Campos: Via Corpos

Curso: Desenvolvimento Full-Stack

Disciplina: Mundo3 – Nível 1 Iniciando o caminho pelo Java

Aluno: Lucas Figueirêdo Costa de Queiroz

Mundo 3 – Nível 1 - Missão Prática 1

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

Objetivos:

Implementar um cadastro de clientes em modo texto, com persistência em

arquivos, baseado na tecnologia Java utilizando-se de:

- Herança e polimorfismo na definição de entidades.

- Persistência de objetos em arquivos binários.

- Implementação de uma interface cadastral em modo texto.

- Utilização do controle de exceções da plataforma Java.

Tendo assim implementado um sistema cadastral em Java, utilizando os recursos da programação orientada a objetos e a persistência em arquivos binários.

1ª Parte

Códigos:

Classe Pessoa

```java
package model;

import java.io.Serializable;

public class Pessoa implements Serializable{
    //Atributos
    private int id;
    private String nome;

    //Construtores
    public Pessoa() {}

    public Pessoa(int id, String nome) {
        this.id  = id;
        this.nome = nome;
    }

    //getters e setters
    public void set_id(int id) {
        this.id = id;
        System.out.println("ID Atualizado.");
    }
    public int get_id() {
        return this.id;
    }
    public void set_nome(String nome) {
        this.nome = nome;
        System.out.println("Nome Atualizado.");
    }
    public String get_nome() {
        return this.nome;
    }

    //Métodos
    public void exibir() {
        System.out.println("ID: " + this.id);
        System.out.println("Nome: " + this.nome);
    }
}
```

Classe PessoaFisica

```java
package model;

import java.io.Serializable;

public class PessoaFisica extends Pessoa implements Serializable{
    //Atributos
    private String cpf;
    private int idade;

    //Construtores
    public PessoaFisica() {}

    public PessoaFisica(int id, String nome, String cpf, int idade) {
        super(id, nome);
        this.cpf = cpf;
        this.idade = idade;
    }

    //getters e setters
    public void set_cpf(String cpf) {
        this.cpf = cpf;
        System.out.println("CPF Atualizado.");
    }
    public String get_cpf() {
        return this.cpf;
    }
    public void set_idade(int idade) {
        this.idade = idade;
        System.out.println("Idade Atualizado.");
    }
    public int get_idade() {
        return this.idade;
    }

    //Métodos
    public void exibir() {
        super.exibir();
        System.out.println("CPF: " + this.cpf);
        System.out.println("Idade: " + this.idade);
    }
}
```

Classe PessoaJuridica

```java
package model;
```

```java
import java.io.Serializable;

public class PessoaJuridica extends Pessoa implements Serializable{
    //Atributos
    private String cnpj;

    //Construtores
    public PessoaJuridica() {}

    public PessoaJuridica(int id, String nome, String cnpj) {
        super(id, nome);
        this.cnpj = cnpj;
    }

    //getters e setters
    public void set_cnpj(String cnpj) {
        this.cnpj = cnpj;
        System.out.println("CNPJ Atualizado.");
    }
    public String get_cnpj() {
        return this.cnpj;
    }

    //Métodos
    public void exibir() {
        super.exibir();
        System.out.println("CNPJ: " + this.cnpj);
    }
}
```

