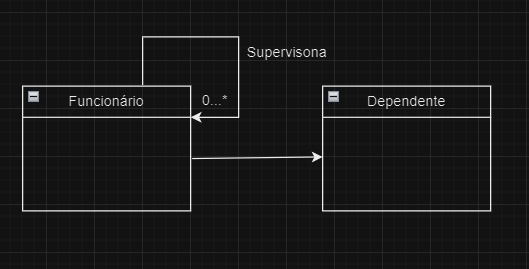
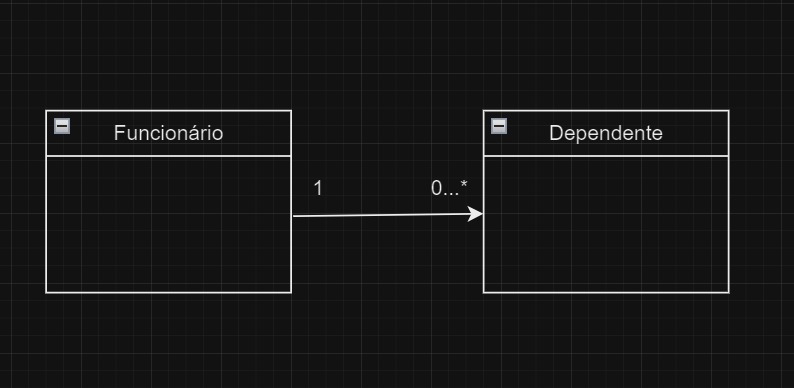
**Associação Unária (ou reflexiva)**

* Ocorre quando há um relacionamento de um objeto de uma classe com objetos da mesma classe



**Associação Binária:**

* Quando há duas classes envolvidas na associação de forma direta de uma para outra.



Public class Funcionario{

Private int matricula;

....

Private dependente[] dependentes;

//metodos

}

Public class Dependente {

Private String nome;

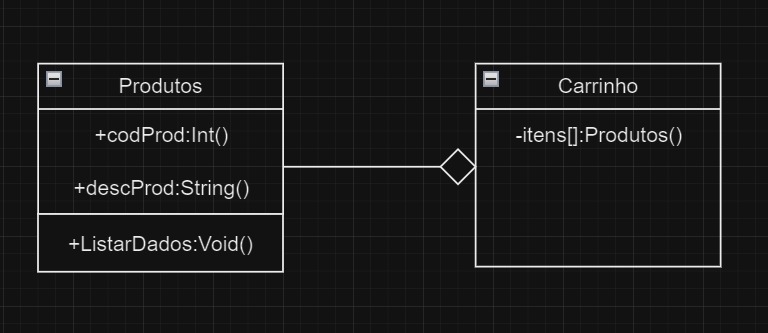
Private String parentesco;

Private Funcionario;

}

**Agregação:**

* Associação que tenta demonstrar as informações de um objeto todo, precisam ser complementadas por informações contidas em um objeto parte: ***Existência do objeto parte faz sentido mesmo não existindo o objeto todo.***

