

Documentação do Projeto - Gerenciamento de Locais

Descrição do Projeto

Este projeto é uma aplicação em Java para gerenciar locais que um usuário já frequenta, com a possibilidade de avaliar e comentar sobre cada local. O sistema organiza esses locais geograficamente, utilizando uma estrutura hierárquica (país, estado, cidade e zona da cidade) e aplicativos importantes de Programação Orientada a Objetos.

Estrutura do Projeto

O código foi organizado em cinco pacotes principais para facilitar o entendimento e a manutenção:

1. **app** - Contém o arquivo principal `GerenciamentoLocaisApp.java`, responsável pela execução do sistema.
 2. **entidades** - Armazena as classes principais `Usuario` e `Local`, representando o usuário e os locais frequentados.
 3. **geografia** - Contém as classes `Pais`, `Estado`, `Cidade` e `ZonaCidade` para organizar a localização geográfica dos locais.
 4. **interfaces** - Armazena uma interface `Avaliavel`, que define o comportamento de avaliação para os locais.
 5. **utils** - Inclui a classe `Avaliacao`, que representa cada avaliação.
-

Classe Geografico(geografia)

Descrição

`Geografico` é uma **classe abstrata** que serve como base para classes geográficas (`Pais`, `Estado`, `Cidade` e `ZonaCidade`). Todos os níveis geográficos associados ao atributo `nome`, que são passados pelo construtor e acessados por métodos `get` e `set`.

Conceito Aplicado

- **Herança** : Geografico é uma superclasse para Pais, Estado, Cidade e ZonaCidade, evitando duplicação de código.
 - **Encapsulamento** : O atributo nome é privado, e seu acesso é controlado pelos métodos públicos getNome e setNome.
-

Aulas Geográficas (Pais, Estado, Cidade, ZonaCidade)

Cada uma dessas classes herdada Geografico e representa um nível da estrutura geográfica, contendo listas do próximo nível hierárquico.

Descrição

A classe Pais herdada e nome mantém Geografico uma lista de Estados, que representa os estados dentro do país. A mesma estrutura se repete para Estado, que contém Cidades, e Cidade, que contém ZonaCidades.

Conceito Aplicado

- **Composição** : Cada Pais contém uma lista de Estados. Essa composição organiza os locais em uma posição geográfica.
 - **Encapsulamento** : O atributo estados é privado, e o acesso ocorre através dos métodos adicionarEstado e getEstados.
-

Classe Local(entidades)

Descrição

Local representa um lugar que o usuário frequentou. A aula contém informações como nome, endereço, e detalhes da localização (zonaCidade, cidade, estado, pais). Além disso, Local possui uma lista de Avaliacoes.

Conceito Aplicado

- **Interface** : Local implementa Avaliavel, garantindo que todos os objetos da classe Local tenham o método avaliar.

- **Encapsulamento** : Atributos como nome, endereço e avaliações privados, com acesso controlado por métodos.
 - **Composição** : Cada um Local possui uma lista de Avaliações, demonstrando que um local é composto por avaliações.
-

Interface Avaliavel(interfaces)

Descrição

Avaliavel é uma interface que define o método avaliar, que toda classe que implementa Avaliavel deve definir.

Conceito Aplicado

- **Polimorfismo** : Qualquer classe que implemente Avaliavel pode ser tratada como um tipo Avaliavel, independentemente da classe específica. Isso permite flexibilidade na adição de novos tipos de objetos disponíveis no futuro.
-

Classe Avaliacao(utilitários)

Descrição

Avaliacao representa uma única avaliação com atributos nota e comentário.

Conceito Aplicado

- **Encapsulamento** : nota e comentários são privados, e o acesso é feito através de métodos como getDetalhes.
-

Classe Usuario(entidades)

Descrição

Usuario representa o usuário do sistema, que armazena uma lista dos Locais visitados e pode avaliar esses locais.

Conceito Aplicado

- **Associação Simples** : `Usuario` e `Local` tem uma relação de associação simples, já que `Usuario` tem uma lista de locais visitados.
 - **Encapsulamento** : Atributos como `nome` e `locaisFrequentados` são privados, com acesso controlado por métodos.
-

Diretor de classe `GerenciamentoLocaisApp(app)`

Descrição

`GerenciamentoLocaisApp` é o ponto de entrada do sistema. Ele inicializa um usuário e um local, cadastra o local para o usuário e adiciona uma avaliação.

Conceito Aplicado

- **Encapsulamento e Organização Modular** : Cada funcionalidade é distribuída em classes distintas e chamadas por métodos, organizando o fluxo do programa de forma clara e modular.
-

Resumo dos Conceitos de POO Aplicados

1. **Encapsulamento** : Aplicado em todas as classes, com atributos privados e acesso controlado por métodos `get` e `set`.
 2. **Herança** : Usada para organizar as aulas de localização geográfica, com a classe `Geografico` autônoma como superclasse.
 3. **Polimorfismo** : Utilizado com interface `Avaliavel`, permitindo que várias classes possam ser avaliadas de forma padronizada.
 4. **Composição** : `Local` possui uma lista de `Avaliacao`s, representando uma composição forte entre as classes.
 5. **Associação Simples** : `Usuario` possui uma
-

Essa estrutura para