



POLO OLARIA - RIO DE JANEIRO - RJ

DESENVOLVIMENTO FULL STACK

Nível 1: Iniciando o Caminho Pelo Java | Turma 9001

Erik Bastos de Moraes

Missão Prática | Nível 1 | Mundo 3

2º Procedimento | Criação do Cadastro em Modo Texto

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.

main.java

```
package cadastropoo;

import model.*;

import java.util.Scanner;

public class main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        PessoaFisicaRepo repoPF = new PessoaFisicaRepo();

        PessoaJuridicaRepo repoPJ = new PessoaJuridicaRepo();

        int opcao;

        do {

            System.out.println("\n==== MENU PRINCIPAL ====");

            System.out.println("1 - Incluir");

            System.out.println("2 - Alterar");

            System.out.println("3 - Excluir");

            System.out.println("4 - Exibir por ID");

            System.out.println("5 - Exibir todos");

            System.out.println("6 - Salvar dados");

            System.out.println("7 - Recuperar dados");

            System.out.println("0 - Sair");

            System.out.print("Escolha uma opção: ");

            opcao = Integer.parseInt(scanner.nextLine());
```

```

switch (opcao) {

    case 1: // Incluir

        System.out.print("Tipo (F = Física, J = Jurídica): ");

        String tipo = scanner.nextLine();

        if (tipo.equalsIgnoreCase("F")) {

            System.out.print("ID: ");

            int id = Integer.parseInt(scanner.nextLine());

            System.out.print("Nome: ");

            String nome = scanner.nextLine();

            System.out.print("CPF: ");

            String cpf = scanner.nextLine();

            System.out.print("Idade: ");

            int idade = Integer.parseInt(scanner.nextLine());

            repoPF.inserir(new PessoaFisica(id, nome, cpf, idade));

        } else if (tipo.equalsIgnoreCase("J")) {

            System.out.print("ID: ");

            int id = Integer.parseInt(scanner.nextLine());

            System.out.print("Nome: ");

            String nome = scanner.nextLine();

            System.out.print("CNPJ: ");

            String cnpj = scanner.nextLine();

            repoPJ.inserir(new PessoaJuridica(id, nome, cnpj));

        }

        break;

    case 2: // Alterar

```

```

System.out.print("Tipo (F = Física, J = Jurídica): ");

tipo = scanner.nextLine();

System.out.print("ID: ");

int idAlt = Integer.parseInt(scanner.nextLine());

if (tipo.equalsIgnoreCase("F")) {

    PessoaFisica pf = repoPF.obter(idAlt);

    if (pf != null) {

        pf.exibir();

        System.out.print("Novo nome: ");

        pf.setNome(scanner.nextLine());

        System.out.print("Novo CPF: ");

        pf.setCpf(scanner.nextLine());

        System.out.print("Nova idade: ");

        pf.setIdade(Integer.parseInt(scanner.nextLine()));

        repoPF.alterar(pf);

    } else {

        System.out.println("Pessoa física não encontrada.");

    }

} else if (tipo.equalsIgnoreCase("J")) {

    PessoaJuridica pj = repoPJ.obter(idAlt);

    if (pj != null) {

        pj.exibir();

        System.out.print("Novo nome: ");

        pj.setNome(scanner.nextLine());

        System.out.print("Novo CNPJ: ");

        pj.setCnpj(scanner.nextLine());

```

```
        repoPJ.alterar(pj);  
  
    } else {  
  
        System.out.println("Pessoa jurídica não encontrada.");  
  
    }  
  
}  
  
break;
```

case 3: // Excluir

```
    System.out.print("Tipo (F = Física, J = Jurídica): ");  
  
    tipo = scanner.nextLine();  
  
    System.out.print("ID: ");  
  
    int idExc = Integer.parseInt(scanner.nextLine());  
  
    if (tipo.equalsIgnoreCase("F")) {  
  
        repoPF.excluir(idExc);  
  
    } else if (tipo.equalsIgnoreCase("J")) {  
  
        repoPJ.excluir(idExc);  
  
    }  
  
    break;
```

