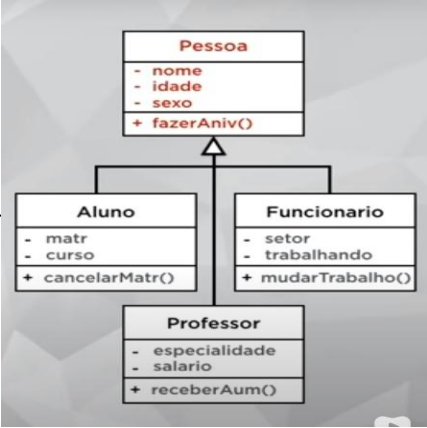
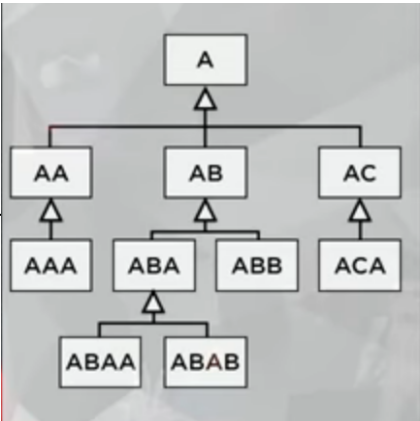


PILARES DA POO: HERANÇA		PILARES DA POO: HERANÇA	
O que é a Herança?		<p><b>Herança:</b> é um dos conceitos que formam os pilares da POO, a Herança é o conceito onde uma classe pode herdar atributos e métodos de outra classe. Onde nós teremos uma "<b>Super Classe</b>" (que pode ser chamada de: "<b>Classe Mãe</b>" ou "<b>Progenitora</b>"), que será a classe que vai compartilhar seus elementos com outra classe. Temos também a "<b>Sub Classe</b>" (também chamada de "<b>Classe Filha</b>"), que é a classe que herda os elementos da Super Classe e poderá compartilhar de todas as funcionalidades da Classe Mãe.</p>	
Quando devemos usar a Herança?		<p>A herança deve ser usada <b>quando temos várias classes que compartilham os mesmos elementos em comum: os mesmos atributos e talvez até os mesmos métodos</b>. Por exemplo imagine que temos a aplicação para uma escola onde existem as entidades: aluno, professor e funcionários, embora cada uma destas entidades tenham seus atributos e métodos muito particulares, todos eles vão compartilhar algumas características semelhantes, como: nome, sexo, idade e etc. Em casos como esses podemos usar o conceito de herança, onde criamos uma Super Classe "Pessoa" contendo os atributos e métodos semelhantes que serão atribuídos a cada entidade distinta.</p>	
Ilustre como podemos representar o conceito de Herança em UML			
- reservada para a questão acima -			
O que é uma Árvore de Herança?		<p><b>Árvore de Herança:</b> é um termo usado para identificar a estrutura hierarquica pela qual as heranças são formadas. Já que toda herança parte de uma <b>Super Classe Raiz</b>, todas as derivadas dela serão as Sub Classes, onde poderão ser geradas novas derivações a partir das Sub Classes já existentes, transformando Sub Classes e Super Classes das Sub Classes derivadas a partir delas, e as Super Classes anteriores se tornam <b>Classes Ancestrais</b> das novas Sub Classes. Isso acontece quando temos uma <b>Multiderivação de Classes</b>.</p>	
Ilustre uma Árvore de Herança Multiderivada			
- reservada para a questão acima -			
Quais são as <b>Nomenclaturas</b> usadas numa árvore de classes?		<p>- <b>Raiz:</b> É a Super Classe que está no topo da Hierarquia, cujas Classes derivadas vão todas herdar seus elementos (ou alguns deles dependendo do método de acesso) a partir dela. Essa é a Super Classe que não herda nada de Classe nenhuma;</p> <p>- <b>Super Classe:</b> São todas as Classes que possuem herança sobre Sub Classes, não importa se elas também são herdeiras, se elas tiverem Sub Classes, elas são Super Classes;</p> <p>- <b>Sub Classe:</b> São todas as Classes que herdam elementos de uma outra Classe, mesmo Super Classes podem ser Sub Classes desde que não sejam a Classe Raiz;</p> <p>- <b>Classe Ancestral:</b> São as Super Classes cuja Sub Classe derivou outra Sub Classe, para todas as Sub Classes geradas a partir da 1ª Sub Classe dela essa Super Classe será uma Classe Ancestral para elas;</p> <p>- <b>Classe Descendente:</b> São as Sub Classes cuja Super Classe possui uma Super Classe Progenitora, por herdar os elementos daquela Super Classe Progenitora, essas Classes são consideradas Classes Descendentes da Progenitora;</p> <p>- <b>Classe Folha:</b> São as Sub Classes no ramo mais baixo da árvore de herança, elas não possuem nenhuma Sub Classe herdando os elementos delas;</p> <p>- <b>Generalização:</b> É o nome dado as Classes que são superiores as Sub Classes, toda Classe superior de uma Sub Classe é a Generalização daquela Sub Classe;</p> <p>- <b>Especialização:</b> É o nome dado as Classes que são inferiores as Super Classes, toda Classe inferior é uma Super Classe é a Especialização daquela Super Classe;</p>	