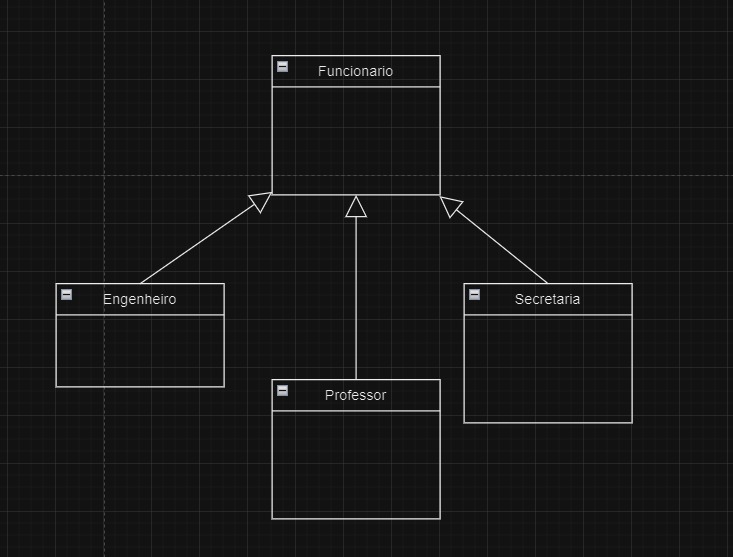


**Composição:**

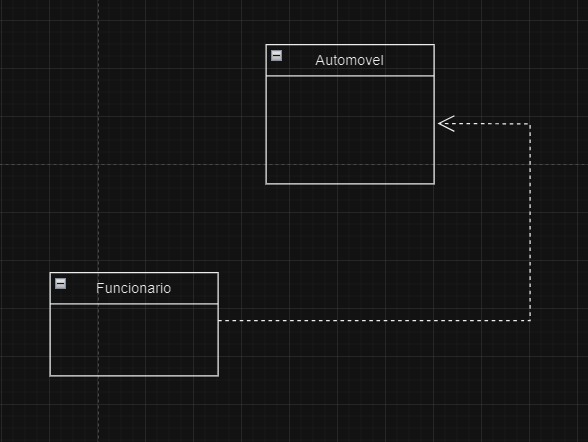
* Variação da agregação que é considerada mais forte, ou seja, o objeto parte não pode existir sem o objeto todo;

**Especialização/Generalização:**

* É a relação de herança (classe filho vai herdar os atributos da classe pai)



**Dependência:** Dependência de uma classe e outra, é diferente de uma associação porquê a conexão entre as classes é temporária.

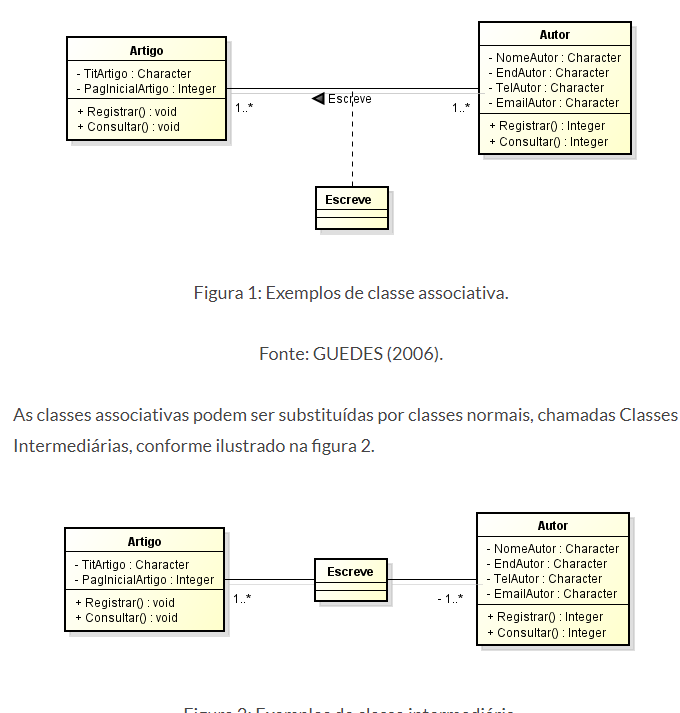


Obs: Funcionário não instância um automóvel, apenas usa como parâmetro de um método.

**CLASSE INTERMEDIÁRIA**: Substitui a associativa e porem tem seus próprios atributos (é igual a associativa porém tem seus atributos)

**CLASSE ASSOCIATIVA**:

* Classes associativas quando a associação entre duas classes possui multiplicidade muitos (\*) em ambas as extremidades. Esta classe é necessária para armazenar atributos transmitidos pela associação de ambas as classes, o que não impede que a classe associativa possua atributos próprios. A figura 1 apresenta um exemplo. De muitos para muitos.



**RESTRIÇÃO**: Quando tem um campo como se fosse observações composto por chaves {} onde dentro das chaves ficam o texto de restrição.

**INTERFACE**: É a interação usuário -> software