



Relatório discente de acompanhamento

Gabriel Henrique dos Santos – Matrícula 202208292411

Polo: Limeira Centro
TURMA 2023.3 - MUNDO 3 - Iniciando o caminho pelo Java

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.

1º Procedimento| Criação das Entidades e Sistema de Persistência

Na etapa inicial, foi necessário criar as entidades, persistir os dados e conduzir testes. As classes Pessoa, PessoaFisica (herdando de Pessoa) e PessoaJuridica (herdando de Pessoa) foram utilizadas para instanciar objetos, que foram posteriormente armazenados em instâncias das classes PessoaJuridicaRepo e PessoaFisicaRepo. Essas instâncias, por meio do método de inserção, registravam os objetos em um ArrayList.

```
public class PessoaJuridicaRepo implements Serializable {  
    private ArrayList<PessoaJuridica> listaPessoasJuridicas = new ArrayList<>();  
  
    public void inserir (PessoaJuridica pessoaJuridica) { this.listaPessoasJuridicas.add(pessoaJuridica); }  
  
    public class PessoaFisicaRepo {  
        private ArrayList<PessoaFisica> listaPessoasFisicas = new ArrayList<>();  
  
        public void inserir (PessoaFisica pessoaFisica) { this.listaPessoasFisicas.add(pessoaFisica); }  
    }  
}
```

Para conduzir o teste de persistência, foram criadas duas instâncias de

classes de repositório para cada categoria (Física e Jurídica). Na primeira instância, dois objetos foram adicionados por meio de código, e em seguida, o método persistir foi chamado, especificando o nome do arquivo para salvá-los. Posteriormente, na segunda instância, que inicialmente não continha nenhum objeto armazenado, o método "recuperar" foi executado, utilizando o mesmo nome de arquivo. Dessa forma, o repositório carregou os objetos previamente armazenados pela outra instância.

```
public class Main {
    public static void main(String[] args) {
        int contadorIds = 1;

        PessoaFisica leandro = new PessoaFisica(contadorIds, nome: "Leandro", cpf: "123.123.123", idade: 12);
        contadorIds++;
        PessoaFisica claudio = new PessoaFisica(contadorIds, nome: "Claudio", cpf: "123.121233.123", idade: 12);
        contadorIds++;

        PessoaFisicaRepo repo1 = new PessoaFisicaRepo();
        repo1.inserir(leandro);
        repo1.inserir(claudio);

        try {
            repo1.persistir( nomeArquivo: "pessoas.fisica.bin");
        } catch (FileNotFoundException e) {
            System.out.println(e);
        }

        PessoaFisicaRepo repo2 = new PessoaFisicaRepo();

        try {
            repo2.recuperar( nomeArquivo: "pessoas.fisica.bin");
            repo2.obterTodos().forEach(pessoaFisica -> {
                System.out.println("Id: " + pessoaFisica.getId());
                System.out.println("Nome: " + pessoaFisica.getNome());
                System.out.println("CPF: " + pessoaFisica.getCpf());
                System.out.println("Idade: " + pessoaFisica.getIdade());
                System.out.println("-----");
            });
        } catch (FileNotFoundException e) {
            System.out.println(e);
        }
    }
}
```

Resultado do terminal

```
"C:\Program Files\Java\jdk-17\bin\java.exe"
Dados de Pessoa Fisica Armazenados.
Dados de Pessoa Fisica Recuperados.
Id: 1
Nome: Leandro
CPF: 123.123.123
Idade: 12
-----
Id: 2
Nome: Claudio
CPF: 123.121233.123
Idade: 12
-----
Dados de Pessoa Juridica Armazenados.
Dados de Pessoa Juridica Recuperados.
Id: 3
Nome: Casas Bahia
CNPJ: 11111111111
-----
Id: 4
Nome: magalu
CNPJ: 11111111111
-----
Process finished with exit code 0
```

Análise e Conclusão:

(a) - Quais as vantagens e desvantagens do uso de herança?

A principal vantagem é a reutilização de código: A herança permite que uma classe herde características e comportamentos de uma classe existente, promovendo a reutilização de código.

Quanto à desvantagem é atenuada, já que mudanças na classe base podem afetar as classes derivadas, levando a um acoplamento forte entre elas.

(b) - Por que a interface Serializable é necessária ao efetuar persistência em arquivos binários?

A interface Serializable em Java é usada para indicar que a classe pode ser serializada, ou seja, pode ser convertida em uma sequência de bytes.

(c) - Como o paradigma funcional é utilizado pela API stream no Java?

A API Stream em Java é baseada no paradigma funcional e introduz funcionalidades de programação funcional na linguagem, permitindo assim o trabalho adequado com uso de bibliotecas.

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

Em Java, ao trabalhar com a persistência de dados em arquivos, um padrão comum é o uso do padrão de projeto "DAO" (Data Access Object).

2º Procedimento | Criação do Cadastro em Modo Texto

Este procedimento envolveu a criação de uma interface para que o usuário pudesse realizar ações no programa. Inicialmente, é exibido o seguinte menu:

```
=====
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
=====
```

O menu foi feito com um “Do while”, que tem como condição de parada a resposta 0. As ações das opções são realizadas por um switch case. Nas opções 1, 2, 3, 4 e 5, é solicitado a escolha entre Pessoa Física e Jurídica.

```
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====
5
F - Pessoa Física | J - Pessoa Juridica
|
```

As opções de persistir e recuperar dados pedem um nome do arquivo que é usado como prefixo da seguinte forma: [prefixo].fisica.bin e [prefixo].juridica.bin.

```
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====
0
Nome do arquivo:
```

Análise e Conclusão:

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

Elementos estáticos em Java são membros de uma classe que pertencem à própria classe em vez de pertencerem a instâncias individuais da classe. O método main é o ponto de entrada para programas Java. Ele precisa ser declarado como public static void main para que o Java possa chamar o método sem criar uma instância da classe.

(b) - Para que serve a classe Scanner?

A classe Scanner em Java é usada para obter entradas do usuário a partir do console ou de outros fluxos de entrada.

(c) - Como o uso de classes de repositório impactou na organização do código?

O uso de classe facilitou muito a organização do código. O fato de se aproximar da realidade onde cada objeto da vida real possui elementos e realiza certas ações faz com que facilite o pensamento lógico da aplicação.