



**Campos Via Corpos**

**Desenvolvimento Full-Stack**

**Iniciando o Caminho Pelo Java**

**Turma 2022.3 – Mundo 3**

**Nome: Lucas Figueirêdo Costa de Queiroz.**

**Repositório: <https://github.com/Lucas8956/Mundo-3---MP1>**

**Missão Prática Nível 1 do Mundo 3**

**Parte 1**

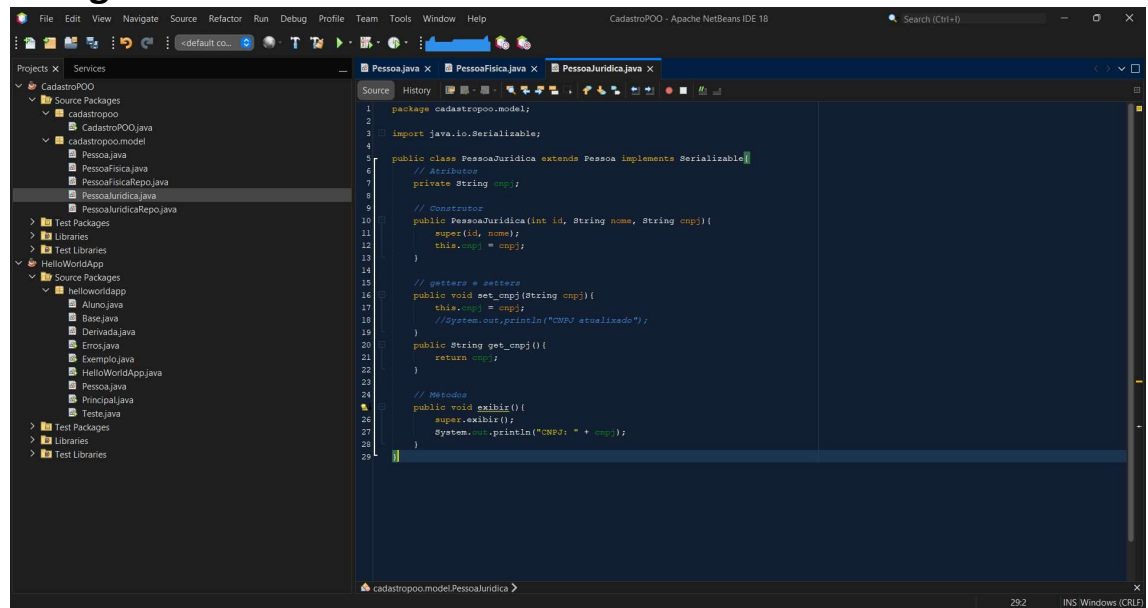
**Código Classe Pessoa**

```
1 package cadastrapoo.model;
2
3 import java.io.Serializable;
4
5 public class Pessoa implements Serializable {
6     // Atributos
7     private int id;
8     private String nome;
9
10    // Construtor
11    public Pessoa() {}
12
13    public Pessoa(int id, String nome) {
14        this.id = id;
15        this.nome = nome;
16    }
17
18    // getters e setters
19    public void set_id(int id) {
20        this.id = id;
21        //System.out.println("Id atualizado");
22    }
23    public int get_id() {
24        return id;
25    }
26    public void set_nome(String nome) {
27        this.nome = nome;
28        //System.out.println("Nome atualizado");
29    }
30    public String get_nome() {
31        return nome;
32    }
33
34    // Métodos
35    public void exibir() {
36        System.out.println("Id: " + id);
37        System.out.println("Nome: " + nome);
38    }
39}
```