case 4: // Exibir por ID

```
    System.out.print("Tipo (F = Física, J = Jurídica): ");  
  
    tipo = scanner.nextLine();  
  
    System.out.print("ID: ");  
  
    int idExibir = Integer.parseInt(scanner.nextLine());  
  
    if (tipo.equalsIgnoreCase("F")) {  
  
        PessoaFisica pf = repoPF.obter(idExibir);  
  
        if (pf != null) pf.exibir();  
  
    }  
  
    break;
```

```

        else System.out.println("Pessoa física não encontrada.");
    } else if (tipo.equalsIgnoreCase("J")) {

        PessoaJuridica pj = repoPJ.obter(idExibir);

        if (pj != null) pj.exibir();

        else System.out.println("Pessoa jurídica não encontrada.");
    }

    break;

```

case 5: // Exibir todos

```

    System.out.print("Tipo (F = Física, J = Jurídica): ");

    tipo = scanner.nextLine();

    if (tipo.equalsIgnoreCase("F")) {

        for (PessoaFisica pf : repoPF.obterTodos()) {

            pf.exibir();

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

        }

    } else if (tipo.equalsIgnoreCase("J")) {

        for (PessoaJuridica pj : repoPJ.obterTodos()) {

            pj.exibir();

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

        }

    }

    break;

```

case 6: // Salvar dados

```

    System.out.print("Prefixo para salvar arquivos: ");

    String prefixoSalvar = scanner.nextLine();

```

```
try {  
    repoPF.persistir(prefixoSalvar + ".fisica.bin");  
    repoPJ.persistir(prefixoSalvar + ".juridica.bin");  
    System.out.println("Dados salvos com sucesso.");  
} catch (Exception e) {  
    System.out.println("Erro ao salvar: " + e.getMessage());  
}  
  
break;
```

case 7: // Recuperar dados

```
System.out.print("Prefixo para recuperar arquivos: ");  
String prefixoRec = scanner.nextLine();  
  
try {  
    repoPF.recuperar(prefixoRec + ".fisica.bin");  
    repoPJ.recuperar(prefixoRec + ".juridica.bin");  
    System.out.println("Dados recuperados com sucesso.");  
} catch (Exception e) {  
    System.out.println("Erro ao recuperar: " + e.getMessage());  
}  
  
break;
```

case 0:

```
System.out.println("Encerrando o sistema.");  
  
break;
```

default:

```
System.out.println("Opção inválida.");
```

```
break;
```

```
}
```

```
} while (opcao != 0);
```

```
scanner.close();
```

```
}
```

```
}
```


Resultado

run:

==== MENU PRINCIPAL ====

1 - Incluir

2 - Alterar

3 - Excluir

4 - Exibir por ID

5 - Exibir todos

6 - Salvar dados

7 - Recuperar dados

0 - Sair

Escolha uma opção: 1

Tipo (F = Física, J = Jurídica): F

ID: 1

Nome: Roland

CPF: 123.123.123-12

Idade: 30

==== MENU PRINCIPAL ====

1 - Incluir

2 - Alterar

3 - Excluir

4 - Exibir por ID

5 - Exibir todos

6 - Salvar dados

7 - Recuperar dados

0 - Sair

Escolha uma opção: 5

Tipo (F = Física, J = Jurídica): F

ID: 1, Nome: Roland

CPF: 123.123.123-12, Idade: 30

==== MENU PRINCIPAL ====

1 - Incluir

2 - Alterar

3 - Excluir

4 - Exibir por ID

5 - Exibir todos

6 - Salvar dados

7 - Recuperar dados

0 - Sair

Escolha uma opção: 0

Encerrando o sistema.

BUILD SUCCESSFUL (total time: 1 minute 5 seconds)

Análise e Conclusão

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

R: São elementos que pertencem a classe, e não a suas instâncias. Como o main é chamado pela JVM logo no início do programa, ele não pode depender de uma instância da classe, então precisa ser estático.

B. Para que serve a classe Scanner?

R: A classe scanner serve para receber dados de entrada através do teclado.

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

R: Isolando certas partes do código para mais fácil leitura.

<https://github.com/ErikBM2661/Cadastro.git>