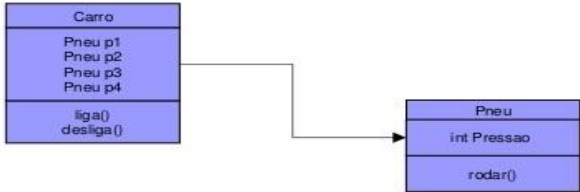
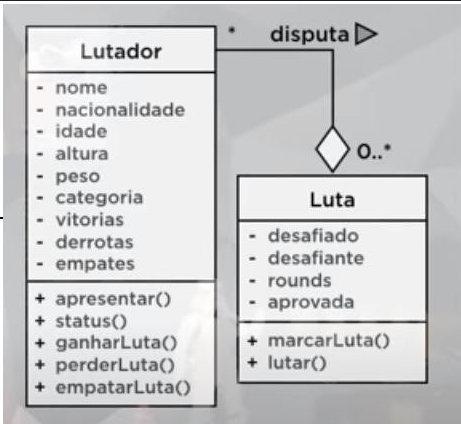


RELACIONAMENTOS EM POO	RELACIONAMENTOS EM POO
O que são os <b>Relacionamentos</b> em POO?	<p><b>Relacionamentos:</b> ocorrem quando os elementos de uma classe se relacionam com os elementos de outra classe. Na POO temos 3 tipos diferentes de relacionamentos dependendo do seu nível de entrelaçamento e grau de impacto sobre os elementos das classes relacionadas. Esses relacionamentos são:</p> <ul style="list-style-type: none"><li>- <b>Associação</b> (Relacionamento Fraco - Muitos para Muitos);</li><li>- <b>Agregação</b> (Relacionamento Médio - 1 para Muitos);</li><li>- <b>Composição</b> (Relacionamento Forte - Relacionamento de Morte);</li></ul> <p><b>OBS: Cuidado para não confundir com o conceito de Herança! Relacionamentos e Herança são conceitos diferentes.</b></p>
O que é o conceito de <b>Associação</b> ?	<p><b>Associação:</b> Ocorre quando temos classes completamente independentes entre si mas que eventualmente estarão relacionadas. A Associação é considerada um relacionamento muitos para muitos, onde não há herança e nem dependência entre elas, sua relação será totalmente eventual. Podemos exemplificá-la pela relação que existe entre professores e alunos, 1 aluno pode ter vários professores e 1 professor pode ter vários alunos. Um não depende do outro para existir. Professores podem existir sem alunos e e alunos podem existir sem professores, mas eventualmente eles terão de trabalhar juntos.</p>
Como podemos <b>representar uma Associação com ferramentas UML</b> ?	<p>Uma Associação deve ser representada pelos seguintes elementos:</p> <ul style="list-style-type: none"><li>- <b>Relaciomento:</b> uma seta que direciona para a classe atribuindo valores associados;</li><li>- <b>Grau de Relacionamento:</b> é sempre muitos para muitos;</li></ul> <p><b>OBS: esse é o grau de relacionamentos mais fraco, onde nenhum elemento depende do outro, esse é o tipo de relacionamento menos utilizado.</b></p>
Ilustre o uso das ferramentas UML para ilustrar uma Associação	
Como a Associação <b>pode ser realizada no momento do desenvolvimento do código</b> ?	<p><b>Primeiro:</b> temos que ter 2 classes, onde uma será a Classe Abstraída - aquela que irá construir as instâncias que serão associadas - e a outra será a Classe associadora - aquela que irá coletar os objetos da outra classe;</p> <p><b>Segundo:</b> vamos atribuir o elemento que desejamos associar a uma variável, depois vamos atribuir essa variável ao elemento da classe associadora;</p> <p><b>Terceiro:</b> qualquer modificação acontecerá na variável, os elementos das classes são totalmente independentes;</p>
O que é o conceito de <b>Agregação</b> ?	<p><b>Agregação:</b> É o como chamamos um relacionamento entre duas ou mais classes, onde é possível que uma classe possa utilizar as instâncias da outra classe atribuindo essas instâncias sobre os próprios objetos gerados na segunda classe e gerendo operações sobre essas instâncias. Quando um objeto recebe a instância de outra classe temos uma atribuição do tipo: "<b>Tem 1</b>", onde um objeto de uma classe contém outro objeto de outra classe. Esse tipo de dado recebe o nome de "<b>Tipo Abstrato</b>", pois ele é justamente objeto abstraído de uma classe, por isso recebe esse nome.</p>
Como podemos <b>representar uma Agregação com ferramentas UML</b> ?	<p>Uma Agregação deve ser representada pelos seguintes elementos:</p> <ul style="list-style-type: none"><li>- <b>Relaciomento:</b> uma linha com um losângulo na ponta onde a agregação está sendo recebida;</li><li>- <b>Multiplicidade:</b> semelhante as cardinalidades em bancos de dados, a multiplicidade demonstra o número de interações que pode haver entre uma classe e outra para aquele tipo de relacionamento. Dentre esses números temos: <b>0..*</b> (nenhum ou muitos)      <b>1</b> (1 relacionamento de ambas as partes) <b>1..*</b> (1 ou muitos)      <b>*</b> (muitos relacionamentos de ambas as partes);</li></ul>
Ilustre o uso das ferramentas UML para ilustrar uma agregação	
- reservada para a questão acima -	
Como a Agregação <b>pode ser realizada no momento do desenvolvimento do código</b> ?	<p><b>Primeiro:</b> temos que ter 2 classes, onde uma será a Classe Abstraída - aquela que irá construir as instâncias que serão agregadas - e a outra será a Classe Agregadora - aquela que irá agregar as instâncias da outra classe, geralmente dentro de um vetor;</p> <p><b>Segundo:</b> devemos criar um método de inserção dos tipos abstratos para a classe agregadora;</p> <p><b>Terceiro:</b> depois de atribuídos sobre a classe agregadora, é só utilizar os métodos sobre as instâncias;</p>