**Código Classe PessoaFisica**

```
1 package cadastrapoo.model;
2
3 import java.io.Serializable;
4
5 public class PessoaFisica extends Pessoa implements Serializable {
6     // Atributos
7     private String cpf;
8     private int idade;
9
10    // Construtor
11    public PessoaFisica(int id, String nome, String cpf, int idade) {
12        super(id, nome);
13        this.cpf = cpf;
14        this.idade = idade;
15    }
16
17    // getters e setters
18    public void set_cpf(String cpf) {
19        this.cpf = cpf;
20        //System.out.println("CPF atualizado");
21    }
22    public String get_cpf() {
23        return cpf;
24    }
25    public void set_idade(int idade) {
26        this.idade = idade;
27        //System.out.println("Idade atualizada");
28    }
29    public int get_idade() {
30        return idade;
31    }
32
33    // Métodos
34    public void exibir() {
35        super.exibir();
36        System.out.println("CPF: " + cpf);
37        System.out.println("Idade: " + idade);
38    }
39}
```

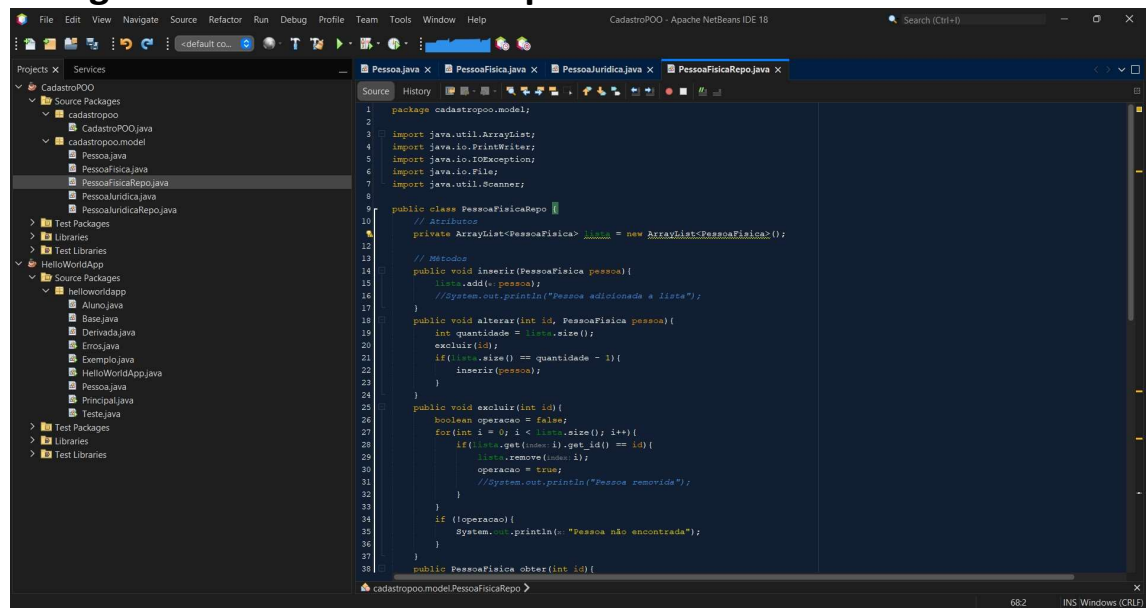
# Código Classe PessoaJuridica



The screenshot shows the NetBeans IDE with the file explorer on the left displaying the project structure. The main editor window shows the code for `PessoaJuridica.java`. The code includes package declarations, imports for `Serializable`, a constructor, getters and setters for the `empj` attribute, and a `exibir()` method that prints the object details.

