



Universidade Estácio
Campus Vargem Grande Paulista
Curso: Desenvolvimento Full Stack
Disciplina: Iniciando o Caminho Pelo Java
Turma: 2024.3
3º semestre letivo
Marcia da Silva e Souza

Título da Prática: Iniciando o caminho pelo java
Implementação de um cadastro de clientes em modo texto, com persistência em arquivos, baseado na tecnologia Java.

Objetivo da prática:

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

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.



Códigos utilizados na atividade prática 1

2º Procedimento:

main.java:

```
package model;
```

```
import java.io.IOException;
```

```
import java.util.Scanner;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        try (Scanner scanner = new Scanner(System.in)) {
```

```
            PessoaFisicaRepo repoFisica = new
```

```
PessoaFisicaRepo();
```

```
            PessoaJuridicaRepo repoJuridica = new
```

```
PessoaJuridicaRepo();
```

```
        while (true) {
```

```
            System.out.println("Escolha uma opção:");
```

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

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

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

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

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

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

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

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



```
int opcao = scanner.nextInt();  
scanner.nextLine(); // Consumir a nova linha
```

```
if (opcao == 0) {  
    break;  
}
```

```
switch (opcao) {  
    case 1 -> incluir(scanner, repoFisica, repoJuridica);  
    case 2 -> alterar(scanner, repoFisica,  
repoJuridica);  
    case 3 -> excluir(scanner, repoFisica,  
repoJuridica);  
    case 4 -> exibirPorId(scanner, repoFisica,  
repoJuridica);  
    case 5 -> exibirTodos(scanner, repoFisica,  
repoJuridica);  
    case 6 -> salvarDados(scanner, repoFisica,  
repoJuridica);  
    case 7 -> recuperarDados(scanner, repoFisica,  
repoJuridica);  
    default -> System.out.println("Opção inválida.");  
}  
}  
}
```

```
private static void incluir(Scanner scanner,  
PessoaFisicaRepo repoFisica, PessoaJuridicaRepo  
repoJuridica) {  
    System.out.println("Escolha o tipo (1 - Física, 2 -  
Jurídica):");  
    int tipo = scanner.nextInt();
```



```
scanner.nextLine(); // Consumir a nova linha
switch (tipo) {
    case 1 -> {
        System.out.println("Digite o ID:");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consumir a nova linha
        System.out.println("Digite o nome:");
        String nome = scanner.nextLine();
        System.out.println("Digite o CPF:");
        String cpf = scanner.nextLine();
        System.out.println("Digite a idade:");
        int idade = scanner.nextInt();
        scanner.nextLine(); // Consumir a nova linha
        repoFisica.inserir(new PessoaFisica(id, nome, cpf,
idade));
    }
    case 2 -> {
        System.out.println("Digite o ID:");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consumir a nova linha
        System.out.println("Digite o nome:");
        String nome = scanner.nextLine();
        System.out.println("Digite o CNPJ:");
        String cnpj = scanner.nextLine();
        repoJuridica.inserir(new PessoaJuridica(id, nome,
cnpj));
    }
    default -> System.out.println("Tipo inválido.");
}
}
```



```
private static void alterar(Scanner scanner,
PessoaFisicaRepo repoFisica, PessoaJuridicaRepo
repoJuridica) {
    System.out.println("Escolha o tipo (1 - Física, 2 -
Jurídica):");
    int tipo = scanner.nextInt();
    scanner.nextLine(); // Consumir a nova linha

    if (tipo == 1) {
        System.out.println("Digite o ID:");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consumir a nova linha
        PessoaFisica pf = repoFisica.obter(id);
        if (pf != null) {
            System.out.println("Dados atuais: ");
            pf.exibir();
            System.out.println("Digite o novo nome:");
            String nome = scanner.nextLine();
            System.out.println("Digite o novo CPF:");
            String cpf = scanner.nextLine();
            System.out.println("Digite a nova idade:");
            int idade = scanner.nextInt();
            scanner.nextLine(); // Consumir a nova linha

            repoFisica.alterar(new PessoaFisica(id, nome, cpf,
idade));
        } else {
            System.out.println("Pessoa Física não encontrada.");
        }
    } else if (tipo == 2) {
        System.out.println("Digite o ID:");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consumir a nova linha
```



```
PessoaJuridica pj = repoJuridica.obter(id);
if (pj != null) {
    System.out.println("Dados atuais: ");
    pj.exibir();
    System.out.println("Digite o novo nome:");
    String nome = scanner.nextLine();
    System.out.println("Digite o novo CNPJ:");
    String cnpj = scanner.nextLine();

    repoJuridica.alterar(new PessoaJuridica(id, nome,
cnpj));
} else {
    System.out.println("Pessoa Jurídica não
encontrada.");
}
} else {
    System.out.println("Tipo inválido.");
}
}

private static void excluir(Scanner scanner,
PessoaFisicaRepo repoFisica, PessoaJuridicaRepo
repoJuridica) {
    System.out.println("Escolha o tipo (1 - Física, 2 -
Jurídica):");
    int tipo = scanner.nextInt();
    scanner.nextLine(); // Consumir a nova linha
    switch (tipo) {
        case 1:
            {
                System.out.println("Digite o ID:");
                int id = scanner.nextInt();
                scanner.nextLine(); // Consumir a nova linha
```



```
        repoFisica.excluir(id);
        break;
    }
    case 2:
    {
        System.out.println("Digite o ID:");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consumir a nova linha
        repoJuridica.excluir(id);
        break;
    }
    default:
        System.out.println("Tipo inválido.");
        break;
}
}
```

```
private static void exibirPorId(Scanner scanner,
PessoaFisicaRepo repoFisica, PessoaJuridicaRepo
repoJuridica) {
    System.out.println("Escolha o tipo (1 - Física, 2 -
Jurídica):");
    int tipo = scanner.nextInt();
    scanner.nextLine(); // Consumir a nova linha
    switch (tipo) {
        case 1 -> {
            System.out.println("Digite o ID:");
            int id = scanner.nextInt();
            scanner.nextLine(); // Consumir a nova linha
            PessoaFisica pf = repoFisica.obter(id);
            if (pf != null) {
                pf.exibir();
            } else {
```



```
        System.out.println("Pessoa Física não
encontrada.");
    }
}
case 2 -> {
    System.out.println("Digite o ID:");
    int id = scanner.nextInt();
    scanner.nextLine(); // Consumir a nova linha
    PessoaJuridica pj = repoJuridica.obter(id);
    if (pj != null) {
        pj.exibir();
    } else {
        System.out.println("Pessoa Jurídica não
encontrada.");
    }
}
default -> System.out.println("Tipo inválido.");
}
}

private static void exibirTodos(Scanner scanner,
PessoaFisicaRepo repoFisica, PessoaJuridicaRepo
repoJuridica) {
    System.out.println("Escolha o tipo (1 - Física, 2 -
Jurídica):");
    int tipo = scanner.nextInt();
    scanner.nextLine(); // Consumir a nova linha
    switch (tipo) {
        case 1 -> {
            for (PessoaFisica pf : repoFisica.obterTodos()) {
                pf.exibir();
            }
        }
        case 2 -> {
            for (PessoaJuridica pj : repoJuridica.obterTodos()) {
```




```
        pj.exibir();
    }
}
default -> System.out.println("Tipo inválido.");
}
}
```

```
private static void salvarDados(Scanner scanner,
PessoaFisicaRepo repoFisica, PessoaJuridicaRepo
repoJuridica) {
    System.out.println("Digite o prefixo dos arquivos:");
    String prefixo = scanner.nextLine();

    try {
        repoFisica.persistir(prefixo + ".fisica.bin");
        System.out.println("Dados de pessoas físicas
armazenados.");
    } catch (IOException e) {
        System.out.println("Erro ao salvar dados de pessoas
físicas.");
    }

    try {
        repoJuridica.persistir(prefixo + ".juridica.bin");
        System.out.println("Dados de pessoas jurídicas
armazenados.");
    } catch (IOException e) {
        System.out.println("Erro ao salvar dados de pessoas
jurídicas.");
    }
}
```



```
private static void recuperarDados(Scanner scanner,
PessoaFisicaRepo repoFisica, PessoaJuridicaRepo
repoJuridica) {
    System.out.println("Digite o prefixo dos arquivos:");
    String prefixo = scanner.nextLine();

    try {
        repoFisica.recuperar(prefixo + ".fisica.bin");
        System.out.println("Dados de pessoas físicas
recuperados.");
    } catch (IOException | ClassNotFoundException e) {
        System.out.println("Erro ao recuperar dados de
pessoas físicas.");
    }
}
```

Resultado da execução:

run:

Escolha uma opção:

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



Análise de conclusão:

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

Ele permite que o java invoque o main sem ter a necessidade de criar uma instancia da classe.

Para que serve a classe Scanner?

É utilizada para ler diferentes tipos de entradas, como dados de arquivos e teclado.

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

Diminuindo o acoplamento de classes, facilitando a implementação de testes de unidade e padronizando códigos e serviços.

link do programa CadastroPOO no github:

<https://github.com/mah2405/CadastroPOO.git>