

## EXERCÍCIO PROGRAMA 1 – VENDA DE INGRESSOS DE TEATRO

Sua tarefa é implementar um conjunto de classes em Java (utilizando o conteúdo ministrado em aula) para controlar a venda de ingressos para o “Teatro Lima Nunes”. O teatro é uma importante casa de espetáculos que disponibiliza quatro setores de assentos, com localização e preços diferenciados: camarote, frente, meio e fundo. As classes implementadas devem ser capazes de executar todo o processo de vendas de ingressos, incluindo: abertura diária do caixa, definição da quantidade de locais em cada setor, verificação de locais disponíveis quando um cliente deseja comprar ingressos, cálculo de troco adequado aos clientes de acordo com as regras do teatro e exibição de informações (relatórios) sobre os dados atuais, quando solicitados.

A seguir são definidas as classes e alguns métodos obrigatórios. Você deve escolher o tipo de retorno e os tipos de parâmetros de forma adequada para cada método, assim como o tipo adequado para cada atributo das classes.

**Atenção: seu programa será verificado obedecendo aos nomes das classes e métodos fornecidos. Portanto, observe atentamente as definições.** Se necessário (e provavelmente será), crie outros métodos (adicionais) não definidos aqui. Escolha quais variáveis devem ser atributos dessas classes.

### Classe Caixa:

Esta classe modela o fluxo de caixa do teatro. Tem como tarefa gerenciar a quantidade de notas de 1, 5, 10, 20, 50 e 100 reais. Para isso, deve receber pagamentos e fornecer trocos.

A classe Caixa deve implementar pelo menos os seguintes métodos:

- **recebePagamento (notas1, notas5, notas10, notas20, notas50, notas100):** onde **notasX** é quantidade de notas X recebidas (X sendo 1, 5, 10, 20, 50 ou 100).
- **forneceTroco(valor):** se há o valor do troco pedido no caixa, retira do caixa as notas necessárias para o troco (decrementa os contadores das notas) e retorna o valor 1, que significará que o troco foi feito com sucesso. O cálculo do troco deve sempre dar preferência para as notas de valores mais altos. Se não houver o valor do troco no caixa, então exibe mensagem informando o fato e retorna o valor zero, que significará que houve falha no troco.

### Classe Setor:

Esta classe modela cada setor do teatro. Tem a tarefa de gerenciar os lugares disponíveis.

A classe Setor deve implementar pelo menos os seguintes métodos:

- **abreSetor(codigo, capacidade, preco):** onde **codigo** é um número inteiro qualquer que identifica o setor, **capacidade** é a quantidade de lugares que este setor comporta, e **preco** é o preço unitário (de um lugar) neste setor.
- **reservaLugar(quantidade):** reserva **quantidade** de lugares neste setor (isto é, decrementa essa quantidade do número de lugares disponíveis) se houver essa quantidade de lugares disponível. Senão, emite mensagem informando tal fato. Retorna o número de lugares reservados.
- **capacidadeSetor():** retorna a capacidade do setor.
- **disponibilidadeSetor():** retorna o número de lugares disponíveis no setor.

## Classe Vendedor

Esta classe modela uma entidade (real ou virtual) responsável pela venda de ingressos e também pelo teatro. É o vendedor quem deve abrir o teatro e o caixa. Para tanto, esta classe deve interagir com o usuário e com as classes Caixa e Setor. Além disso, é responsável por gerar relatórios da situação atual do teatro (setores) e do caixa.