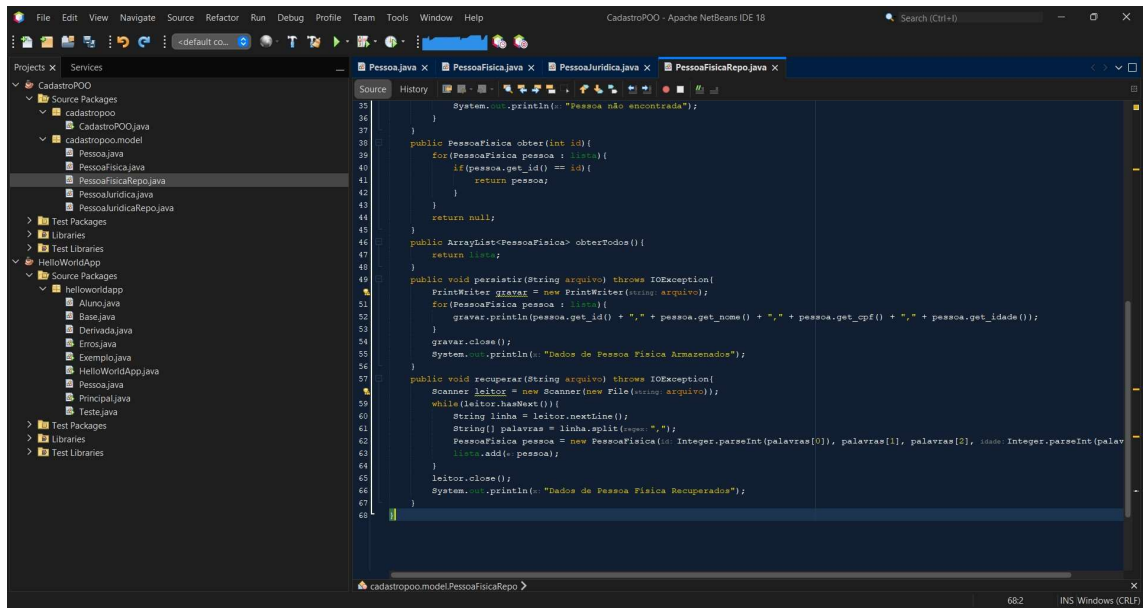
```
1 package cadastrpoo.model;
2
3 import java.io.Serializable;
4
5 public class PessoaJuridica extends Pessoa implements Serializable {
6
7     // Atributos
8     private String empj;
9
10    // Construtor
11    public PessoaJuridica(int id, String nome, String empj) {
12        super(id, nome);
13        this.empj = empj;
14    }
15
16    // getters e setters
17    public void set_empj(String empj) {
18        this.empj = empj;
19        //System.out.println("CPF atualizado");
20    }
21    public String get_empj() {
22        return empj;
23    }
24
25    // Métodos
26    public void exibir() {
27        super.exibir();
28        System.out.println("CPF: " + empj);
29    }
30 }
```

# Código Classe PessoaFisicaRepo

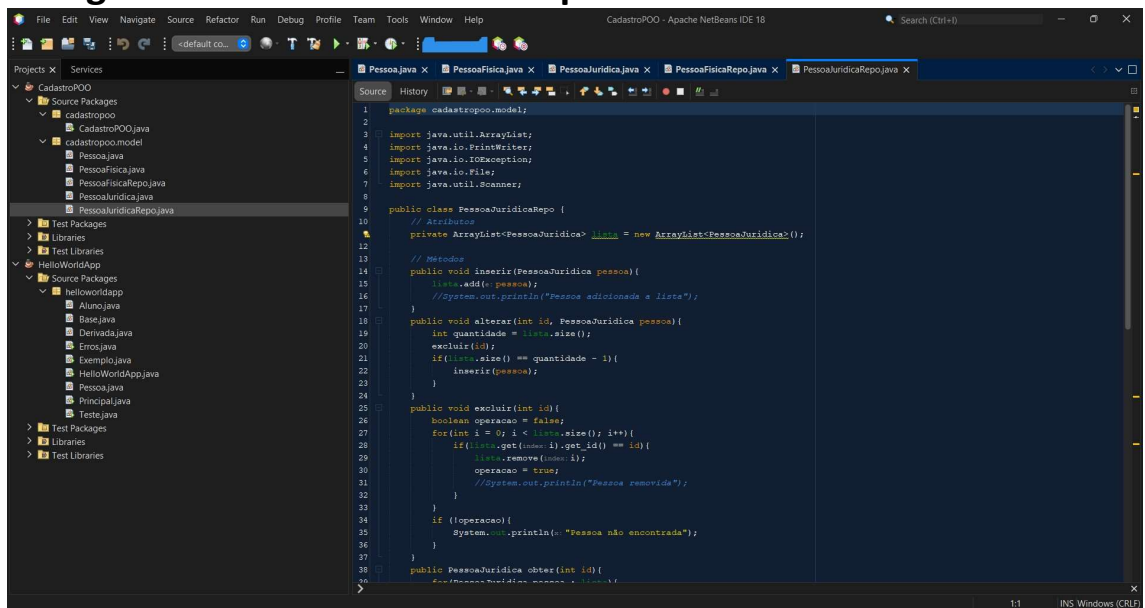


The screenshot shows the NetBeans IDE with the file explorer on the left displaying the project structure. The main editor window shows the code for `PessoaFisicaRepo.java`. The code includes package declarations, imports for `ArrayList`, `PrintWriter`, `IOException`, `File`, and `Scanner`. It defines a `PessoaFisicaRepo` class with a private `ArrayList` of `PessoaFisica` objects. The methods include `inserir()`, `alterar()`, `excluir()`, and `obter()`.

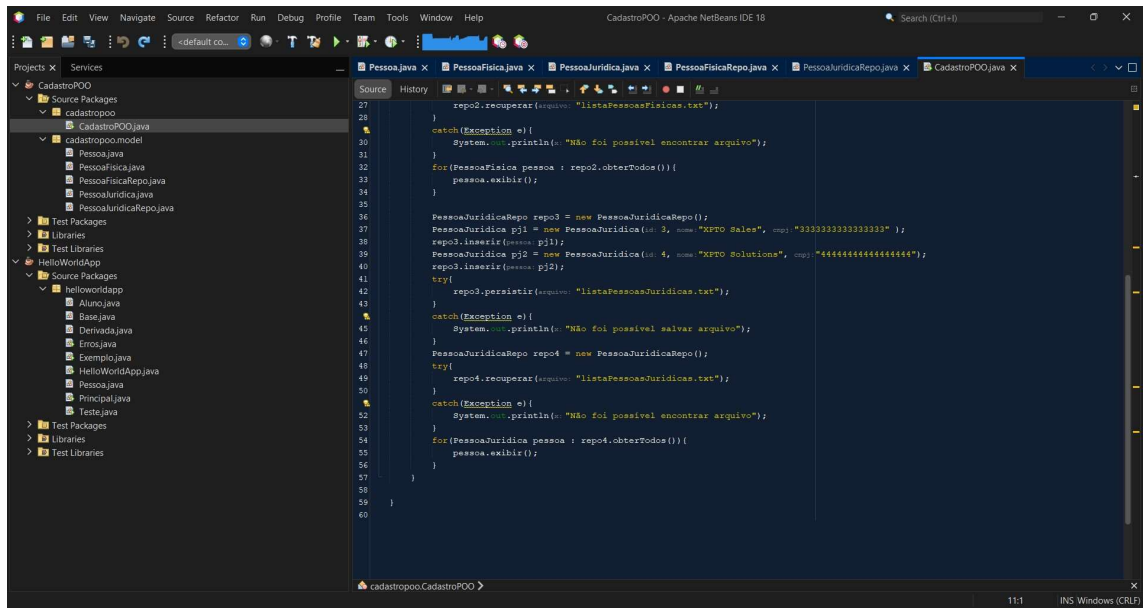
```
1 package cadastrpoo.model;
2
3 import java.util.ArrayList;
4 import java.io.PrintWriter;
5 import java.io.IOException;
6 import java.io.File;
7 import java.util.Scanner;
8
9 public class PessoaFisicaRepo {
10
11     // Atributos
12     private ArrayList<PessoaFisica> lista = new ArrayList<PessoaFisica>();
13
14     // Métodos
15     public void inserir(PessoaFisica pessoa) {
16         lista.add(pessoa);
17         //System.out.println("Pessoa adicionada a lista");
18     }
19     public void alterar(int id, PessoaFisica pessoa) {
20         int quantidade = lista.size();
21         excluir(id);
22         if (lista.size() == quantidade - 1) {
23             inserir(pessoa);
24         }
25     }
26     public void excluir(int id) {
27         boolean operacao = false;
28         for (int i = 0; i < lista.size(); i++) {
29             if (lista.get(i).get_id() == id) {
30                 lista.remove(i);
31                 operacao = true;
32                 //System.out.println("Pessoa removida");
33             }
34         }
35         if (!operacao) {
36             System.out.println("Pessoa não encontrada");
37         }
38     }
39     public PessoaFisica obter(int id) {
40
41     }
```



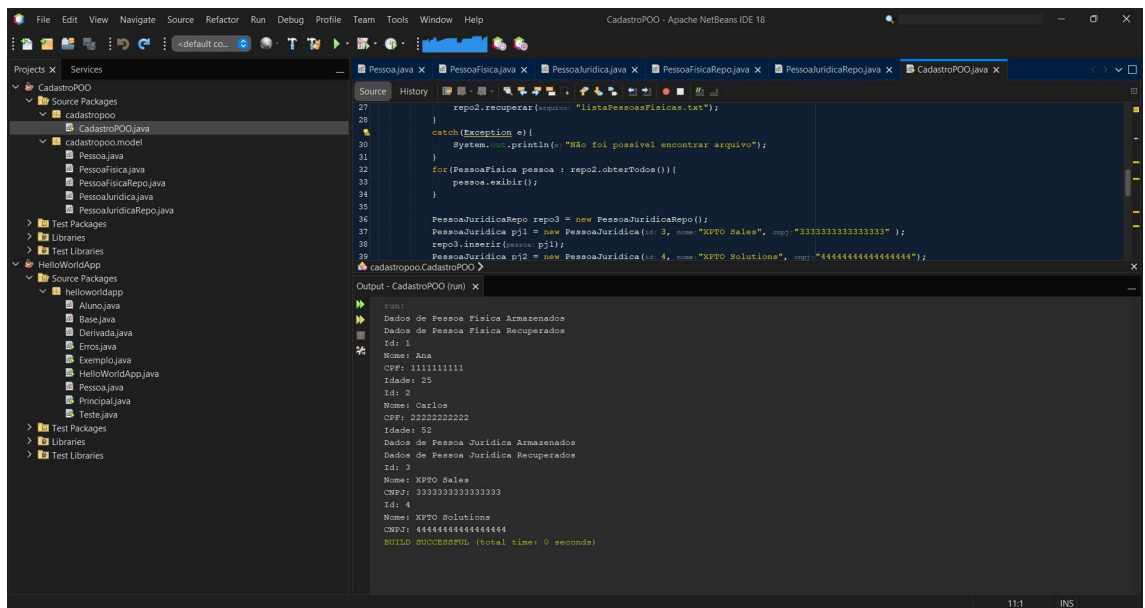
## Código Classe PessoaJuridicaRepo





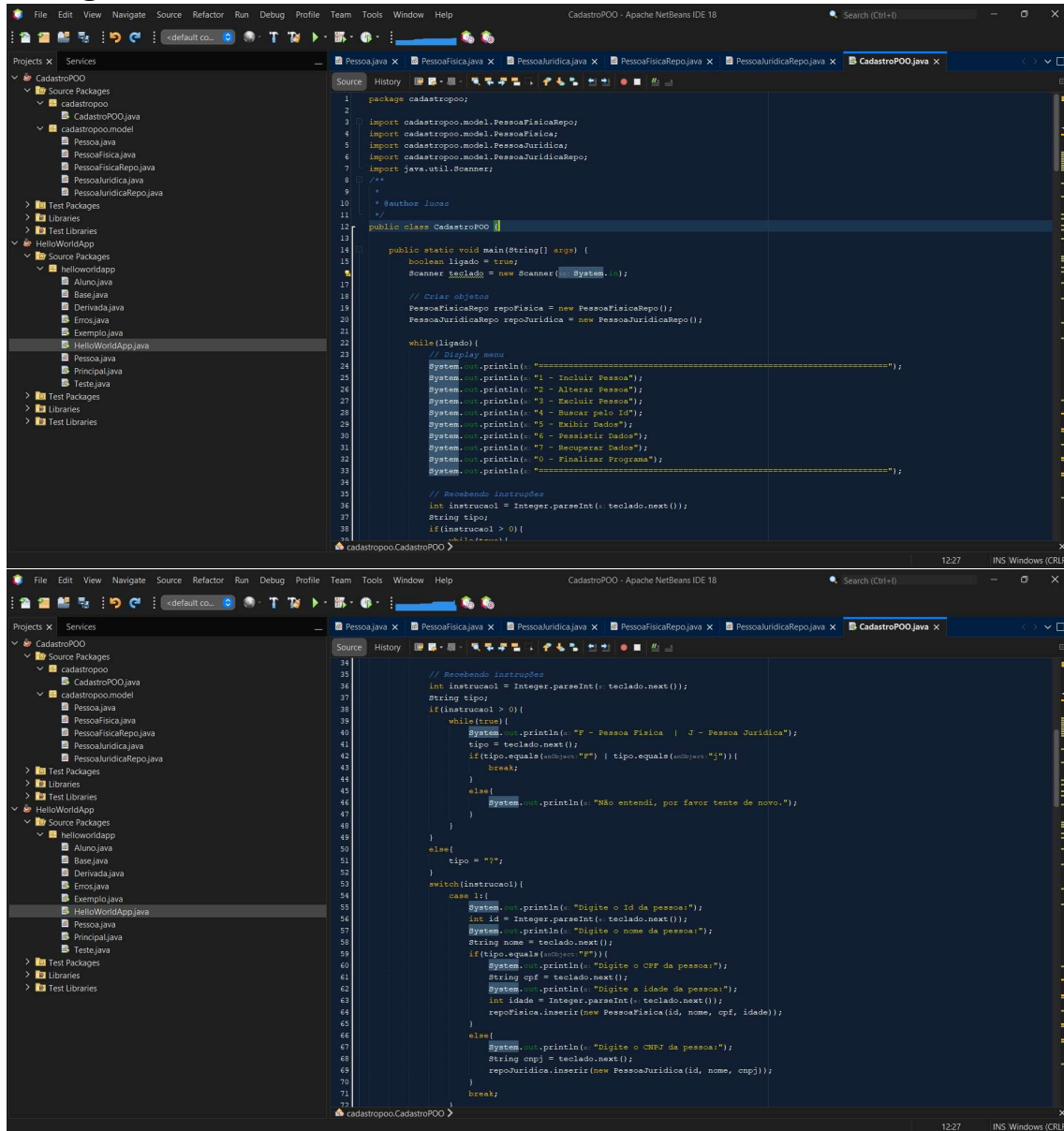


## Resultado



## Parte 2

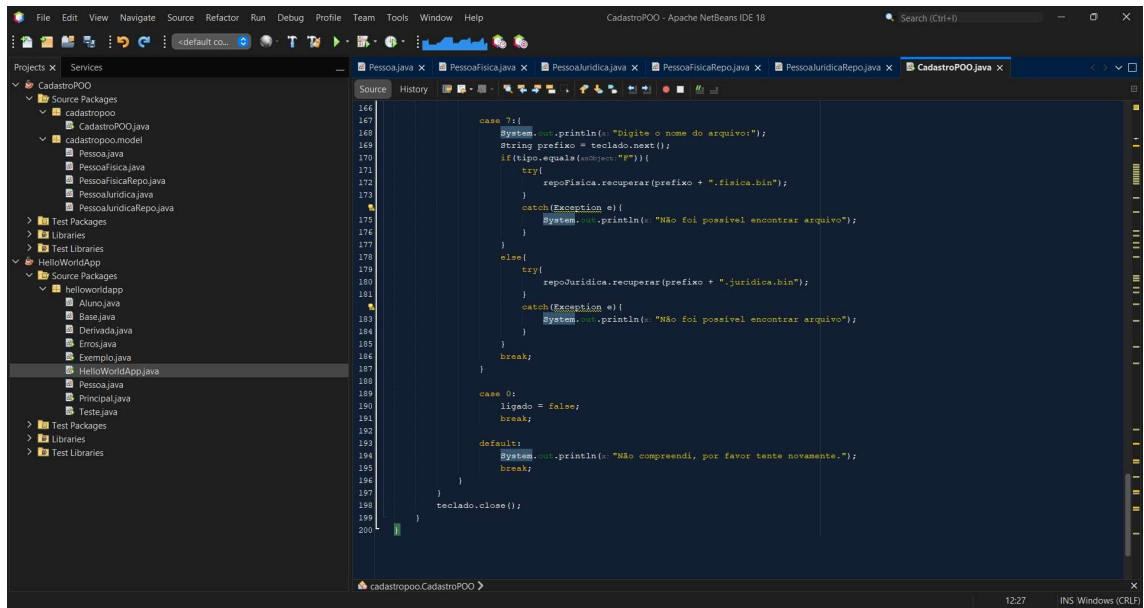
# Código Classe CadastroPOO



```
1 package cadastropoo;
2
3 import cadastropoo.model.PessoaFisicaRepo;
4 import cadastropoo.model.PessoaFisica;
5 import cadastropoo.model.PessoaJuridica;
6 import cadastropoo.model.PessoaJuridicaRepo;
7 import java.util.Scanner;
8
9 /**
10  * @author lucas
11  */
12 public class CadastroPOO {
13
14     public static void main(String[] args) {
15         Boolean ligado = true;
16         Scanner teclado = new Scanner(System.in);
17
18         // Criar objetos
19         PessoaFisicaRepo repofisica = new PessoaFisicaRepo();
20         PessoaJuridicaRepo repojuridica = new PessoaJuridicaRepo();
21
22         while (ligado) {
23             // Display menu
24             System.out.println("=====");
25             System.out.println("1 - Incluir Pessoa");
26             System.out.println("2 - Alterar Pessoa");
