

Programação Orientada a Objetos - POOS3

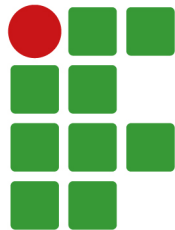
1

Tecnologia em Análise e Desenvolvimento de Sistemas

Laboratório 1

Implementação de sistema considerando conceitos de herança, polimorfismo, classe abstrata e array.

2º semestre de 2018



INSTITUTO FEDERAL

São Paulo

Câmpus Araraquara



Problema

- Devemos implementar um sistema para o Teatro Municipal de Araraquara que armazenará as vendas de ingressos de um espetáculo.
- **Observações:**
 - As cadeiras do teatro são organizadas em fileiras, assim, cada cadeira possui o número da fileira e o número da cadeira.
 - Cada ingresso vendido é associado a uma única cadeira, e a mesma cadeira não pode ser vendida mais de uma vez.
 - Existem dois tipos de ingresso: Normal e Meia Entrada, sendo que a diferença entre eles é o preço pago.
 - Deve-se registrar a venda de todos os ingressos e ao final das vendas informar o faturamento total do espetáculo.

Projeto

- Crie no Eclipse um novo projeto denominado:
 - “Laboratorio1_Teatro”
- Crie os pacotes:
 - model
 - view

Cadeira

- Observa-se que cada cadeira do teatro possui uma identificação:
 - Fileira
 - Assento
- Além disso, observa-se que cada cadeira possui um estado:
 - Ocupada ou Não Ocupada

```
package model;

public class Cadeira {
    private int fileira;
    private int assento;
    private boolean ocupada;

    public Cadeira(int fileira, int assento) {
        this.fileira = fileira;
        this.assento = assento;
        this.ocupada = false;
    }

    public void ocupar() {
        this.ocupada = true;
    }

    public boolean estaLivres() {
        return this.ocupada == false? true : false;
    }

    public int getAssento() {
        return assento;
    }

    public int getFileira() {
        return fileira;
    }
}
```

Para instanciar uma cadeira é necessário definir sua identificação. Sempre que criada, uma cadeira não estará ocupada.

Durante as vendas uma cadeira poderá ser ocupada a qualquer momento.

Antes de vender uma cadeira é preciso saber se a mesma está ou não livre.

Teatro

- Sabe-se que as cadeiras estão fixas em um ambiente, o Teatro.
- Assim, o Teatro conterá todos os assentos/cadeiras que poderão ser vendidos para o espetáculo.

```
package model;

public class Teatro {
    private final int fileiras = 10;
    private final int cadeiras = 20;
    private Cadeira[] assentos;

    public Teatro() {
        int posicao;

        assentos = new Cadeira[tamanhoTeatro()];

        posicao = 0;
        for(int i = 1; i <= fileiras; i++) {
            for(int j = 1; j <= cadeiras; j++) {
                assentos[posicao] = new Cadeira(i, j);
                posicao += 1;
            }
        }
    }
}
```

O Teatro possui quantidade fixa de cadeiras disponibilizadas em fileiras. Assim, definiu-se a quantidade de fileiras e q quantidade de cadeiras por fileira como constantes de Teatro.

Ao construir o teatro, são instanciadas todas as cadeiras do teatro, indicando sua identificação.

Vender Cadeira do Teatro

```
public boolean vender(int fila, int assento) {  
    Cadeira cadeira;  
    boolean deuCerto = false;  
  
    cadeira = buscaCadeira(fila, assento);  
    if(cadeira.estaLivre()) {  
        cadeira.ocupar();  
        deuCerto = true;  
    }  
    return deuCerto;  
}
```

Sempre que uma cadeira é vendida, ela deve ser configurada como ocupada.

Buscar a cadeira no Teatro

```
public Cadeira buscaCadeira(int fila, int assento) {  
    Cadeira cadeira = null;  
    int i = 0;  
  
    while(cadeira == null && i < tamanhoTeatro()) {  
        if(assentos[i].getFileira() == fila) {  
            if(assentos[i].getAssento() == assento) {  
                cadeira = assentos[i];  
            }  
        }  
        i++;  
    }  
    return cadeira;  
}
```

A cadeira deve existir no Teatro para que possa ser ocupada.

Ingressos

- O ingresso é vinculado a uma cadeira e pode ser de dois tipos: Inteira ou Meia entrada.
- Implementaremos uma classe abstrata Ingresso que será base para os dois tipos de entrada.

```
package model;

public abstract class Ingresso {
    protected final double precoNormal = 100;
    protected Cadeira cadeira;

    public Ingresso(Cadeira cadeira) {
        this.cadeira = cadeira;
    }

    public Cadeira getCadeira() {
        return cadeira;
    }

    public abstract double getValor();
}
```

Definimos a classe abstrata com dois argumentos, observe que o preço do ingresso é fixo.

Além disso, definiu-se o método abstrato `getValor()` que devolverá o valor do ingresso, podendo este retornar 100 ou 50.

```
package model;

public class Normal extends Ingresso{

    public Normal(Cadeira cadeira) {
        super(cadeira);
    }

    @Override
    public double getValor() {
        return this.precoNormal;
    }

}
```

```
package model;

public class Meia extends Ingresso{

    public Meia(Cadeira cadeira) {
        super(cadeira);
    }

    @Override
    public double getValor() {
        return (this.precoNormal / 2);
    }

}
```

Venda de ingressos

- Um ingresso é vendido após a informação da cadeira desejada e do tipo de ingresso que se deseja.
- Temos que considerar que nem sempre é possível vender uma cadeira pois essa pode estar ocupada.

```
package model;

public class Vendas {

    public static final int INTEIRA = 1;
    public static final int MEIA = 2;

    private Teatro teatro;
    private Ingresso[] ingressos;
    private int ultimaVenda;

    public Vendas() {
        this.teatro = new Teatro();
        ultimaVenda = -1;
        ingressos = new Ingresso[teatro.tamanhoTeatro()];
    }
}
```

Realizar venda

```
public boolean novaVenda(int fila, int assento, int tipoIngresso) {  
    boolean deuCerto = false;  
    Ingresso ingresso;  
    Cadeira cadeira;  
  
    if(teatro.vender(fila, assento)) {  
        cadeira = teatro.buscaCadeira(fila, assento);  
        if(tipoIngresso == INTEIRA)  
            ingresso = new Normal(cadeira);  
        else  
            ingresso = new Meia(cadeira);  
  
        ultimaVenda += 1;  
        ingressos[ultimaVenda] = ingresso;  
        deuCerto = true;  
    }  
    return deuCerto;  
}
```

Valor total das vendas

```
public double valorArrecadado() {  
    int i=0;  
    double soma = 0;  
  
    for(i=0; i<=ultimaVenda; i++) {  
        soma += ingressos[i].getValor();  
    }  
  
    return soma;  
}
```



```
package view;

import javax.swing.JOptionPane;
import model.Vendas;

public class Main {

    public static void main(String[] args) {
        Vendas vendas;
        int fileira;
        int assento;
        int inteira;
        boolean vendeu;

        vendas = new Vendas();

        do {
            JOptionPane.showMessageDialog(null, "Venda de ingressos", "Ingressos do Teatro P00",
                JOptionPane.INFORMATION_MESSAGE);

            fileira = Integer.parseInt(JOptionPane.showInputDialog("Fileira"));
            assento = Integer.parseInt(JOptionPane.showInputDialog("Cadeira"));
            inteira = JOptionPane.showConfirmDialog(null, "Ingresso preço cheio?",
                "Ingressos do Teatro P00", JOptionPane.YES_NO_OPTION);
        }
```

```

        if(inteira == JOptionPane.YES_OPTION) {
            vendeu = vendas.novaVenda(fileira, assento, Vendas.INTEIRA);
        } else {
            vendeu = vendas.novaVenda(fileira, assento, Vendas.MEIA);
        }

        if(!vendeu) {
            JOptionPane.showMessageDialog(null, "Cadeira ocupada", "Ingresso do Teatro P00",
                JOptionPane.ERROR_MESSAGE);
        } else {
            JOptionPane.showMessageDialog(null, "Bom espetáculo", "Ingresso do Teatro P00",
                JOptionPane.WARNING_MESSAGE);
        }

    } while(JOptionPane.showConfirmDialog(null, "Deseja encerrar o sistema?",
        "Ingressos do Teatro P00", JOptionPane.YES_NO_OPTION) == JOptionPane.NO_OPTION);

    JOptionPane.showMessageDialog(null, "Total de vendas: R$" + vendas.valorArrecadado(),
        "Teatro P00", JOptionPane.INFORMATION_MESSAGE);
}
}

```

Atividades

- O sistema possui um erro:
 - Ao digitar os dados de uma cadeira que não existe o sistema para de funcionar.
 - Resolva o problema descrito acima.
- Implemente uma solução que informe quantas cadeiras estão disponíveis no teatro antes de realizar uma venda.
- Implemente um novo tipo de ingresso: o Isento.

Entrega

- Criar um repositório no github.com com o projeto.
- No repositório deve-se incluir como colaborador o usuário: ednilsonrossi
- O projeto deve ser disponibilizado no github, o último commit deve ser realizado no mesmo dia da aula.
- Atenção: não deve-se carregar o projeto compactado no github e sim cada um dos arquivos usando comandos do git.
- O link do repositório deve ser postado no moodle.