F para incluir uma Pessoa Fisica

ou

Digite J para incluir uma Pessoa Juridica

f

Insira uma ID:

3

ID aceitavel.

Insira o Nome:

jeronimo

Insira o CPF:

192.837.465-00

Insira a Idade:

92

Pessoa fisica inserida com sucesso!

Para continuar digite:

=====

1 - Incluir Pessoa

2 - Alterar Pessoa

3 - Excluir Pessoa

4 - Buscar pelo ID

5 - Exibir Todos

6 - Persistir Dados

7 - Recuperar Dados

0 - Finalizar Programa

=====

2

Digite F para alterar uma Pessoa Fisica

ou

Digite J para alterar uma Pessoa Juridica

f

Digite a ID da pessoa que deseja alterar:

3

Essa ID não existe.

Essa ID não existe.

Pessoa Fisica encontrada.

Estes sao os dados atuais:

ID: 3

Nome: jeronimo

CPF: 192.837.465-00

Idade: 92

Agora insira as alteracoes.

Insira o nome:

Jeronimo M

Insira o CPF:

111.222.333-00

Insira a idade:

29

Alteracao realizada com sucesso.

Para continuar digite:

=====

1 - Incluir Pessoa

2 - Alterar Pessoa

3 - Excluir Pessoa

4 - Buscar pelo ID

5 - Exibir Todos

6 - Persistir Dados

7 - Recuperar Dados

0 - Finalizar Programa

=====

6

Salvando os dados...

Qual sera o nome do arquivo?

teste

Dados persistidos com sucesso!

Dados persistidos com sucesso!

Lembre-se do nome desse arquivo

voce pode precisar dele mais tarde.

Para continuar digite:

=====

1 - Incluir Pessoa

2 - Alterar Pessoa

3 - Excluir Pessoa

4 - Buscar pelo ID

5 - Exibir Todos

6 - Persistir Dados

7 - Recuperar Dados

0 - Finalizar Programa

=====

3

Digite F para excluir uma Pessoa Fisica

ou

Digite J para excluir uma Pessoa Juridica

f

Qual a ID da pessoa que dejesa exluir?

1

Pessoa Fisica encontrada

Esses sao os dados:

ID: 1

Nome: Italo Augusto

CPF: 123.456.789-00

Idade: 18

Para excluir digite 'confirmar' dessa mesma forma.

Caso contrario, digite qualquer outra coisa.

confirmar

Pessoa excluida com sucesso.

Essa ID nao existe.

Essa ID nao existe.

Para continuar digite:

=====

1 - Incluir Pessoa

2 - Alterar Pessoa

- 3 - Excluir Pessoa
- 4 - Buscar pelo ID
- 5 - Exibir Todos
- 6 - Persistir Dados
- 7 - Recuperar Dados
- 0 - Finalizar Programa

=====

5

Digite F para visualizar todas as Pessoas Fisicas

ou

Digite J para visualizar todas as Pessoas Juridicas

f

Aqui estao todos os dados de Pessoas Fisicas

ID: 2

Nome: Nathalia

CPF: 987.654.321-00

Idade: 18

ID: 3

Nome: Jeronimo M

CPF: 111.222.333-00

Idade: 29

Para continuar digite:

=====

- 1 - Incluir Pessoa
- 2 - Alterar Pessoa
- 3 - Excluir Pessoa
- 4 - Buscar pelo ID
- 5 - Exibir Todos

6 - Persistir Dados

7 - Recuperar Dados

0 - Finalizar Programa

=====

7

Recuperando os dados...

Qual o nome do arquivo?

teste

Dados recuperados com sucesso!

Dados recuperados com sucesso!

Para continuar digite:

=====

1 - Incluir Pessoa

2 - Alterar Pessoa

3 - Excluir Pessoa

4 - Buscar pelo ID

5 - Exibir Todos

6 - Persistir Dados

7 - Recuperar Dados

0 - Finalizar Programa

=====

5

Digite F para visualizar todas as Pessoas Fisicas

ou

Digite J para visualizar todas as Pessoas Juridicas

f

Aqui estao todos os dados de Pessoas Fisicas

ID: 1

Nome: Italo Augusto

CPF: 123.456.789-00

Idade: 18

ID: 2

Nome: Nathalia

CPF: 987.654.321-00

Idade: 18

ID: 3

Nome: Jeronimo M

CPF: 111.222.333-00

Idade: 29

Para continuar digite:

=====

1 - Incluir Pessoa

2 - Alterar Pessoa

3 - Excluir Pessoa

4 - Buscar pelo ID

5 - Exibir Todos

6 - Persistir Dados

7 - Recuperar Dados

0 - Finalizar Programa

=====

0

Análise e Conclusão:

A – O que são elementos estáticos e qual o motivo do método **main** adotar esse modificador?

Elementos estáticos são aqueles que são usados em todas as instâncias da classe, como métodos e variáveis estáticas. O método **main** precisa ser estático para poder ser acessado sem nenhuma instância da classe, pois ele será invocado pelo próprio ambiente de execução.

B – Para que serve a classe Scanner?

Essa classe serve para avisar ao executor que esse elemento receberá um valor do usuário através do terminal, sendo responsável pela interação com o usuário.

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

A organização das classes em diferentes arquivos ajudou bastante, tanto na visualização quanto na correção de erros, além de tornar a herança muito mais interessante.