27             System.out.println("3 - Excluir Pessoa");
28             System.out.println("4 - Buscar pelo ID");
29             System.out.println("5 - Exibir Dados");
30             System.out.println("6 - Pesquisar Dados");
31             System.out.println("7 - Recuperar Dados");
32             System.out.println("0 - Finalizar Programa");
33             System.out.println("=====");
34
35             // Recebendo instruções
36             int instrucao1 = Integer.parseInt(teclado.next());
37             String tipo;
38             if (instrucao1 > 0) {
39                 while (true) {
40                     System.out.println("P - Pessoa Fisica | J - Pessoa Juridica");
41                     tipo = teclado.next();
42                     if (tipo.equals("P") || tipo.equals("J")) {
43                         break;
44                     } else {
45                         System.out.println("Não entendi, por favor tente de novo.");
46                     }
47                 }
48             } else {
49                 tipo = "?";
50             }
51             switch (instrucao1) {
52                 case 1:
53                     System.out.println("Digite o ID da pessoa:");
54                     int id = Integer.parseInt(teclado.next());
55                     System.out.println("Digite o nome da pessoa:");
56                     String nome = teclado.next();
57                     if (tipo.equals("P")) {
58                         System.out.println("Digite o CPF da pessoa:");
59                         String cpf = teclado.next();
60                         System.out.println("Digite a idade da pessoa:");
61                         int idade = Integer.parseInt(teclado.next());
62                         repofisica.inserir(new PessoaFisica(id, nome, cpf, idade));
63                     } else {
64                         System.out.println("Digite o CNPJ da pessoa:");
65                         String cnpj = teclado.next();
66                         repojuridica.inserir(new PessoaJuridica(id, nome, cnpj));
67                     }
68                 break;
69             }
70         }
71     }
72 }
```







## Resultado:

