

Relacionamento entre classes

Carlos Olarte

21 de Agosto de 2019

Classes

Classes

- Abstração para agrupar objetos comuns que têm o mesmo comportamento.
- Descrevem de maneira **abstrata** o comportamento dos objetos.

Objetos

Objetos são **instâncias** de uma classe:

- Encapsulam um **estado**
- Respondem às mensagens com a execução de um **método**.

Os objetos de uma mesma classe **compartem o comportamento** definido pela sua classe.

Nesta aula:

As classes **não existem sozinhas**. Em vez disso, elas têm **relacionamentos entre si** que descrevem como as objetos **interagem uns com os outros**.

Objetivos:

- Explicar como os objetos **se relacionam** em um sistema orientado a objetos.
- Identificar os diferentes **tipos de relacionamento** entre classes.
- Implementar relações entre classes em Python.

Interfaces e Encapsulamento

Membros privados de Classe

- Atributos: representando o **estado** que **não deve ser visível** para os usuários da classe.
- Métodos: usados internamente.

Membros Públicos

- Métodos que podem ser invocados **externamente**.
- Especificam um **contrato**: operações que o objeto pode e deve realizar

Alterações no estado de um objeto geralmente ocorrem através da **invocação de seus métodos**.

Encapsulamento

- Os dados de um objeto **não podem** ser acessados diretamente.
 - O relacionamento entre os objetos deve ocorrer por meio de **métodos públicos** (a **interface** da classe).
- A ideia: **Esconder os detalhes** de implementação.
 - As mudanças na implementação não devem afetar os usuários da classe.

Relacionamentos de Associação

- Indicam algum **relacionamento** significativo e de interesse entre objetos
- Especificam que objetos de uma classe estão **ligados** a objetos de outras classes.
- Exemplos:
 - ▶ Em um Estacionamento estão **estacionados** os Carros.
 - ▶ Um professor **ministra** várias disciplinas.
 - ▶ Uma mensagem **possui** um Remitente é um Destinatário.

Relacionamentos de Associação



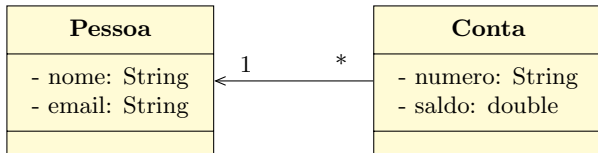
- Um objeto tipo A está ligado com zero o muitos objetos tipo B.
- Um objeto tipo B está ligado com um objeto tipo A.

Multiplicidade:

- 0: zero
- 0..1: zero ou um
- 1 : um
- 1..*: um ou muitos
- 0..* ou *: zero ou mais

Exemplo

Considere um banco: cada conta bancaria está relacionada com uma pessoa (o titular da conta):



Ver exemplo Conta - Pessoa

Agregação

- Representa uma relação **todo-parte**.
- As partes podem existir sem o todo.

Exemplos:

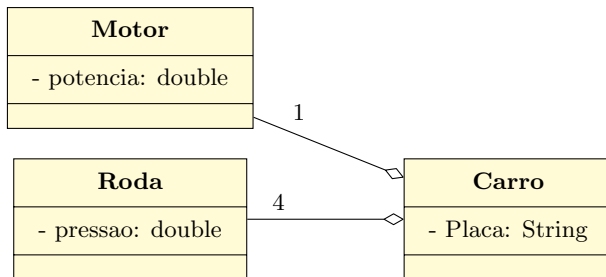
- Considere a relação entre o Carro, as Rodas e o Motor.
- Considere a relação entre um computador, o teclado, a tela, etc.



Agragação

Exemplo

Considere um carro: O carro possui um Motor e 4 rodas.



Ver exemplo Carro

Agregação na Prática

- Objetos do tipo Motor e Roda são instanciados.
- Estas instâncias são associadas a um objeto do tipo Carro no inicializador deste último.
- Ou seja, Motor, Roda e Carro podem ser instanciados independentemente.
- Quando a instância de Carro é removida da memória, as instâncias de Motor e Roda associadas continuam a existir

Composição

- Representa uma relação **todo-parte** mais “forte”.
- Se o objeto todo deixar de existir, os seus objetos parte deverão também deixar de existir.

Exemplos:

- Um estacionamento e seus andares.
- Um prédio e suas salas.

Composição



- A é o todo e B as partes.
- A existência de B não faz sentido se A não existir

Exemplo

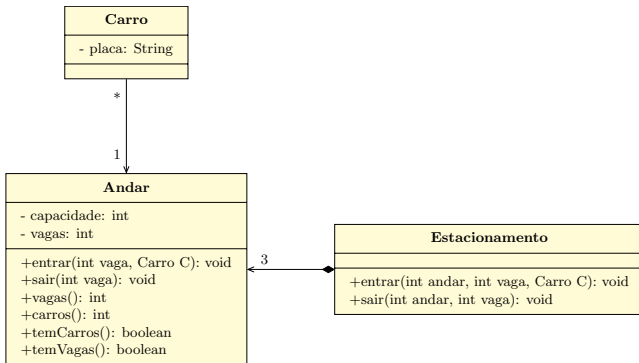
Ver exemplo Prédio -Sala

Relações entre objetos

- **Associação**: modela uma relação genérica entre duas classes. Nenhum objeto é considerado dono/proprietário do outro objeto
- **Agregação**: modela uma relação do tipo “tem um” fraca, ou seja, uma classe A é formada por objetos de outra classe B. Objetos da classe A são considerados donos/proprietários. Objetos da classe B podem existir sem que objetos da classe proprietária A existam
- **Composição**: modela uma relação do tipo “tem um” forte, ou seja, uma classe A é formada por objetos de outra B. Objetos da classe A são considerados donos/proprietários. Objetos da classe B deixam de existir quando objetos da classe proprietária A deixam de existir

Exercício

Vamos modelar um estacionamento com andares e controlar os carros dentro dele. O design proposto é:



Implemente as classes necessárias para este sistema.