Wellington Cesar Fonseca

DESENVOLVIMENTO FULL STACK

MISSÃO PRÁTICA | NÍVEL 1 | MUNDO 3

RPG0014 - INICIANDO O CAMINHO PELO JAVA

OBJETIVO

1. Utilizar herança e polimorfismo na definição de entidades.
2. Utilizar persistência de objetos em arquivos binários.
3. Implementar uma interface cadastral em modo texto.
4. Utilizar o controle de exceções da plataforma Java.
5. 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.

1° PROCEDIMENTO | CRIAÇÃO DAS ENTIDADES E SISTEMA DE PERSISTÊNCIA

Quais as vantagens e desvantagens do uso de herança?

- vantagem: reutilização de código
- vantagem: traz uma melhor organização do código, pois será feito mais uso e menos linhas
- desvantagem: se usar de modo excessivo por trazer confusão e dificuldade na manutenção
- desvantagem: pesquisando sobre encontrei que não existe a possibilidade de incrementar múltiplas heranças

Por que a interface Serializable é necessária ao efetuar persistência em arquivos binários?

- creio que ele funcione como banco de dados, é através dele que possuímos as "conexões" e "conversões" necessárias para salvar os dados

Como o paradigma funcional é utilizado pela API stream no Java?

- O paradigma funcional utiliza meios de operações de alto nível que podem manipular coleções de elementos, essas operações permitem escrever código de forma mais eficiente.

Quando trabalhamos com Java, qual padrão de desenvolvimento é adotado na persistência de dados em arquivos?

- Um dos padrões comumentes usado é o DAO (Data Access Object). O padrão DAO separa a lógica de acesso aos dados da lógica de negócios do aplicativo, permitindo uma melhor manutenção do código. De modo geral envolve criação de classes que envolvem operações de CRUD (Create, Read, Update, Delete)

CadastroPOO1.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this
template
 */
package cadastropoo;
import java.io.IOException;
import model.PessoaFisica;
import model.PessoaJuridica;
import model.PessoaFisicaRepo;
import model.PessoaJuridicaRepo;
import java.util.ArrayList;
import java.util.Scanner;
/**
 *
 * @author wellingtonfonseca
 */
public class CadastroPOO1 {

 /**
 * @param args the command line arguments
 */
 public static void main(String[] args) {
 try {
 PessoaFisicaRepo repo1 = new PessoaFisicaRepo();

 PessoaFisica pessoaFisica1 = new PessoaFisica(1, "Wellington", "123.456.789-00", 31);
 PessoaFisica pessoaFisica2 = new PessoaFisica(2, "Amanda", "001.234.567-89", 30);

 System.out.println("pessoas_fisicas: inserindo");

 repo1.inserir(pessoaFisica1);
 repo1.inserir(pessoaFisica2);

 System.out.println("pessoas_fisicas: persistindo");

 repo1.persistir("pessoas_fisicas.dat");

 PessoaFisicaRepo repo2 = new PessoaFisicaRepo();

 System.out.println("pessoas_fisicas: recuperando");

 repo2.recuperar("pessoas_fisicas.dat");

 ArrayList<PessoaFisica> pessoasFisicasRecuperadas = repo2.obterTodos();

 for (PessoaFisica pessoa : pessoasFisicasRecuperadas) {
 pessoa.exibir();
 }

 PessoaJuridicaRepo repo3 = new PessoaJuridicaRepo();

 PessoaJuridica pessoaJuridica1 = new PessoaJuridica(1, "Empresa A",
"12345678901234");
 PessoaJuridica pessoaJuridica2 = new PessoaJuridica(2, "Empresa B",
"98765432109876");

 System.out.println("pessoas_fisicas: inserindo");

 repo3.inserir(pessoaJuridica1);
 repo3.inserir(pessoaJuridica2);

 System.out.println("pessoas_fisicas: persistindo");

 repo3.persistir("pessoas_juridicas.bin");

 PessoaJuridicaRepo repo4 = new PessoaJuridicaRepo();

 System.out.println("pessoas_fisicas: recuperando");

 repo4.recuperar("pessoas_juridicas.bin");

 ArrayList<PessoaJuridica> pessoasJuridicasRecuperadas = repo4.obterTodos();

 for (PessoaJuridica pessoa : pessoasJuridicasRecuperadas) {
 pessoa.exibir();
 }
 } catch (IOException ex) {
 System.err.println("Erro ao manipular arquivos: " + ex.getMessage());

 } catch (ClassNotFoundException ex) {
 System.err.println("Erro ao recuperar objeto: Classe não encontrada.");
 }
 }
}
```

model/Pessoa.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
template
 */
package model;
import java.io.Serializable;


/**
 *
 * @author wellingtonfonseca
 */
public class Pessoa implements Serializable {
 public void exibir() {
 System.out.println("id: " + id);
 System.out.println("nome: " + nome);
 }

 private int id;
 private String nome;

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

 public int getId() {
 return id;
 }

 public void setId(int id) {
 this.id = id;
 }

 public String getNome() {
 return nome;
 }

 public void setNome(String nome) {
 this.nome = nome;
 }
}
```

model/PessoaFisica.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
template
 */
package model;
import java.io.Serializable;

/**
 *
 * @author wellingtonfonseca
 */
public class PessoaFisica extends Pessoa implements Serializable {
 @Override
 public void exibir() {
 super.exibir();
 System.out.println("cpf: " + cpf);
 System.out.println("idade: " + idade);
 }

 private String cpf;
 private int idade;

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

 public String getCpf() {
 return cpf;
 }

 public void setCpf(String cpf) {
 this.cpf = cpf;
 }

 public int getIdade() {
 return idade;
 }

 public void setIdade(int idade) {
 this.idade = idade;
 }
}
```

model/PessoaFisicaRepo.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
template
 */
package model;
import java.io.*;
import java.util.ArrayList;
/**
 *
 * @author wellingtonfonseca
 */
public class PessoaFisicaRepo implements Serializable {
 private ArrayList<PessoaFisica> pessoasFisicas = new ArrayList<>();

 public void inserir(PessoaFisica pessoaFisica) {
 pessoasFisicas.add(pessoaFisica);
 }

 public void alterar(PessoaFisica pessoa) {
 for (int i = 0; i < pessoasFisicas.size(); i++) {
 PessoaFisica p = pessoasFisicas.get(i);
 if (p.getId() == pessoa.getId()) {
 pessoasFisicas.set(i, pessoa);
 return;
 }
 }
 throw new IllegalArgumentException("Pessoa não encontrada para alteração.");
 }

 public void excluir(int id) {
 pessoasFisicas.removeIf(pessoa -> pessoa.getId() == id);
 }

 public PessoaFisica obter(int id) {
 for (PessoaFisica pessoa : pessoasFisicas) {
 if (pessoa.getId() == id) {
 return pessoa;
 }
 }
 throw new IllegalArgumentException("Pessoa não encontrada.");
 }

 public ArrayList<PessoaFisica> obterTodos() {
 return pessoasFisicas;
 }

 public void persistir(String nomeArquivo) throws IOException {
 try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(nomeArquivo))) {
 out.writeObject(pessoasFisicas);
 }
 }

 public void recuperar(String nomeArquivo) throws IOException,
ClassNotFoundException {
 try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream(nomeArquivo))) {
 pessoasFisicas = (ArrayList<PessoaFisica>) in.readObject();
 }
 }
}
```

model/PessoaJuridica.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
template
 */
package model;
import java.io.Serializable;

/**
 *
 * @author wellingtonfonseca
 */
public class PessoaJuridica extends Pessoa implements Serializable {
 @Override
 public void exibir() {
 super.exibir(); // Chamando o método exibir da superclasse Pessoa
 System.out.println("CNPJ: " + cnpj);
 }

 private String cnpj;

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

 public String getCnpj() {
 return cnpj;
 }

 public void setCnpj(String cnpj) {
 this.cnpj = cnpj;
 }
}
```

model/PessoaJuridicaRepo.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
template
 */
package model;
import java.io.*;
import java.util.ArrayList;
/**
 *
 * @author wellingtonfonseca
 */
public class PessoaJuridicaRepo implements Serializable {
 private ArrayList<PessoaJuridica> pessoasJuridicas = new ArrayList<>();

 public void inserir(PessoaJuridica pessoaJuridica) {
 pessoasJuridicas.add(pessoaJuridica);
 }

 public void alterar(PessoaJuridica pessoaJuridica) {
 for (int i = 0; i < pessoasJuridicas.size(); i++) {
 PessoaJuridica p = pessoasJuridicas.get(i);
 if (p.getId() == pessoaJuridica.getId()) {
 pessoasJuridicas.set(i, pessoaJuridica);
 return;
 }
 }
 throw new IllegalArgumentException("Pessoa jurídica não encontrada para
alteração.");
 }

 public void excluir(int id) {
 pessoasJuridicas.removeIf(pessoaJuridica -> pessoaJuridica.getId() == id);
 }

 public PessoaJuridica obter(int id) {
 for (PessoaJuridica pessoaJuridica : pessoasJuridicas) {
 if (pessoaJuridica.getId() == id) {
 return pessoaJuridica;
 }
 }
 throw new IllegalArgumentException("Pessoa jurídica não encontrada.");
 }

 public ArrayList<PessoaJuridica> obterTodos() {
 return pessoasJuridicas;
 }

 public void persistir(String nomeArquivo) throws IOException {
 try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(nomeArquivo))) {
 out.writeObject(pessoasJuridicas);
 }
 }

 public void recuperar(String nomeArquivo) throws IOException,
ClassNotFoundException {
 try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream(nomeArquivo))) {
 pessoasJuridicas = (ArrayList<PessoaJuridica>) in.readObject();
 }
 }
}
```