

Disciplina:	RPG0014 – Iniciando o Caminho pelo Java
Nome:	Iggor Motta de Oliveira
Turma:	2024.3
Campus:	EAD
Curso:	Desenvolvimento Full stack
Link:	https://github.com/Iggor-motta/Miss-oPr-tica-N-vel-1-part2-Mundo3.git

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

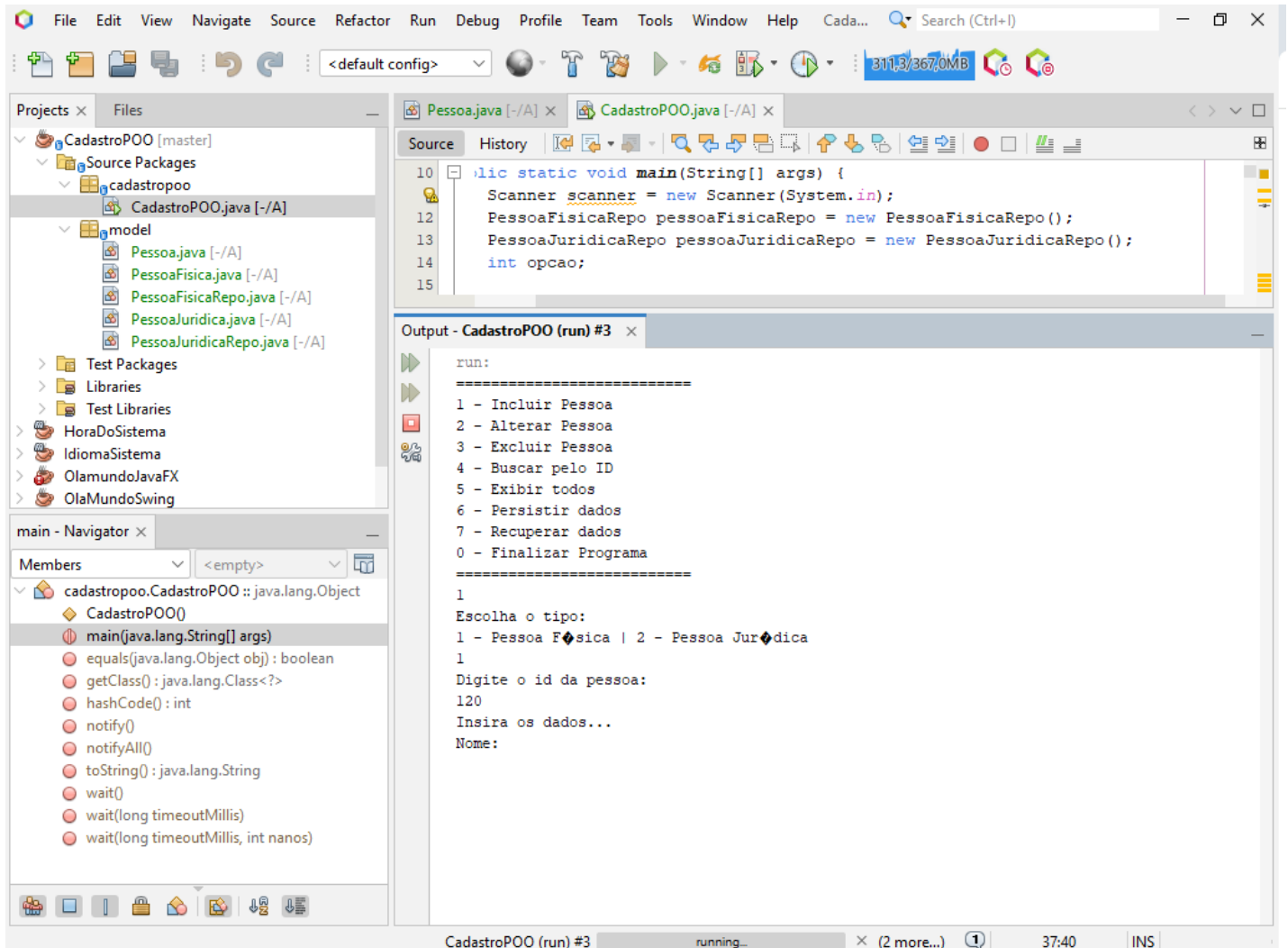
1. Título da Prática: “Procedimento | Criação do Cadastro em Modo Texto”

2. Objetivo da Prática:

- Implementar uma interface cadastral em modo texto.
- Utilizar o controle de exceções da plataforma Java.
- Utilizar os recursos da programação orientada a objetos e a persistência em arquivos binários.