Classe PessoaFisicaRepo

```java
package model;

import java.util.ArrayList;
import java.io.IOException;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectOutputStream;
import java.io.ObjectInputStream;

public class PessoaFisicaRepo {
    //Atributos
    private ArrayList<PessoaFisica> lista;

    //Construtor
    public PessoaFisicaRepo() {
        this.lista = new ArrayList<PessoaFisica>();
```

```
    }

    //Métodos
    public void inserir(PessoaFisica pessoa) {
        lista.add(pessoa);
        //System.out.println("Pessoa Adicionada");
    }

    public void alterar(int id, PessoaFisica pessoa) {
        int quantidade = this.lista.size();
        this.excluir(id);
        if(this.lista.size() == quantidade - 1) {
            this.inserir(pessoa);
            System.out.println("Pessoa Alterada");
        }
    }

    public void excluir(int id) {
        boolean operacao = false;
        for(int i = 0; i < this.lista.size(); i++) {
            if(this.lista.get(i).get_id() == id) {
                this.lista.remove(i);
                operacao = true;
                System.out.println("Pessoa Removida");
                break;
            }
        }
        if(!operacao) {
            System.out.println("Pessoa não encontrada");
        }
    }

    public PessoaFisica obter(int id) {
        for(PessoaFisica pessoa : this.lista) {
            if(pessoa.get_id() == id) {
                return pessoa;
            }
        }
        return new PessoaFisica();
    }

    public ArrayList<PessoaFisica> obterTodos(){
        return this.lista;
    }

    public void persistir(String arquivo_nome) throws IOException {
        ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(arquivo_nome));
        out.writeObject(this.lista);
```

```
        out.close();
        System.out.println("Dados de Pessoas Físicas Armazenados");
    }

    public void recuperar(String arquivo_nome) throws IOException,
ClassNotFoundException{
        ObjectInputStream in =  new ObjectInputStream(new
FileInputStream(arquivo_nome));
        this.lista = (ArrayList<PessoaFisica>)in.readObject();
        in.close();
        System.out.println("Dados de Pessoas Físicas Recuperados");
    }
}
```

Classe PessoaJuridicaRepo

```
package model;

import java.util.ArrayList;
import java.io.IOException;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectOutputStream;
import java.io.ObjectInputStream;

public class PessoaJuridicaRepo {
    //Atributos
    ArrayList<PessoaJuridica> lista;

    //Construtor
    public PessoaJuridicaRepo() {
        this.lista = new ArrayList<PessoaJuridica>();
    }

    //Métodos
    public void inserir(PessoaJuridica pessoa) {
        this.lista.add(pessoa);
        //System.out.println("Pessoa Adicionada");
    }

    public void alterar(int id, PessoaJuridica pessoa) {
        int quantidade = this.lista.size();
        this.excluir(id);
        if(this.lista.size() == quantidade - 1) {
            this.inserir(pessoa);
            System.out.println("Pessoa Alterada");
        }
    }
```

```
    public void excluir(int id) {
        boolean operacao = false;
        for(int i = 0; i < this.lista.size(); i++) {
            if(this.lista.get(i).get_id() == id) {
                this.lista.remove(i);
                operacao = true;
                System.out.println("Pessoa Removida");
                break;
            }
        }
        if(!operacao) {
            System.out.println("Pessoa não encontrada");
        }
    }

    public PessoaJuridica obter(int id) {
        for(PessoaJuridica pessoa : this.lista) {
            if(pessoa.get_id() == id) {
                return pessoa;
            }
        }
        return new PessoaJuridica();
    }

    public ArrayList<PessoaJuridica> obterTodos() {
        return this.lista;
    }

    public void persistir(String arquivo_nome) throws IOException {
        ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(arquivo_nome));
        out.writeObject(this.lista);
        out.close();
        System.out.println("Dados de Pessoas Físicas Armazenados");
    }

    public void recuperar(String arquivo_nome) throws IOException,
ClassNotFoundException {
        ObjectInputStream in =  new ObjectInputStream(new
FileInputStream(arquivo_nome));
        this.lista = (ArrayList<PessoaJuridica>)in.readObject();
        in.close();
        System.out.println("Dados de Pessoas Físicas Recuperados");
    }
}
```

