

Atividade – Aula 15 – Padrão de Projeto Observer

Aluno: Nicolas Silva Araújo – 5º Período

Explicação do Padrão e do código – plataforma de vídeo on-demand (Foi utilizado o padrão de projeto: **Observer**, pois com esse padrão é possível que um objeto (o *sujeito*) notifique automaticamente outros objetos (*observadores*) sobre mudanças em seu estado, nesse código o usuário pode explicitamente escolher se deseja receber ou não notificações, e também é permitido que ele possa alterar essa escolha a qualquer momento.)

O sistema é composto por:

- **Conteudo:** Classe que representa um filme ou série, com título e gênero.
- **Plataforma:** Classe que gerencia a lista de filmes e usuários. Ela notifica os usuários quando um novo filme é adicionado.
- **Usuario:** Classe que representa um usuário, que pode se inscrever em um ou mais gêneros de filmes.
- **Observer Interface:** Interface que define o método de notificação dos usuários.

Explicação do Código

- **Inscrições Dinâmicas:** Cada usuário tem um mapa (Map<String, Boolean>) que gerencia suas inscrições por gênero, o usuário pode se inscrever e cancelar sua inscrição a qualquer momento.
- **Notificação Condicional:** Quando um novo filme é adicionado, a plataforma notifica apenas os usuários inscritos no gênero correspondente, o usuário que cancelou sua inscrição em um gênero não receberá notificações para filmes desse gênero.
- **Flexibilidade:** A implementação permite que os usuários tenham controle total sobre as notificações que desejam receber, com a possibilidade de inscrição e cancelamento a qualquer momento.
- **Ativação e Desativação de Notificações:** Cada usuário pode escolher explicitamente se deseja ser notificado sobre filmes de determinado gênero, utilizando os métodos `ativarNotificacao` e `desativarNotificacao`.

- As preferências são armazenadas em um mapa (Map<String, Boolean>), onde a chave é o gênero e o valor é true ou false dependendo da escolha do usuário.
- **Método update:** O método update é chamado quando um novo filme é adicionado à plataforma. Se o usuário tiver ativado as notificações para o gênero do filme, ele será notificado.

Explicação da UML

Conteudo

- **Atributos:**
 - **titulo: String:** Armazena o título do filme ou série.
 - **genero: String:** Armazena o gênero do filme ou série.
- **Métodos:**
 - **Conteudo(titulo: String, genero: String):** Construtor da classe, que inicializa os atributos titulo e genero.
 - **getTitulo(): String:** Retorna o título do conteúdo.
 - **getGenero(): String:** Retorna o gênero do conteúdo.

Resumo: A classe Conteudo encapsula os dados de um filme ou série, incluindo seu título e gênero.

Plataforma

- **Atributos:**
 - **conteudos: List<Conteudo>:** Uma lista que armazena todos os conteúdos (filmes e séries) adicionados à plataforma.
- **Métodos:**
 - **Plataforma():** Construtor da classe, que inicializa a lista de conteúdos.
 - **inscreverUsuario(usuario: Usuario):** Adiciona um usuário à lista de usuários inscritos na plataforma.
 - **adicionarFilmeSerie(conteudo: Conteudo):** Adiciona um novo conteúdo à plataforma e notifica os usuários inscritos.

- **notificarUsuarios(conteudo: Conteudo):** Método privado que percorre todos os usuários inscritos e os notifica sobre o novo conteúdo, se estiverem interessados no gênero.

Resumo: A classe Plataforma gerencia os conteúdos disponíveis e os usuários inscritos. Ela é responsável por adicionar novos conteúdos e notificar os usuários de acordo com suas preferências de gênero.

Usuario

- **Atributos:**

- **nome:** String: Nome do usuário.
- **preferenciasGeneros:** Map<String, Boolean>: Um mapa que armazena as preferências de notificações do usuário, onde a chave é o gênero e o valor é um booleano indicando se as notificações estão ativadas para aquele gênero.

- **Métodos:**

- **Usuario(nome: String):** Construtor da classe, que inicializa o nome do usuário e suas preferências de gêneros.
- **ativarNotificacao(genero: String):** Ativa notificações para o gênero especificado.
- **desativarNotificacao(genero: String):** Desativa notificações para o gênero especificado.
- **update(conteudo: Conteudo):** Método chamado pela plataforma para notificar o usuário sobre um novo conteúdo. Verifica se o usuário está interessado no gênero e, em caso afirmativo, exibe a notificação.
- **getNome(): String:** Retorna o nome do usuário.

Resumo: A classe Usuario representa cada usuário da plataforma, permitindo que eles escolham receber notificações apenas para os gêneros que lhes interessam.

iObserver

- **Função:** Este é um ponto de referência para o padrão Observer, que define o contrato (interface) para os métodos que devem ser implementados pelos observadores (neste caso, os usuários).

Resumo: iObserver é o papel desempenhado pela interface que os observadores devem implementar para se inscreverem nas notificações de conteúdo.

Relações entre as Classes

- **Agregação:**

- A Plataforma tem uma agregação com Usuario, indicando que vários usuários estão inscritos nela.
- Plataforma também agrega uma lista de Conteudo, que representa os filmes e séries disponíveis na plataforma.

- **Herança/Implementação:**

- Usuario implementa os métodos definidos pelo iObserver, seguindo o padrão Observer.

UML:

