Programação orientada a objeto
Aula 2

Fernando Duwe

v 1.0

Nesta aula veremos alguns conceitos teóricos de orientação a objeto, que nos ajudarão a entender melhor como as classes se relacionam e como podemos otimizar a criação de classes em nossa aplicação.

Os conceitos abordados serão:

- Acoplamento;
- Coesão;
- Associação;
- Agregação;
- Composição;

Acoplamento

Acoplamento é uma medida entre componentes, no caso de OO, uma medida entre o relacionamento de classes. Podemos definir como o grau de dependência entre classes;

Refere-se ao nível em que uma classe conhece ou usa membros de uma outra classe;

Os dois espectros dessa medida são:

- Baixo Acoplamento ou Acoplamento Fraco;
- Alto Acoplamento ou Acoplamento Forte ;

Baixo Acoplamento

Se o único conhecimento que a classe A tem sobre a classe B, é que a classe B foi exposta através de sua interface, então as classes A e B tem um baixo acoplamento;

Baixo acoplamento é o estado desejável para classes bem encapsuladas que minimizam as referências umas às outras.

Alto Acoplamento

Se a classe A se baseia em partes da classe B que não fazem parte da interface de B, então elas são bastante acopladas;

Em outras palavras, se A sabe mais do que deveria sobre a forma como B foi implementada, então A e B estão bastante ligadas;

O alto acoplamento é o estado indesejável de se ter classes que desobedecem às regras do baixo acoplamento.

Problemas do Alto Acoplamento

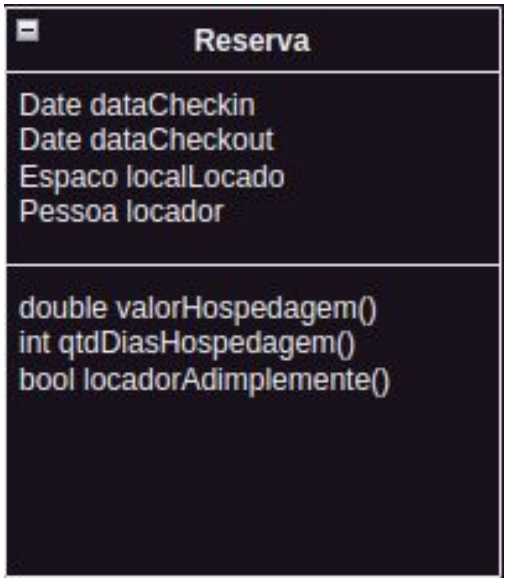
- Difícil entendimento: A classe é mais difícil de se entender isoladamente;
- Difícil reutilização: A classe é mais difícil de ser reusada, já que depende da presença de outras classes;
- Propagação de mudanças: Mudanças em uma classe relacionada força mudanças locais à classe;

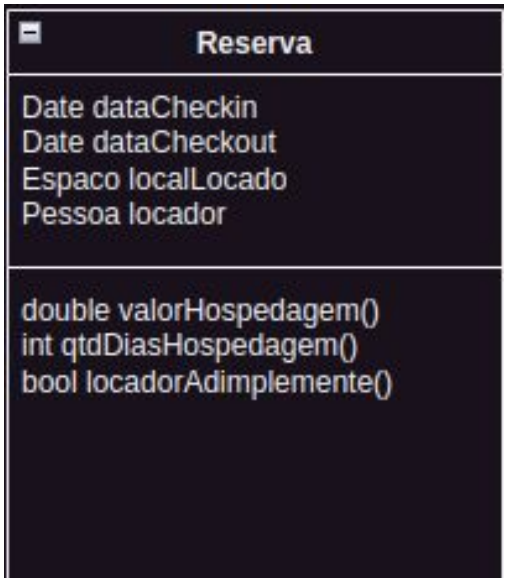
Coesão

Coesão está ligada ao princípio da responsabilidade única, que diz que uma classe deve ter apenas uma única responsabilidade e realizá-la de maneira satisfatória, ou seja, uma classe não deve assumir responsabilidades que não são suas.

Uma vez sendo ignorado este princípio, passamos a ter problemas, como dificuldades de manutenção e de reuso.

Ver mais


Reserva
Date dataCheckin
Date dataCheckout
Espaco localLocado
Pessoa locador
double valorHospedagem()
int qtdDiasHospedagem()
bool locadorAdimplemente()


Reserva
Date dataCheckin
Date dataCheckout
Espaco localLocado
Pessoa locador
double valorHospedagem()
int qtdDiasHospedagem()
bool locadorAdimplemente()


Pessoa
String nome
Date nascimento
bool locadorAdimplemente()


OOP

Associação

A associação descreve um vínculo que ocorre entre classes , sendo que a mais comum é a associação binária, mas é possível que uma classe esteja vinculada a si própria, e aí temos a associação unária; outro cenário é onde temos uma associação que seja compartilhada por mais de uma classe, o que conhecemos por associação ternária ou N-ária, sendo este tipo de associação a mais rara e também mais complexa.

Falamos sobre associação entre dois objetos quando cada um deles pode usar o outro, mas também cada um deles pode existir sem o outro. Não há dependência entre eles.

Agregação

Esse relacionamento é um tipo especial de associação onde as informações de um objeto (chamado objeto-todo) precisam ser complementadas pelas informações contidas em um ou mais objetos de outra classe (chamados objetos-parte); temos então o que conhecemos como todo/parte. A agregação é um tipo de composição que representa um vínculo fraco entre duas classes.

O relacionamento de agregação às vezes é referido como relacionamento "tem um" ou 'has a'.

Neste tipo de relacionamento, um objeto pode ser composto de um ou mais objetos na forma de suas propriedades.

Assim temos que :

- Todo Cliente tem um endereço para o qual o produto solicitado será enviado;
- Cada Pedido tem um cliente, um endereço de envio e um produto representado como um PedidoItem;

Podemos então concluir que nosso objeto da classe Pedido é composto pelos objetos Cliente, Endereco e PedidoItem.

O objeto PedidoItem é ainda composto pelo objeto Produto e a classe Pedido compõe esses objetos com suas propriedades.

Composição

O relacionamento Composição, representa um vínculo forte entre duas classes , e , é também um relacionamento caracterizado como parte/todo, mas, neste caso, o todo é responsável pelo ciclo de vida da parte. Assim a existência do Objeto-Parte NÃO faz sentido se o Objeto-Todo não existir.

No nosso exemplo o objeto da classe Pedido é composto por um Cliente e um PedidoItem. Se rompermos o relacionamento entre as classes Pedido e Cliente, a classe Cliente ainda vai poder existir, mas se a relação entre a classe Pedido e a classe PedidoItem for quebrada, a classe PedidoItem não pode existir.(um pedido é composto por um ou vários itens).

Suponha que a funcionalidade do nosso aplicativo mude no futuro e, em vez de aceitar pedidos de produtos, agora oferece alguns outros serviços aos clientes existentes, digamos um serviço de mensagens.

Nesse cenário, a classe Pedido não servirá para nada. No entanto, a classe Cliente que já foi composta pela classe Pedido ainda pode existir sem ela, já a classe PedidoItem não pode.

Exercício 6

Vamos criar um sistema de venda de ingressos em um clube. As seguintes classes devem ser criadas:

- Classe de Pessoa: Possuindo os dados básicos de uma pessoa;
- MembroClube: Classe filha de Pessoa, com os dados adicionais de um membro de um clube, como a matrícula por exemplo;
- VendaIngresso: Classe com os dados de venda, evento, data e comprador;

Somente membros do clube podem efetuar compras de ingressos.

Exercício 7

Vamos criar um aplicativo de cálculo de consumo de um veículo. O app terá as seguintes classes:

- CalculaConsumoMedia: terá por atributo a quantidade de combustível total, a quantidade de combustível atual e a quilometragem percorrida;
- Carro: Terá por atributo a capacidade do tanque, a quilometragem percorrida e a quantidade atual de combustível disponível;
- Na classe de carro, possuir os seguintes métodos: encherTanque() e calcularMedia();

Exercício 8

Nesse exercício vamos controlar um aeroporto. Nosso sistema será organizado da seguinte forma:

- Uma classe de Pessoa;
- Uma classe de Tripulante, filha de Pessoa;
- Uma classe de Passageiro, filha de Pessoa;
- Uma classe com os dados da aeronave, chamada de Avião;
- Uma classe que controlará o voo, chamada Voo;
- Uma classe que armazenará os passageiros de um voo, com o número de poltrona de cada um, chamada VooMembro;