

Repositório público do Prof. Victor Machado

Material usado nas minhas disciplinas do IBMEC/RJ

[View My GitHub Profile](#)

IBM0516 - Programação Orientada a Objetos - Pedido da AP1

Pedido

Você recebeu um pedido de projeto para desenvolver um aplicativo para venda de ingressos online para eventos. O aplicativo deve permitir que os usuários comprem ingressos para os eventos e realizem o pagamento de forma online.

Funcionalidades do projeto

- O aplicativo deve permitir que o usuário visualize a lista de partidas disponíveis para venda;
- Para cada partida, o usuário deve poder escolher a quantidade e o tipo de ingressos desejados;
- O aplicativo deve calcular o preço total da compra de acordo com a quantidade e o tipo de ingressos selecionados;
- Após o pagamento, o usuário deve receber uma mensagem com as informações sobre a partida e a quantidade e o tipo de ingressos comprados.

Implementações necessárias

Para a primeira entrega, será necessário que você implemente o seguinte:

- **Evento:** classe abstrata que representa um evento. Deve conter os seguintes atributos e métodos:
 - *Atributo nome (String):* nome do evento;
 - *Atributo data (String):* data e horário do evento;
 - *Atributo local (String):* local onde será realizado o evento;
 - *Atributo ingressosInteira (int):* quantidade de ingressos tipo inteira disponíveis;
 - *Atributo ingressosMeia (int):* quantidade de ingressos tipo meia disponíveis;
 - *Atributo precoCheio (double):* valor do ingresso cheio para o evento, sem o desconto de meia entrada;

- *Método isIngressoDisponivel(TipoIngresso tipo, int quantidade)*: verifica se há ingressos disponíveis para o tipo e quantidade especificados;
- *Método venderIngresso(TipoIngresso tipo)*: realiza a venda de ingressos do tipo e quantidade especificados e retorna o valor total da venda;
- Implemente os métodos getters e setters conforme a necessidade;
- **TipoIngresso**: enumeração que representa os tipos de ingresso disponíveis. Deve conter os seguintes valores:
 - *INTEIRA*: para ingressos inteiros;
 - *MEIA*: para ingressos com desconto de meia-entrada.
- **Ingresso**: classe abstrata que representa um ingresso vendido. Deve conter os seguintes atributos e métodos:
 - *Atributo evento (Evento)*: evento para o qual o ingresso foi vendido;
 - *Atributo tipo (TipoIngresso)*: indica se é inteira ou meia entrada;
 - *Método getPreco()*: retorna o preço a depender do tipo de ingresso.

Além desses requisitos básicos, foi selecionado que você evolua o projeto, incluindo subclasses das classes **Evento** e **Ingresso**. Como eventos que podem ser registrados no sistema, é possível citar:

- **Shows**: o evento deve conter o nome do artista e o gênero da música. O ingresso precisa ter qual a localização do ingresso dentro do local do evento, podendo ser "pista" ou "camarote";
- **Exposição**: o evento deve conter a faixa etária mínima recomendada e a duração, em dias, da exposição. O ingresso deve possuir um novo atributo booleano, chamado *descontoSocial*. Caso esse valor seja `true`, o preço de venda do ingresso é zerado;
- **Jogo**: o evento deve conter o esporte e as equipes que estão competindo. O ingresso possui um atributo *percentualDescontoTorcedor*, que é aplicado ao valor do ingresso no cálculo do preço. Por exemplo, se o desconto de torcedor for 15% e o ingresso foi meia entrada, com um preço cheio de R\$100, o preço final fica $0.85 * 0.5 * 100 = R\$42.50$.

Para a criação de enumerações, consulte [este material online](#).

Menu de interface

Para a primeira entrega, foi solicitado que você elabore uma interface por linha de comando (CLI), de forma que o usuário possa realizar as seguintes operações:

- Cadastre um novo evento;
- Realize a venda de um ingresso, em que o usuário escolhe o tipo do ingresso (inteira ou meia) e apresenta as características particulares do ingresso (desconto social, por exemplo), e o programa gera um ingresso e exibe na

tela as informações. O usuário deve confirmar se as informações estão corretas e o programa deve sinalizar que a compra foi realizada;

- Exibir informações do evento;
- Exibir o número de ingressos restantes;
- Exibir informações do último ingresso vendido;

O programa só precisa armazenar um evento e um ingresso por vez. Ou seja, se um novo evento for cadastrado, o evento anterior é apagado.

Avaliação

O projeto será avaliado de acordo com os seguintes critérios:

- Qualidade de escrita do código: variáveis foram bem nomeadas, código está de acordo com os padrões discutidos em sala (*casing*, indentação, espaçamento, etc.);
- Qualidade da implementação: o código não apresenta redundâncias ou acoplamento excessivo. O projeto está bem organizado e os métodos estão implementados nas classes adequadas;
- Qualidade da interface de texto: a interface está organizada, as opções estão claras e não há confusão nas informações de dados;
- Testes realizados pelo professor não apresentam falhas.

Informações gerais:

- Prazo: 13/10/2023;
- Grupos de no mínimo 3 e no máximo 5 pessoas;
- Entrega via Sala de Aula Virtual (estudante.ibmec.br);
- Submeter pelo estudante um link para um repositório privado específico para o projeto, adicionando o professor (@victor0machado) como colaborador. O repositório precisa ter o nome de todos os participantes.