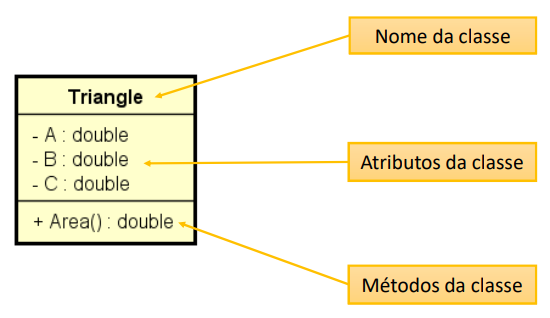
**- Linguagem UML**

A Linguagem de Modelagem Unificada (UML) foi criada para estabelecer uma linguagem de modelagem visual comum, semanticamente e sintaticamente rica, para arquitetura, design e implementação de sistemas de software complexos, tanto estruturalmente quanto para comportamentos.

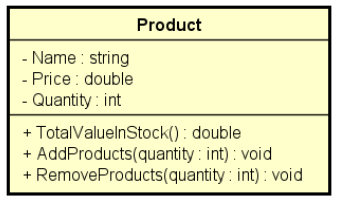
A UML não é uma linguagem de programação, ela mostra as classes (com seus atributos e métodos) e seus relacionamentos com outras classes, tudo em forma de desenho.

Podemos dizer que a UML mostra o projeto da classe ou diagrama de classes.

**- Estrutura padrão de um diagrama de classe UML**



**- Diagrama UML de uma classe “comum”**



Detalhes:

O nome da classe é escrito em negrito e com alinhamento centralizado.

Temos modificadores de acesso igual ao Java:

"-" para private

"#" para protected

"+" para public

"~" para default

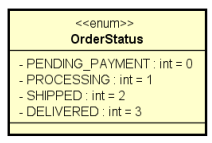
Os atributos possuem alinhamento a esquerda na seguinte estrutura:

<modificador de acesso> <nome da variável>: <tipo da variável>

Os métodos possuem alinhamento a esquerda na seguinte estrutura:

<modificador de acesso> <nome do método>(<nome do parâmetro>: <tipo do parâmetro>): <tipo da variável de retorno>

**- Diagrama UML de uma classe “enum”**



Detalhes:

Inicialmente colocamos a descrição "<<enum>>"

O nome da enum é escrito em negrito e com alinhamento centralizado.

É utilizado o indicador "private" para cada constante.

As constantes possuem alinhamento a esquerda na seguinte estrutura:

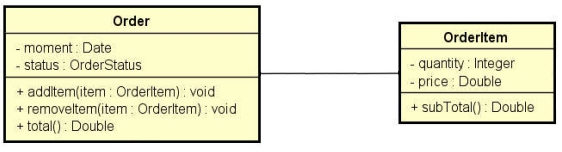
"<modificador de acesso> <nome da constante em maiúsculo>: int = <valor de posição>"

**- Relacionamentos**

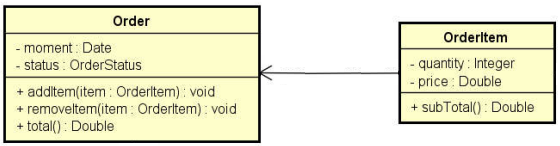
São as ligações entre os elementos dos modelos UML. Os elementos estão ligados uns aos outros, especificando o que cada elemento significa ao outro e qual o grau de ligação deles, ou seja, qual a relação lógica existe entre os elementos. A estas ligações damos o nome de Relacionamentos.

**- Tipos de relacionamentos**

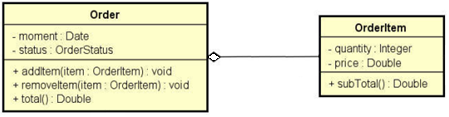
Associação - Representa uma ligação entre dois elementos.



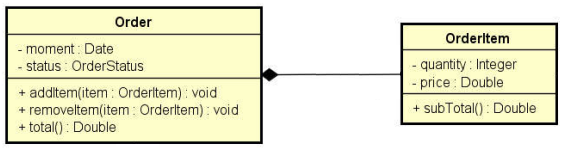
Herança ou Generalização - É um tipo de relacionamento onde a classe generalizada (onde a "ponta da seta" do conector fica) fornece recursos para a classe especializada (herdeira). Excetuando conceitos mais avançados (como padrões de projeto, interfaces, visibilidades específicas etc.), tudo que a classe mãe (generalizada) tem, a filha (especializada) terá.



Agregação - É um tipo de relacionamento onde a classe agregada usa outras classes para "existir", mas pode viver sem ela. Por exemplo, a classe "CorpoHumano" possui uma agregação com a classe "Mao". Sem a "Mao" a classe "CorpoHumano" pode existir.



Composição - É um tipo de relacionamento onde a classe composta depende de outras classes para "existir". Por exemplo, a classe "CorpoHumano" possui uma composição com a classe "Coracao". Sem a classe "Coracao", a classe "CorpoHumano" não pode existir.



**- Multiplicidade**

A multiplicidade nada mais é do que os limites inferior e superior da quantidade de objetos aos quais outro objeto está associado. Esses limites podem receber os valores: apenas um; zero ou muitos; um ou muitos; zero ou um; ou até um intervalo específico. Segue a tabela abaixo com a notação de cada limite no diagrama.

Tipos Significa

0..1 Zero ou uma instância. A notação n..m indica n para m instâncias

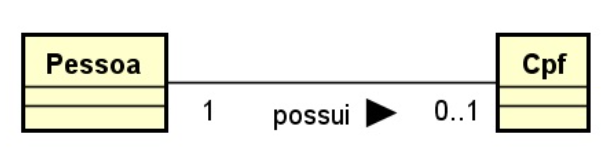
0..\* ou \* Não existe limite para o número de instâncias

1 Exatamente uma instância

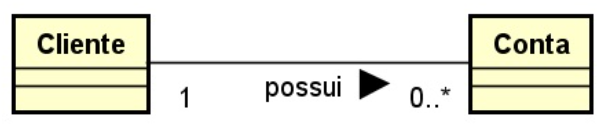
1..\* Ao menos uma instância

Na UML, as associações permitem representar essa informação nos extremos da sua linha.

Um para Um:



Um para Muitos:



Muitos para Muitos:

