



Campus:	Polo Jardim Goiás	Curso:	Desenvolvimento Full Stack
Disciplina:	RPG0014 - Iniciando o Caminho Pelo Java		
Semestre Letivo:	2024.3	Turma:	9001
Aluno:	Edson Luiz Pacheco Junior		
Repositório GIT:	EdsonJr73/cadastropoo (github.com)		

Implementação de um cadastro de clientes em modo texto, com persistência em arquivos, baseado na tecnologia Java.

1. Título da Prática: “2º Procedimento | Criação do Cadastro em Modo Texto
2. Objetivos da Prática:
 - a. Utilizar herança e polimorfismo na definição de entidades.
 - b. Utilizar persistência de objetos em arquivos binários.
 - c. Implementar uma interface cadastral em modo texto.
 - d. Utilizar o controle de exceções da plataforma Java.
 - e. Implementar um sistema cadastral em Java, utilizando os recursos da programação orientada a objetos e a persistência em arquivos binários.
3. Códigos solicitados nesta aula:

CadastroPOO.java (main):

```
package cadastropoo;
```

```
import model.*;  
import java.io.*;  
import java.util.Scanner;
```

```
/**  
*
```

* @author edson-202308892185

*/

public class CadastroPOO {

```
    private final static PessoaFisicaRepo repoFisica = new PessoaFisicaRepo();
    private final static PessoaJuridicaRepo repoJuridica = new PessoaJuridicaRepo();
    private final static Scanner scanner = new Scanner(System.in);
```

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

```
        int opcao;
```

```
        do {
```

```
            exibirMenu();
```

```
            opcao = scanner.nextInt();
```

```
            scanner.nextLine();
```

```
            processarOpcao(opcao);
```

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

```
        System.out.println("Programa finalizado.");
```

```
    }
```

```
    private static void exibirMenu() {
```

```
        System.out.println("=====");
```

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

```
        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("=====");
```

```
    }
```

```
    private static void processarOpcao(int opcao) {
```

```
        switch (opcao) {
```

```
            case 1:
```

```
                incluir();
```

```

        break;
    case 2:
        alterar();
        break;
    case 3:
        excluir();
        break;
    case 4:
        exibirPorId();
        break;
    case 5:
        exibirTodos();
        break;
    case 6:
        salvarDados();
        break;
    case 7:
        recuperarDados();
        break;
    case 0:
        break;
    default:
        System.out.println("Opção inválida.");
    }
}

```

```

private static void incluir() {
    System.out.println("F - Física | J - Jurídica");
    String tipo = scanner.nextLine();
    switch (tipo) {

        case "F":
        case "f":
            PessoaFisica pf = criarPessoaFisica();
            repoFisica.inserir(pf);
            break;
        case "J":
        case "j":

```

```

        PessoaJuridica pj = criarPessoaJuridica();
        repoJuridica.inserir(pj);
        break;
    default:
        System.out.println("Tipo invalido.");
        break;
    }
}

```

```

private static void alterar() {
    System.out.println("F - Fisica | J - Juridica");
    String tipo = scanner.nextLine();

    System.out.print("Digite o ID da pessoa: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    switch (tipo) {
        case "F":
        case "f":
            PessoaFisica pf = repoFisica.obter(id);
            if (pf != null) {
                System.out.println("Dados atuais: ");
                pf.exibir();
                System.out.println("Digite os novos dados:");
                PessoaFisica novaPf = criarPessoaFisica();
                novaPf.setId(id);
                repoFisica.alterar(novaPf);
            } else {
                System.out.println("Pessoa fisica não encontrada.");
            }
            break;
        case "J":
        case "j":
            PessoaJuridica pj = repoJuridica.obter(id);
            if (pj != null) {
                System.out.println("Dados atuais: ");
                pj.exibir();
                System.out.println("Digite os novos dados:");
            }
            break;
    }
}

```

```

        PessoaJuridica novaPj = criarPessoaJuridica();
        novaPj.setId(id);
        repoJuridica.alterar(novaPj);
    } else {
        System.out.println("Pessoa juridica não encontrada.");
    }
    break;
default:
    System.out.println("Tipo invalido.");
    break;
}
}

```

```

private static void excluir() {
    System.out.println("F - Fisica | J - Juridica");
    String tipo = scanner.nextLine();

    System.out.print("Digite o ID da pessoa: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    switch (tipo) {
        case "F":
        case "f":
            repoFisica.excluir(id);
            System.out.println("Pessoa fisica removida.");
            break;
        case "J":
        case "j":
            repoJuridica.excluir(id);
            System.out.println("Pessoa juridica removida.");
            break;
        default:
            System.out.println("Tipo invalido.");
            break;
    }
}
}

```

```

private static void exibirPorId() {

```

```

System.out.println("F - Fisica | J - Juridica");
String tipo = scanner.nextLine();

System.out.print("Digite o ID da pessoa: ");
int id = scanner.nextInt();
scanner.nextLine();
switch (tipo) {
    case "F":
    case "f":
        PessoaFisica pf = repoFisica.obter(id);
        if (pf != null) {
            pf.exibir();
        } else {
            System.out.println("Pessoa fisica não encontrada.");
        }
        break;
    case "J":
    case "j":
        PessoaJuridica pj = repoJuridica.obter(id);
        if (pj != null) {
            pj.exibir();
        } else {
            System.out.println("Pessoa juridica não encontrada.");
        }
        break;
    default:
        System.out.println("Tipo invalido.");
        break;
}
}

```

```

private static void exibirTodos() {
    System.out.println("F - Fisica | J - Juridica");
    String tipo = scanner.nextLine();

    switch (tipo) {
        case "F":
        case "f":

```

```

        for (PessoaFisica pf : repoFisica.obterTodos()) {
            pf.exibir();
        }
        break;
    case "J":
    case "j":
        for (PessoaJuridica pj : repoJuridica.obterTodos()) {
            pj.exibir();
        }
        break;
    default:
        System.out.println("Tipo invalido.");
        break;
    }
}

private static void salvarDados() {
    System.out.println("Digite o prefixo do arquivo");
    String prefixo = scanner.nextLine();

    try {
        repoFisica.persistir(prefixo + ".fisica.bin");
        repoJuridica.persistir(prefixo + ".juridica.bin");
        System.out.println("Dados salvos com sucesso.");
    } catch (IOException e) {
        System.out.println("Erro ao salvar dados: " + e.getMessage());
    }
}

private static void recuperarDados() {
    System.out.println("Digite o prefixo do arquivo");
    String prefixo = scanner.nextLine();

    try {
        repoFisica.recuperar(prefixo + ".fisica.bin");
        repoJuridica.recuperar(prefixo + ".juridica.bin");
        System.out.println("Dados recuperados com sucesso.");
    } catch (IOException | ClassNotFoundException e) {

```

```

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

```

```

private static PessoaFisica criarPessoaFisica() {
    System.out.print("Digite o nome: ");
    String nome = scanner.nextLine();
    System.out.print("Digite o CPF: ");
    String cpf = scanner.nextLine();
    System.out.print("Digite a idade: ");
    int idade = scanner.nextInt();
    scanner.nextLine();
    return new PessoaFisica(0, nome, cpf, idade);
}

```

```

private static PessoaJuridica criarPessoaJuridica() {
    System.out.print("Digite o nome: ");
    String nome = scanner.nextLine();
    System.out.print("Digite o CNPJ: ");
    String cnpj = scanner.nextLine();
    return new PessoaJuridica(0, nome, cnpj);
}
}

```

*Pequena modificação nas repo para incrementação de ID:

```
private int proximold = 1;
```

Adicionei este atributo em cada repo e no método inserir eu fiz isso:

```

public void inserir(PessoaFisica pessoaFisica) {

    pessoaFisica.setId(proximold);

    listaPessoaFisica.add(pessoaFisica);

    proximold++;

}

```

4. Resultado da Execução do Código: Estarão no final do relatório.

5. Análise e Conclusão:

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

São elementos que pertencem à classe em si e não a uma instância específica da classe, ou seja, ele poderá ser acessado sem a necessidade de criar um objeto da classe. A main o adota para que a JVM o chame sem precisar criar uma instância da classe, pois ele precisa ser executado para que o programa possa iniciar e como não há nenhum objeto instanciado ainda ele precisa ser estático.

- b. Para que serve a classe Scanner?

Para ler dados de entrada do usuário e arquivos, facilitando a entrada de dados no programa permitindo a leitura de dados inteiros, strings, floats, booleanos etc.

- c. Como o uso de classes de repositório impactou a organização do código? Possibilitou a reutilização de códigos em diferentes lugares do meu programa, facilita a manutenção pois separa as responsabilidades, então quando eu quiser alterar a lógica de como tal dado é tratado eu só preciso ir às classes gerenciadoras, ou caso eu queira adicionar novos dados eu irei nas classes modelos. Isso faz com que o meu código fique mais modular.

Execuções:

1. Incluir Pessoa:

```
=====
Escolha uma opcao:
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
F - Fisica | J - Juridica
f
Digite o nome: Edson
Digite o CPF: 123.456.789-10
Digite a idade: 23
```

2. Alterar Pessoa:

```
=====
Escolha uma opcao:
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
F - Fisica | J - Juridica
F
Digite o ID da pessoa: 1
Dados atuais:
ID: 1
Nome: Edson
CPF: 123.456.789-10
Idade: 23
Digite os novos dados:
Digite o nome: Edson Luiz
Digite o CPF: 123.456.789-11
Digite a idade: 24
```

3. Excluir Pessoa:

```
=====
Escolha uma opcao:
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
F - Fisica | J - Juridica
J
Digite o nome: Coffe Plus
Digite o CNPJ: 12.345.678/0001-09
=====
Escolha uma opcao:
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
F - Fisica | J - Juridica
J
Digite o ID da pessoa: 1
Pessoa juridica removida.
```

4. Buscar pelo ID:

```

=====
Escolha uma opcao:
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
=====
4
F - Fisica | J - Juridica
F
Digite o ID da pessoa: 1
ID: 1
Nome: Edson Luiz
CPF: 123.456.789-11
Idade: 24

```

5. Exibir Todos:

```

3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====
1
F - Fisica | J - Juridica
F
Digite o nome: Sara
Digite o CPF: 987.654.321-01
Digite a idade: 24
=====
Escolha uma opcao:
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
F - Fisica | J - Juridica
J
Digite o nome: Coffe Break
Digite o CNPJ: 23.425.255/0001-75
=====
Escolha uma opcao:
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
F - Fisica | J - Juridica
F
ID: 1
Nome: Edson Luiz
CPF: 123.456.789-11
Idade: 24
ID: 2
Nome: Sara
CPF: 987.654.321-01
Idade: 24

```

6. Persistir Dados:

```
=====
Escolha uma opcao:
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
Digite o prefixo do arquivo
teste
Dados salvos com sucesso.
=====
Escolha uma opcao:
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
Programa finalizado.
BUILD SUCCESSFUL (total time: 7 minutes 7 seconds)
|
```

7. Recuperar Dados:

```
run:
=====
Escolha uma opcao:
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
Digite o prefixo do arquivo
teste
Dados recuperados com sucesso.
=====
Escolha uma opcao:
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
F - Fisica | J - Juridica
F
ID: 1
Nome: Edson Luiz
CPF: 123.456.789-11
Idade: 24
ID: 2
Nome: Sara
CPF: 987.654.321-01
Idade: 24
```

```
=====
Escolha uma opcao:
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
F - Fisica | J - Juridica
J
ID: 2
Nome: Coffe Break
CNPJ: 23.425.255/0001-75
```