- reservada para a questão acima -			
- reservada para a questão acima -			

PILARES DA POO: HERANÇA		PILARES DA POO: HERANÇA	
Ilustre uma <b>Árvore de Herança Multiderivada</b> com suas Nomenclaturas:			
- reservada para a questão acima -			
- reservada para a questão acima -			
O que são <b>Tipos de Herança</b> ? E quais são alguns deles?		<p>Tipos de Herança são as formas como uma Classe é herda elementos á partir de outra, dentre elas temos:</p> <ul style="list-style-type: none"><li>- <b>Herança por Implementação:</b> É o nome dado a todos os elementos herdados á partir de uma Super Classe;</li><li>- <b>Herança para Diferença:</b> É o nome dados os elementos novos, iniciados na própria Sub Classe, que quando acrescentados aos elementos herdados da Super Classe irão compor todos os elementos da Sub Classe;</li><li>- <b>Herança por Multiderivação Direta:</b> É a herança que existe entre a Super Classe a sua Sub Classe.</li><li>- <b>Herança por Multiderivação Indireta:</b> É a herança que existe entre a Classe Descendente e a sua Classe Ancestral;</li></ul>	
- reservada para a questão acima -			
Que cuidado devemos ter referente a <b>instancialização</b> quando trabalhamos com Herança?		<p>Devemos nos lembrar que a grande sacada da Herança é fazer com que as Classes Especializadas possam gerar instâncias específicas aproveitando os elementos das Classes Generalizadoras que poderão ser compartilhados entre as Classes derivadas. Então, por convenção - e as vezes forçadamente - <b>as linguagens POO priorizam a criação de instâncias diretamente sobre as Classes Especializadas</b>. Para isso foram criados os conceitos de Classes: Abstratas, Concretas e Finais e Métodos: Abstratos, Sobrescritos e Finais.</p>	
Explique o conceito de <b>Classes: Abstratas, Concretas e Finais</b> e <b>Métodos: Abstratos, Sobrescritos e Finais</b> .		<ul style="list-style-type: none"><li>- <b>Classe Abstrata:</b> São as Classes que não podem ser instanciadas - algumas linguagens de programação inclusive não permitem a instancialização sobre classes abstratas - o objetivo dessas classes é somente compartilhar seus atributos e seus métodos abstratos com as demais, toda Super Classe Raiz é automaticamente uma Classe Abstrata, mas podemos ter mais Classes Abstratas dentro da nossa árvore de herança. Para isso, o que define uma Classe Abstrata é que ela é extensível e contém Métodos Abstratos;</li><li>- <b>Método Abstrato:</b> São os métodos sem implementação nenhuma que são criados dentro de uma Classe Abstrata. O objetivo dos Métodos Abstratos é serem compartilhados entre as Sub Classes onde eles vão sofrer polimorfismo sobre cada Sub Classe dependendo da aplicação delas;</li><li>- <b>Classe Concreta:</b> São as Classes que herdam os atributos das Classes Abstratas e que possuem Métodos Sobrescritos, que são os métodos gerados por Polimorfismo á partir dos métodos abstratos, elas também podem possuir Métodos Finais dentro de si. O grande identificador de uma Classe Concreta é que elas não possuem Métodos Abstratos;</li><li>- <b>Método Sobrescrito:</b> São os métodos gerados por Polimorfismo á partir dos Métodos Abstratos, onde eles vão adequar o Método Abstrato á sua maneira;</li><li>- <b>Classe Final:</b> São as Classes Folha, que não podem ser herdadas por outras Classes;</li><li>- <b>Método Final:</b> São métodos que não podem ser sobrescritos, nem mesmo pelas suas Sub Classes caso tiverem;</li></ul>	
- reservada para a questão acima -			
- reservada para a questão acima -			