3. Todos os códigos solicitados neste roteiro serão anexados no final do relatório.

4. Resultado da execução dos códigos:



The screenshot displays an IDE interface with the following components:

- Projects View:** Shows the project structure for 'CadastroPOO [master]'. It includes 'Source Packages' (cadastropoo) and 'Test Packages'. The 'cadastropoo' package contains 'CadastroPOO.java [-/A]' and a 'model' package with 'Pessoa.java [-/A]', 'PessoaFisica.java [-/A]', 'PessoaFisicaRepo.java [-/A]', 'PessoaJuridica.java [-/A]', and 'PessoaJuridicaRepo.java [-/A]'.
- Source View:** Displays the source code of 'CadastroPOO.java'. The code defines a static void main method that initializes a Scanner, creates instances of 'PessoaFisicaRepo' and 'PessoaJuridicaRepo', and declares an 'opcao' variable.
- Output View:** Shows the output of the program execution. The output is a menu-driven application for managing people.

Output - CadastroPOO (run) #3

```
run:
=====
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
Escolha o tipo:
1 - Pessoa Física | 2 - Pessoa Jurídica
1
Digite o id da pessoa:
120
Insira os dados...
Nome:
```

5. Análise e Conclusão

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

São membros de uma classe (variáveis ou métodos) que pertencem à classe em si, e não a instâncias individuais dessa classe. Membro estático é compartilhado por todas as instâncias da classe e pode ser acessado sem que seja necessário criar um objeto da classe.

O método main é o ponto de entrada do programa Java. Ele é chamado pelo Java Virtual Machine (JVM) quando o programa é iniciado. Como o main é chamado antes de qualquer objeto ser criado, ele precisa ser um método estático, permitindo que a JVM o execute sem a necessidade de criar uma instância da classe principal.

- Para que serve a classe Scanner?

Pacote `java.util` é utilizado para ler entradas do usuário ou de outras fontes de dados, como arquivos ou fluxos de entrada (`InputStream`). Ela facilita a leitura de diversos tipos de dados, como inteiros, floats, strings, etc.

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

O uso de classes de repositório impacta positivamente na organização e na manutenção do código, aplicando princípios de boas práticas de desenvolvimento. As classes de repositório permitem a reutilização da lógica de persistência em várias partes do código. Se um repositório é bem estruturado, ele pode ser utilizado em diferentes camadas ou sistemas sem precisar ser reescrito. Melhora a clareza, modularidade e manutenção do código, ao mesmo tempo que permite uma melhor adaptação a mudanças futuras no sistema.

Todos os códigos solicitados neste roteiro esta no link abaixo:

<https://github.com/lggor-motta/Miss-oPr-tica-N-vel-1-part2-Mundo3/blob/main/CadastroPOO.java>

Códigos:

```
package cadastrpoo;

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

public class CadastroPOO {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        PessoaFisicaRepo pessoaFisicaRepo = new
PessoaFisicaRepo();
        PessoaJuridicaRepo pessoaJuridicaRepo = new
PessoaJuridicaRepo();
        int opcao;

        try{

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

```

opcao = scanner.nextInt();
scanner.nextLine(); // Limpar buffe;

switch (opcao) {
    case 1: // Incluir
        System.out.println("Escolha o tipo:");
        System.out.println("1 - Pessoa Física | 2 -
Pessoa Jurídica");
        int tipoInclusao = scanner.nextInt();
        scanner.nextLine();

        if (tipoInclusao == 1) {
            System.out.println("Digite o id da pessoa:");
            int id = scanner.nextInt();
            scanner.nextLine();
            System.out.println("Insira os dados...");
            System.out.println("Nome:");
            String nome = scanner.nextLine();
            System.out.println("Informe o CPF:");
            String cpf = scanner.nextLine();
            System.out.println("Informe a idade:");
            int idade = scanner.nextInt();
            scanner.nextLine();

            PessoaFisica pf = new PessoaFisica(id,
nome, cpf, idade);
            pessoaFisicaRepo.inserir(pf);
            System.out.println("Pessoa física inserida
com sucesso!");

        } else if (tipoInclusao == 2) {
            System.out.println("Digite o id da pessoa:");
            int id = scanner.nextInt();
            scanner.nextLine();
            System.out.println("Informe o nome:");
            String nome = scanner.nextLine();
            System.out.println("Informe o CNPJ:");
            String cnpj = scanner.nextLine();

```

```

        PessoaJuridica pj = new PessoaJuridica(id,
nome, cnpj);
        pessoaJuridicaRepo.inserir(pj);
        System.out.println("Pessoa jurídica
inserida com sucesso!");
    }
    break;

```

```

    case 2: // Alterar
        System.out.println("Escolha o tipo para
alterar:");
        System.out.println("1 - Pessoa Física");
        System.out.println("2 - Pessoa Jurídica");
        int tipoAlteracao = scanner.nextInt();
        scanner.nextLine();

        if (tipoAlteracao == 1) {
            System.out.println("Informe o ID da pessoa
física a ser alterada:");
            int id = scanner.nextInt();
            scanner.nextLine();

```

```

            PessoaFisica pfExistente =
pessoaFisicaRepo.obter(id);
            if (pfExistente != null) {
                System.out.println("Dados atuais: " +
pfExistente);
                System.out.println("Informe o novo
nome:");
                String novoNome = scanner.nextLine();
                System.out.println("Informe o novo
CPF:");
                String novoCpf = scanner.nextLine();
                System.out.println("Informe a nova
idade:");
                int novaldade = scanner.nextInt();
                scanner.nextLine();

                pfExistente.setNome(novoNome);
                pfExistente.setCpf(novoCpf);

```

```

        pfExistente.setIdade(novaIdade);
        pessoaFisicaRepo.alterar(pfExistente);
        System.out.println("Pessoa física
alterada com sucesso!");
    } else {
        System.out.println("Pessoa física não
encontrada.");
    }

    } else if (tipoAlteracao == 2) {
        System.out.println("Informe o ID da pessoa
jurídica a ser alterada:");
        int id = scanner.nextInt();
        scanner.nextLine();

        PessoaJuridica pjExistente =
pessoaJuridicaRepo.obter(id);
        if (pjExistente != null) {
            System.out.println("Dados atuais: " +
pjExistente);
            System.out.println("Informe o novo
nome:");
            String novoNome = scanner.nextLine();
            System.out.println("Informe o novo
CNPJ:");
            String novoCnpj = scanner.nextLine();

            pjExistente.setNome(novoNome);
            pjExistente.setCnpj(novoCnpj);
            pessoaJuridicaRepo.alterar(pjExistente);
            System.out.println("Pessoa jurídica
alterada com sucesso!");
        } else {
            System.out.println("Pessoa jurídica não
encontrada.");
        }
    }
    }
    break;

```

```

        case 3: // Excluir
            System.out.println("Escolha o tipo para
excluir:");
            System.out.println("1 - Pessoa Física");
            System.out.println("2 - Pessoa Jurídica");
            int tipoExclusao = scanner.nextInt();
            scanner.nextLine();

            if (tipoExclusao == 1) {
                System.out.println("Informe o ID da pessoa
física a ser excluída:");
                int id = scanner.nextInt();
                scanner.nextLine();

                pessoaFisicaRepo.excluir(id);
                System.out.println("Pessoa física excluída
com sucesso!");
            } else if (tipoExclusao == 2) {
                System.out.println("Informe o ID da pessoa
jurídica a ser excluída:");
                int id = scanner.nextInt();
                scanner.nextLine();

                pessoaJuridicaRepo.excluir(id);
                System.out.println("Pessoa jurídica
excluída com sucesso!");
            }
            break;
        case 4: // Exibir por ID
            System.out.println("Escolha o tipo para exibir
por ID:");
            System.out.println("1 - Pessoa Física");
            System.out.println("2 - Pessoa Jurídica");
            int tipoExibirId = scanner.nextInt();
            scanner.nextLine();

            if (tipoExibirId == 1) {
                System.out.println("Informe o ID da pessoa
física:");
                int id = scanner.nextInt();

```



```

        scanner.nextLine();

        PessoaFisica pfExibir =
        pessoaFisicaRepo.obter(id);
        if (pfExibir != null) {
            pfExibir.exibir();
        } else {
            System.out.println("Pessoa física não
encontrada.");
        }
        } else if (tipoExibirId == 2) {
            System.out.println("Informe o ID da pessoa
jurídica:");
            int id = scanner.nextInt();
            scanner.nextLine();

            PessoaJuridica pjExibir =
            pessoaJuridicaRepo.obter(id);
            if (pjExibir != null) {
                pjExibir.exibir();
            } else {
                System.out.println("Pessoa jurídica não
encontrada.");
            }
        }
        break;

    case 5: // Exibir todos
        System.out.println("Escolha o tipo para exibir
todos:");
        System.out.println("1 - Pessoa Física");
        System.out.println("2 - Pessoa Jurídica");
        int tipoExibirTodos = scanner.nextInt();
        scanner.nextLine();

        if (tipoExibirTodos == 1) {
            for (PessoaFisica pf :
            pessoaFisicaRepo.obterTodos()) {
                pf.exibir();
            }
        }
    }
}

```

```

        } else if (tipoExibirTodos == 2) {
            for (PessoaJuridica pj :
                pessoaJuridicaRepo.obterTodos()) {
                pj.exibir();
            }
        }
        break;
    case 6: // Salvar dados
        System.out.println("Informe o prefixo do
arquivo para salvar:");
        String prefixoSalvar = scanner.nextLine();

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

    case 7: // Recuperar dados
        System.out.println("Informe o prefixo do
arquivo para recuperar:");
        String prefixoRecuperar = scanner.nextLine();

        try {

            pessoaFisicaRepo.recuperar(prefixoRecuperar +
".fisica.bin");

            pessoaJuridicaRepo.recuperar(prefixoRecuperar +
".juridica.bin");
            System.out.println("Dados recuperados
com sucesso!");
        } catch (Exception e) {

```

```

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

    case 0:
        System.out.println("Encerrando o
programa...");
        break;

    default:
        System.out.println("Opção inválida!");
    }

    while (opcao != 0);
} catch (Exception e){
}

scanner.close();
}

private static class pessoaFisicaRepo {

    private static PessoaFisica obter(int id) {
        throw new UnsupportedOperationException("Not
supported yet.");
    }
}
}

```