Principal

```
package model;

public class Principal {

    public static void main(String[] args) {
        PessoaFisicaRepo repo1 = new PessoaFisicaRepo();
        repo1.inserir(new PessoaFisica(1, "Ana", "111.111.111-11", 25));
        repo1.inserir(new PessoaFisica(2, "Carlos", "222.222.222-22",
52));
        try {
            repo1.persistir("arquivoTestePessoaFisica");
        } catch (Exception e) {
            System.out.println("Não foi possível salvar arquivo.");
        }
        PessoaFisicaRepo repo2 = new PessoaFisicaRepo();
        try {
            repo2.recuperar("arquivoTestePessoaFisica");
            for(PessoaFisica pessoa : repo2.obterTodos()) {
                pessoa.exibir();
            }
        } catch (Exception e) {
            System.out.println("Arquivo não encontrado.");
        }
        PessoaJuridicaRepo repo3 = new PessoaJuridicaRepo();
        repo3.inserir(new PessoaJuridica(3, "XPTO Sales",
"333.333.333.333-33"));
        repo3.inserir(new PessoaJuridica(4, "XPTO Solutions",
"444.444.444.444-44"));
        try {
            repo3.persistir("arquivoTestePessoaJuridica");
        } catch (Exception e) {
            System.out.println("Não foi possível salvar arquivo.");
        }
        PessoaJuridicaRepo repo4 = new PessoaJuridicaRepo();
        try {
            repo4.recuperar("arquivoTestePessoaJuridica");
            for(PessoaJuridica pessoa : repo4.obterTodos()) {
                pessoa.exibir();
            }
        } catch (Exception e) {
            System.out.println("Arquivo não encontrado.");
        }
    }
}
```

Resultados:

```
Dados de Pessoas Físicas Armazenados
Dados de Pessoas Físicas Recuperados
ID: 1
Nome: Ana
CPF: 111.111.111-11
Idade: 25
ID: 2
Nome: Carlos
CPF: 222.222.222-22
Idade: 52
Dados de Pessoas Jurídicas Armazenados
Dados de Pessoas Jurídicas Recuperados
ID: 3
Nome: XPTO Sales
CNPJ: 333.333.333.333-33
ID: 4
Nome: XPTO Solutions
CNPJ: 444.444.444.444-44
```

Respostas do questionário:

a) As vantagens são reutilização de códigos, maior organização e facilidade de manutenção.

b) Ela avisa a JVM que o objeto pode ser salvo como uma sequência de dados;

c) Através de métodos das classes Collections.

2ª Parte

Código:

Classe Principal

```java
package model;

import java.util.Scanner;

public class Principal {

    public static void main(String[] args) {
        boolean ligado = true;
        Scanner teclado = new Scanner(System.in);
        PessoaFisicaRepo repoFisica = new PessoaFisicaRepo();
        PessoaJuridicaRepo repoJuridica = new PessoaJuridicaRepo();

        while(ligado) {
```

```
            //Display menu
            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("================================================================
====");

            //Seguindo instruções
            int instrucao1 = -1;
            String tipo = "?";

            try {
                instrucao1 = Integer.parseInt(teclado.next());
                }
            catch (Exception e) {
                    System.out.println("Entrada inválida.");
                }
            if(instrucao1 > 0 & instrucao1 < 8) {
                while(true) {
                    System.out.println("F - Pessoa Física | J - Pessoa Jurídica");
                    tipo = teclado.next();
                    if(tipo.equals("F") | tipo.equals("j")) {
                        break;
                    }
                    else {
                        System.out.println("Não entendi, por favor tente novamente.");
                    }
                }
            }

            switch(instrucao1) {
            case 1:
                while(true) {
                    System.out.println("Digite o ID da pessoa: ");
                    try {
                        int id = Integer.parseInt(teclado.next());
                        System.out.println("Insira os dados... ");
                        System.out.println("Nome: ");
                        String nome = teclado.next();
                        if(tipo.equals("F")) {
                            System.out.println("CPF: ");
                            String cpf = teclado.next();
```

```
                    System.out.println("Idade: ");
                    int idade = Integer.parseInt(teclado.next());
                    repoFisica.inserir(new PessoaFisica(id, nome, cpf, idade));
                    break;
                }
                else {
                    System.out.println("CNPJ: ");
                    String cnpj = teclado.next();
                    repoJuridica.inserir(new PessoaJuridica(id, nome, cnpj));
                    break;
                }
            }
            catch (Exception e) {
                System.out.println("Entrada inválida, por favor tente novamente.");
            }
        }
        break;

    case 2:
        while(true) {
            System.out.println("Digite o ID da pessoa: ");
            try {
                int id = Integer.parseInt(teclado.next());
                System.out.println("Insira os novos dados... ");
                System.out.println("Nome: ");
                String nome = teclado.next();
                if(tipo.equals("F")) {
                    System.out.println("CPF: ");
                    String cpf = teclado.next();
                    System.out.println("Idade: ");
                    int idade = Integer.parseInt(teclado.next());
                    repoFisica.alterar(id, new PessoaFisica(id, nome, cpf, idade));
                    break;
                }
                else {
                    System.out.println("CNPJ: ");
                    String cnpj = teclado.next();
                    repoJuridica.alterar(id, new PessoaJuridica(id, nome, cnpj));
                    break;
                }
            }
            catch (Exception e) {
                System.out.println("Entrada inválida, por favor tente novamente.");
            }
        }
        break;

    case 3:
        while(true) {
```

```
                System.out.println("Digite o ID da pessoa: ");
                try {
                    int id = Integer.parseInt(teclado.next());
                    if(tipo.equals("F")) {
                        repoFisica.excluir(id);
                        break;
                    }
                    else {
                        repoJuridica.excluir(id);
                        break;
                    }
                }
                catch (Exception e) {
                    System.out.println("Entrada inválida, por favor tente novamente.");
                }
            }
            break;

        case 4:
            while(true) {
                System.out.println("Digite o ID da pessoa: ");
                try {
                    int id = Integer.parseInt(teclado.next());
                    if(tipo.equals("F")) {
                        repoFisica.obter(id).exibir();
                        break;
                    }
                    else {
                        repoJuridica.obter(id).exibir();
                        break;
                    }
                }
                catch (Exception e) {
                    System.out.println("Entrada inválida, por favor tente novamente.");
                }
            }
            break;

        case 5:
            if(tipo.equals("F")) {
                for(PessoaFisica pessoa : repoFisica.obterTodos()) {
                    pessoa.exibir();
                }
            }
            else {
                for(PessoaJuridica pessoa : repoJuridica.obterTodos()) {
                    pessoa.exibir();
                }
            }
```

```
                break;

            case 6:
                System.out.println("Digite o nome do arquivo: ");
                String prefixo6 = teclado.next();
                try {
                    if(tipo.equals("F")) {
                        repoFisica.persistir(prefixo6 + ".fisica.bin");
                    }
                    else {
                        repoJuridica.persistir(prefixo6 + ".juridica.bin");
                    }
                }
                catch (Exception e) {
                    System.out.println("Não foi possível salvar arquivo.");
                }
                break;

            case 7:
                System.out.println("Digite o nome do arquivo: ");
                String prefixo7 = teclado.next();
                try {
                    if(tipo.equals("F")) {
                        repoFisica.recuperar(prefixo7 + ".fisica.bin");
                    }
                    else {
                        repoJuridica.recuperar(prefixo7 + ".juridica.bin");
                    }
                }
                catch (Exception e) {
                    System.out.println("Não foi possível encontrar o arquivo.");
                }
                break;

            case 0:
                ligado = false;
                System.out.println("Fechando programa.");
                break;

            default:
                System.out.println("Por favor tente novamente.");
                break;
            }
        }
        teclado.close();
    }
}
```

Pesultados:

```
=============================================================================
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
F - Pessoa Física | J - Pessoa Jurídica
F
Digite o nome do arquivo:
teste
Dados de Pessoas Físicas Recuperados
=============================================================================
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 - Pessoa Física | J - Pessoa Jurídica
F
ID: 1
Nome: Ana2
CPF: 111.111.111-11
Idade: 11
ID: 2
Nome: Carlos2
CPF: 222.222.222-22
Idade: 22
=============================================================================
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
Fechando programa.
```

Respostas do questionário:

d) São elementos que pertencem a classe e não ao objeto. Assim métodos estáticos podem ser chamados diretamente sem precisar instanciar um objeto.

e) Para ler dados de várias fontes diferentes.

f) Deixou muito mais organizado separando as instruções de acesso das de construção.