Missão prática | Nível 1 | Mundo 3

CAMPUS: POLO DISTRITO JK - ANÁPOLIS - GO

CURSO: DESENVOLVIMENTO FULL STACK

NÚMERO DA TURMA: 2024.3

SEMESTRE LETIVO: Segundo semestre de 2024

ALIUNO: Henrique Rodrigues Rabello Vieira

MATRÍCULA: 202301230527

RPG0014 – Iniciando o caminho pelo Java

Objetivos:

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

Segundo procedimento:

Códigos solicitados:

Opções:

```java
package cadastropoo;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;
import model.PessoaFisica;
import model.PessoaFisicaRepo;
import model.PessoaJuridica;
import model.PessoaJuridicaRepo;


public class CadastroPOOsegunda {

    public static void main(String[] args) throws IOException, ClassNotFoundException {

        Scanner in = new Scanner(System.in);
        PessoaFisicaRepo repoPFisica = new PessoaFisicaRepo();
        PessoaJuridicaRepo repoPJuridica = new PessoaJuridicaRepo();

        do{
        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("===========================");

        int input = in.nextInt();
```

Opção incluir:

```java
        switch(input){
            case(1) -> {
                System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
                String pessoaTipoI = in.next();
                if(pessoaTipoI.equals("F")){
                    System.out.println("Insira os dados...");

                    System.out.print("Nome: ");
                    String nome = in.next();

                    System.out.print("CPF: ");
                    String cpf = in.next();

                    System.out.print("Idade: ");
                    int idade = in.nextInt();

                    int id = repoPFisica.obterTodos().size() + 1;

                    PessoaFisica pessoaFNova = new PessoaFisica(id, nome, cpf, idade);
                    repoPFisica.inserir(pessoaFNova);

                    System.out.println("Dados adicionados com sucesso.");
                }
                else if(pessoaTipoI.equals("J")){
                    System.out.println("Insira os dados...");

                    System.out.print("Nome: ");
                    String nome = in.next();

                    System.out.print("CNPJ: ");
                    String cnpj = in.next();

                    int id = repoPJuridica.obterTodos().size() + 1;

                    PessoaJuridica pessoaJNova = new PessoaJuridica(id,nome, cnpj);
                    repoPJuridica.inserir(pessoaJNova);
                    System.out.println("Dados adicionados com sucesso.");
                }
                }
```

Opção alterar:

```
case(2) -> {
    System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
    String pessoaTipoA = in.next();
    if(pessoaTipoA.equals("F")){
        System.out.print("Insira um id: ");
        int id = in.nextInt();
        try{
            PessoaFisica pessoaAlterar = repoPFisica.obter(id);
            pessoaAlterar.exibir();

            System.out.println("Insira os dados...");

            System.out.print("Novo Nome: ");
            String nome = in.next();

            System.out.print("Novo CPF: ");
            String cpf = in.next();

            System.out.print("Nova Idade: ");
            int idade = in.nextInt();

            PessoaFisica pessoaFNova = new PessoaFisica(id, nome, cpf, idade);
            PessoaFisicaRepo pessoaFisicaRepo = new PessoaFisicaRepo();
            pessoaFisicaRepo.alterar(pessoaAlterar, pessoaFNova);
            pessoaAlterar.exibir();
            System.out.print("Dados alterados com sucesso.");
        }catch(Throwable e){
            System.out.println("Erro ao obter dados: " + e.getMessage());
            e.printStackTrace();
        }
    }
```

```
    }
    else if(pessoaTipoA.equals("J")){

        System.out.print("Insira um id: ");
        int id = in.nextInt();
        try{
            PessoaJuridica pessoaAlterar = repoPJuridica.obter(id);
            pessoaAlterar.exibir();

            System.out.println("Insira os dados...");

            System.out.print("Novo Nome: ");
            String nome = in.next();

            System.out.print("Novo CNPJ: ");
            String cnpj = in.next();

            PessoaJuridica pessoaJNova = new PessoaJuridica(id, nome, cnpj);
            PessoaJuridicaRepo pessoaJuridicaRepo = new PessoaJuridicaRepo();
            pessoaJuridicaRepo.alterar(pessoaAlterar, pessoaJNova);
            pessoaAlterar.exibir();
            System.out.print("Dados alterados com sucesso.");
        }catch(Throwable e){
            System.out.println("Erro ao obter dados: " + e.getMessage());
            e.printStackTrace();
        }
    }
}
```

Opção excluir:

```
case(3) -> {
    System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
    String pessoaTipoE = in.next();
    if(pessoaTipoE.equals("F")){
        System.out.print("Insira um id: ");
        int id = in.nextInt();
        try{
            repoPFisica.excluir(id);
            System.out.println("Dados removidos.");
        }catch(Throwable e){
            System.out.println("Erro ao excluir dados: " + e.getMessage());
            e.printStackTrace();
        }
    }
    else if(pessoaTipoE.equals("J")){
        System.out.print("Insira um id: ");
        int id = in.nextInt();
        try{
            repoPJuridica.excluir(id);
            System.out.println("Dados removidos.");
        }catch(Throwable e){
            System.out.println("Erro ao excluir dados: " + e.getMessage());
            e.printStackTrace();
        }
    }
}
```

Opção obter:

```
159             case(4) -> {
160                 System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
161                 String pessoaTipoO = in.next();
162                 if(pessoaTipoO.equals("F")){
163                     System.out.print("Insira um id: ");
164                     int id = in.nextInt();
165 
166                     try{
167                         PessoaFisica pessoaObtida = repoPFisica.obter(id);
168                         pessoaObtida.exibir();
169                     }catch(Throwable e){
170                         System.out.println("Erro ao obter dados: " + e.getMessage());
                            e.printStackTrace();
172                     }
173                 }
174                 else if(pessoaTipoO.equals("J")){
175                     System.out.print("Insira um id: ");
176                     int id = in.nextInt();
177 
178                     try{
179                         PessoaJuridica pessoaObtida = repoPJuridica.obter(id);
180                         pessoaObtida.exibir();
181                     }catch(Throwable e){
182                         System.out.println("Erro ao obter dados: " + e.getMessage());
                            e.printStackTrace();
```

Opção obter todos:

```
187 
188             case(5) -> {
189                 System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");
190                 String pessoaTipoOT = in.next();
191                 if(pessoaTipoOT.equals("F")){
192                     ArrayList<PessoaFisica> pessoaObtida = repoPFisica.obterTodos();
193                     for(PessoaFisica p: pessoaObtida){
194                         p.exibir();
195                     }
196                 }
197                 else if(pessoaTipoOT.equals("J")){
198                     ArrayList<PessoaJuridica> pessoaObtida = repoPJuridica.obterTodos();
199                     for(PessoaJuridica p: pessoaObtida){
200                         p.exibir();
201                     }
202                 }
203                 }
204 
```

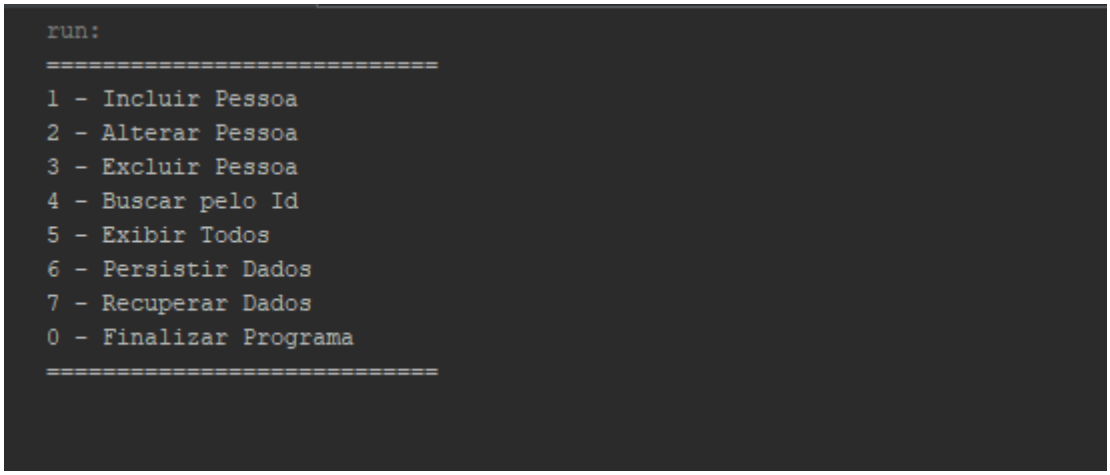
Opção salvar:

```
205             case(6) -> {
206                 System.out.print("Insira um prefixo para salvar o arquivo: ");
207                 String arquivoNomeS = in.next();
208                 try{
209                     repoPFisica.persistir(arquivoNomeS + ".fisica.bin");
210                     repoPJuridica.persistir(arquivoNomeS + ".juridica.bin");
211                     System.out.println("Dados salvos com sucesso.");
212                 }catch(IOException e){
213                     System.out.println("Erro ao salvar dados: " + e.getMessage());
                        e.printStackTrace();
215                 }
216                 }
```

Opção Recuperar:

```
218             case(7) -> {
219                 System.out.print("Insira um prefixo para recuperar o arquivo: ");
220                 String arquivoNomeR = in.next();
221                 try{
222                     repoPFisica.recuperar(arquivoNomeR + ".fisica.bin");
223                     repoPJuridica.recuperar(arquivoNomeR + ".juridica.bin");
224                     System.out.println("Dados recuperados com sucesso.");
225                 }catch(IOException | ClassNotFoundException e){
226                     System.out.println("Erro ao recuperar dados: " + e.getMessage());
                        e.printStackTrace();
228                 }
229                 }
```

Resultados:



Análise e conclusão:

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

Elementos estáticos são elementos que pertencem a classe e podem ser acessados sem a criação de um objeto. O método main adota esse modificador para que possa ser acessado/utilizado sem ser instanciado.

Para que serve a classe Scanner?

Utilizada para ler dados de entrada, muito usada para permitir interações entre o usuário e o programa.

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

O uso de classes repositório impactou positivamente a organização do código deixando ele mais legível.