A classe Vendedor deve implementar pelo menos os seguintes métodos:

- **abreCaixa(notas1, notas5, notas10, notas20, notas50, notas100):** onde **notasX** é a quantidade de notas X (X sendo 1, 5, 10, 20, 50 ou 100) que o caixa possuirá no início do dia. Este método deve instanciar um objeto da classe Caixa e chamar métodos para inicializar seus atributos.
- **abreTeatro(codigoCamarote, capacidadeCamarote, precoCamarote, codigoFrente, capacidadeFrente, precoFrente, codigoMeio, capacidadeMeio, precoMeio, codigoFundo, capacidadeFundo, precoFundo):** cria os 4 setores (camarote, frente, meio e fundo) e inicializa seus atributos. Para a criação dos setores devem ser instanciados objetos da Classe Setor.
- **vendeIngresso(quantidade, codigoSetor, notas1, notas5, notas10, notas20, notas50, notas100):** onde **quantidade** é o número solicitado de ingressos, **codigoSetor** é o código do setor que o comprador (usuário do sistema) deseja e **notasX** é o número de notas X (X sendo 1, 5, 10, 20, 50 ou 100) que o comprador (usuário do sistema) forneceu para pagar os ingressos solicitados. Neste método, o vendedor deve conferir se o dinheiro recebido é suficiente para pagar os ingressos e se há a quantidade solicitada de lugares disponível no setor especificado. Se as duas condições não forem verdadeiras, exibe mensagem informando o fato (conforme a condição que falhou) e NÃO faz a venda. Deve ainda verificar se o usuário pagou além do valor necessário e, neste caso, calcular o valor do troco e pedir para o caixa devolver o troco devido. Caso o caixa não possua dinheiro para fornecer o troco, deve exibir uma mensagem informando o fato e NÃO efetua a venda. Se houver lugar no setor solicitado, se o dinheiro pago for suficiente, e se a devolução de troco for realizada com sucesso, então reserva **quantidade** lugares no setor **codigoSetor** e exibe mensagem informando que a venda foi realizada com sucesso. NOTE: se a venda não for realizada com sucesso mas o pagamento foi feito, o dinheiro pago precisa ser devolvido! (isto é, precisa decrementar os contadores das notas do caixa).
- **geraRelatorio():** deve exibir a situação atual do teatro e do caixa, com as seguintes informações: código, capacidade e número de lugares disponíveis de cada setor; valor em dinheiro existente no momento da abertura do caixa, valor em dinheiro no atual momento (valor total e quantidade de cada tipo de nota) e quanto faturou até o momento; quantidade total de ingressos vendidos. O *layout*

do relatório deve seguir o modelo abaixo, inclusive as linhas em branco (campos definidos entre “<” e “>” são variáveis):

***Layout para o relatório:***

**RELATÓRIO PARCIAL DO TEATRO:**

**SETOR CAMAROTE:**

Código: <código do setor>

Capacidade: <capacidade do setor>

Lugares disponíveis: <número de lugares disponíveis do setor>

**SETOR FRENTE:**

Código: <código do setor>

Capacidade: <capacidade do setor>

Lugares disponíveis: <número de lugares disponíveis do setor>

**SETOR MEIO:**

Código: <código do setor>

Capacidade: <capacidade do setor>

Lugares disponíveis: <número de lugares disponíveis do setor>

**SETOR FUNDO:**

Código: <código do setor>

Capacidade: <capacidade do setor>

Lugares disponíveis: <número de lugares disponíveis do setor>

**CAIXA:**

Valor inicial: <valor inicial>

Valor atual: <valor atual>

Número de notas de R\$ 1,00: <notas1>

Número de notas de R\$ 5,00: <notas5>

Número de notas de R\$ 10,00: <notas10>

Número de notas de R\$ 20,00: <notas20>

Número de notas de R\$ 50,00: <notas50>

Número de notas de R\$ 100,00: <notas100>

Faturamento: <valor faturado até o momento>

Quantidade total de ingressos vendidos: <quantidade total de ingressos vendidos>

**OBSERVAÇÕES SOBRE A ENTREGA DOS TRABALHOS:**

- Os critérios de correção utilizados serão: corretude (resultados corretos), clareza e documentação do código fonte.
- Deverão ser postados no sistema COL os arquivos .java e .class.
- Os arquivos devem ser compactados em um único arquivo com o nome nrUSP\_nomeCompletoSemEspaçosMasComPrimeirasLetrasEmMaiusculoEstiloJava.zip.
- Somente o arquivo ZIP deve ser postado no sistema COL.