Criação do cadastro em modo texto

Objetivo da prática:

* 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.
* 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.

Códigos Solicitados:

public class CadastroPOO {

public static void main(String[] args) {

PessoaFisicaRepo repo1 = new PessoaFisicaRepo();

PessoaFisica p1 = new PessoaFisica("nome",10,"000",18);

PessoaFisica p2 = new PessoaFisica("nome",22,"000",18);

PessoaFisica p3 = new PessoaFisica("nome",29,"000",18);

repo1.inserir(p1,p2,p3);

try {

repo1.persistir("PFr.ser");

} catch (IOException ex) {

Logger.getLogger(CadastroPOO.class.getName()).log(Level.SEVERE, null, ex);

}

PessoaFisicaRepo repo2 = new PessoaFisicaRepo();

try {

repo2.recuperar("PFr.ser");

} catch (IOException | ClassNotFoundException ex) {

Logger.getLogger(CadastroPOO.class.getName()).log(Level.SEVERE, null, ex);

}

PessoaJuridicaRepo repo3 = new PessoaJuridicaRepo();

PessoaJuridica p4 = new PessoaJuridica("nome",01,"000");

PessoaJuridica p5 = new PessoaJuridica("nome",02,"000");

PessoaJuridica p6 = new PessoaJuridica("nome",03,"000");

repo3.inserir(p4,p5,p6);

try {

repo3.persistir("PJr.ser");

} catch (IOException ex) {

Logger.getLogger(CadastroPOO.class.getName()).log(Level.SEVERE, null, ex);

}

PessoaJuridicaRepo repo4 = new PessoaJuridicaRepo();

try {

repo4.recuperar("PJr.ser");

} catch (IOException | ClassNotFoundException ex) {

Logger.getLogger(CadastroPOO.class.getName()).log(Level.SEVERE, null, ex);

}

Scanner scan = new Scanner(System.in);

String opc = "S";

while("s".equals(opc)|| "S".equals(opc)){

System.out.println("========================");

System.out.printf("1 - Incluir Pessoa%n"

+ "2 - Alterar Pessoa%n"

+ "3 - Excluir Pessoa%n"

+ "4 - Buscar pelo ID%n"

+ "5 - Exibir Todos%n"

+ "6 - Persistir Dados%n"

+ "7 - Recuperar Dados%n"

+ "0 - Finalizar Programa%n");

System.out.println("========================");

int n1 = scan.nextInt();

switch(n1){

case 1 -> {

System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");

scan.nextLine();

String s1 = scan.nextLine();

System.out.println("Digite o Id da Pessoa:");

int n2 = scan.nextInt();

switch(s1){

case "F" ->{

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

System.out.println("Nome:");

scan.nextLine();

String s2 = scan.nextLine();

System.out.println("Cpf:");

String s3 = scan.nextLine();

System.out.println("Idade:");

int n3 = scan.nextInt();

PessoaFisica pessoa = new PessoaFisica(s2,n2,s3,n3);

repo2.inserir(pessoa);

}

case "J" ->{

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

System.out.println("Nome:");

String s2 = scan.nextLine();

System.out.println("Cnpj:");

String s3 = scan.nextLine();

PessoaJuridica pessoa = new PessoaJuridica(s2,n2,s3);

repo4.inserir(pessoa);

}

}

break;

}

case 2 ->{

System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");

scan.nextLine();

String s1 = scan.nextLine();

System.out.println("Digite o Id da Pessoa:");

int n2 = scan.nextInt();

switch(s1){

case "F" -> {

repo2.obter(n2);

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

System.out.println("Nome:");

scan.nextLine();

String s2 = scan.nextLine();

System.out.println("Cpf:");

String s3 = scan.nextLine();

System.out.println("Idade:");

int n3 = scan.nextInt();

PessoaFisica pessoa = new PessoaFisica(s2,n2,s3,n3);

repo2.alterar(pessoa);

}

case "J" -> {

repo4.obter(n2);

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

System.out.println("Nome:");

scan.nextLine();

String s2 = scan.nextLine();

System.out.println("Cnpj:");

String s3 = scan.nextLine();

PessoaJuridica pessoa = new PessoaJuridica(s2,n2,s3);

repo4.alterar(pessoa);

}

}

break;

}

case 3 ->{

System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");

scan.nextLine();

String s1 = scan.nextLine();

System.out.println("Digite o Id da Pessoa:");

int n2 = scan.nextInt();

switch(s1){

case "F" -> {

repo2.excluir(n2);

}

case "J" -> {

repo4.excluir(n2);

}

}

break;

}

case 4 ->{

System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");

scan.nextLine();

String s1 = scan.nextLine();

System.out.println("Digite o Id da Pessoa:");

int n2 = scan.nextInt();

switch(s1){

case "F" -> {

repo2.obter(n2);

}

case "J" -> {

repo4.obter(n2);

}

}

break;

}

case 5 ->{

System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");

scan.nextLine();

String s1 = scan.nextLine();

switch(s1){

case "F" -> {

repo2.obterTodos();

}

case "J" -> {

repo4.obterTodos();

}

}

break;

}

case 6 ->{

System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");

scan.nextLine();

String s1 = scan.nextLine();

System.out.println("Nome do local de salvamento do arquivo");

String s2 = scan.nextLine();

switch(s1){

case "F" -> {

try {

repo2.persistir(s2+".fisica.bin");

} catch (IOException ex) {

Logger.getLogger(CadastroPOO.class.getName()).log(Level.SEVERE, null, ex);

}

}

case "J" -> {

try {

repo4.persistir(s2+"juridica.bin");

} catch (IOException ex) {

Logger.getLogger(CadastroPOO.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

break;

}

case 7 ->{

System.out.println("F - Pessoa Fisica | J - Pessoa Juridica");

scan.nextLine();

String s1 = scan.nextLine();

System.out.println("Nome do local de salvamento do arquivo");

String s2 = scan.nextLine();

switch(s1){

case "F" -> {

try {

repo2.recuperar(s2+".fisica.bin");

} catch (IOException | ClassNotFoundException ex) {

Logger.getLogger(CadastroPOO.class.getName()).log(Level.SEVERE, null, ex);

}

}

case "J" -> {

try {

repo4.recuperar(s2+"juridica.bin");

} catch (IOException | ClassNotFoundException ex) {

Logger.getLogger(CadastroPOO.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

break;

}

case 0 ->{

opc = "N";

continue;

}

}

System.out.println("Deseja fazer mais alguma alteração?");

System.out.println("S - Sim | N - Não");

if(n1 == 5 |n1 == 6 |n1 == 7){

opc = scan.nextLine();

}else{

scan.nextLine();

opc = scan.nextLine();

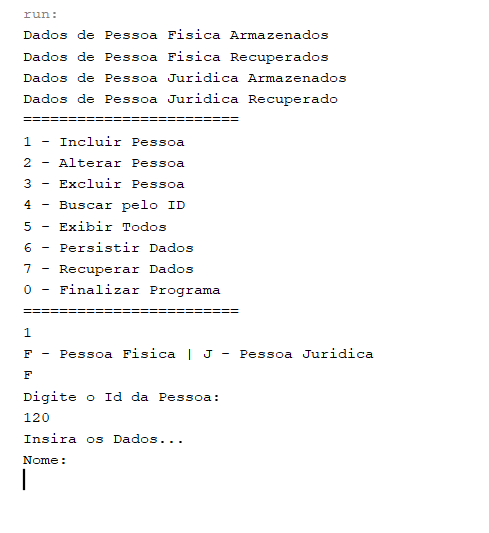
}

}

}

}

Resultado da execução do código:



Análise e Conclusão:

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

São elementos que podem ser executados livremente sem a necessidade de ser instanciado. O método main é executado sem precisar da instanciação por isso adota o static.

* Para que serve a classe Scanner?

Ela possui a finalidade de separar a entrada dos textos em blocos. Deste modo, permite a criação dos tokens.

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

Permitiu que o código ficasse mais limpo e mais organizado, assim concentrando todo o código no repositório.