

MÉTODOS ESPECIAIS	MÉTODOS ESPECIAIS
O que são os <b>Métodos Especiais</b> ?	<p><b>Métodos Especiais:</b> são métodos da POO que executam alguma função específica que trará algum benefício ou facilidade aos programas desenvolvidos em POO. Esses métodos obedecem a uma sintaxe diferenciada dos métodos convencionais, e cada linguagem de programação vai tratar essa estrutura de forma singular, porém parecida. Dentre os métodos especiais podemos listar 3:</p> <ul style="list-style-type: none"><li>- <b>Métodos de Acesso</b> (Getters);</li><li>- <b>Métodos de Modificação</b> (Métodos Mutantes - Setters);</li><li>- <b>Métodos Construtores</b> (Constructors);</li></ul>
O que são os <b>Métodos de Acesso</b> ou <b>Getters</b> ?	<p><b>Métodos de Acesso:</b> conhecidos do inglês "Getters" são métodos de acesso cuja missão é permitir o acesso aos atributos privados de uma instância, com o objetivo de visualizá-los em seu estado atual sem modificar seus valores literalmente. Visto que, se os atributos forem privados, não poderemos nem mesmo visualizá-los numa chamada de instância comum, visto que a privatização é justamente para manter a segurança da informação. Os métodos Getters tem a missão de trazer segurança para os dados privados e ao mesmo tempo permitir a visualização deles aos usuários que acessarem por através do método Getter.</p>
Como seria a <b>sintaxe</b> de um método Getter convencional?	<p>Para criar um método Getter temos que declarar o tipo de dado do atributo seguido pela palavra reservada get e um nome que identifique o método Getter, e dentro dele colocamos a operação que vai atuar sobre o atributo privado.</p> <p><i><b>Lembrando que essa operação somente vai poder trabalhar os dados como eles estiverem no momento, ela não vai causar nenhuma mudança verdadeira sobre o dado.</b></i></p> <p>A sintaxe na prática é assim:</p>
- reservado para a questão acima -	<pre>classe Caneta  private char modelo private float ponta  public char getModelo() {     retorne modelo }  public float getPonta() {     retorne ponta }</pre>
- reservado para a questão acima -	<p>Para chamar o método para uma impressão por exemplo, podemos usar a seguinte sintaxe...</p> <pre>print(c1.getModelo()) print(c1.getPonta())</pre> <p><i><b>OBS: O ideal é criar um método getter para cada visualização disponível</b></i></p>
O que são os <b>Métodos de Modificação</b> ou <b>Setters</b> ?	<p><b>Métodos de Modificação:</b> conhecidos do inglês "Setters" são métodos por através dos quais podemos modificar os valores verdadeiros dos atributos mesmo que eles sejam privados. Os métodos "Setters" devem ser usados somente em casos especiais, onde sabemos que a modificação dos dados não causará danos as instâncias e ao sistema como um todo.</p>
Como seria a <b>sintaxe</b> de um método Setter convencional?	<p>Para criar um método Setter temos que declarar o tipo de dado do atributo seguido pela palavra reservada set e um nome que identifique o método Setter, e dentro dele colocamos a operação que vai atuar sobre o atributo privado.</p> <p><i><b>Lembrando que essa operação vai modificar os dados verdadeiros.</b></i></p> <p>A sintaxe na prática é assim:</p>
- reservado para a questão acima -	<pre>classe Caneta  private char modelo private float ponta  public char setModelo( valor ){     modelo = valor }  public float setPonta( valor ){     ponta = valor }</pre>
- reservado para a questão acima -	<p>Para chamar o método para uma impressão por exemplo, podemos usar a seguinte sintaxe...</p> <pre>c1.setModelo( valor ) c1.setPonta( valor )</pre> <p><i><b>OBS: O ideal é criar um método setter para cada visualização disponível</b></i></p>
O que é o <b>Método Construtor</b> ou <b>Construct</b> ?	<p><b>Método Construtor:</b> O método construtor é um método ativado toda vez que criamos uma nova instância. Esse método geralmente recebe o mesmo nome da Classe, e ele tem a missão de atribuir um comportamento padrão a cada instância criada. Por exemplo, digamos que a cada instância caneta criada, nós desejamos que ela esteja tampada e seja da cor azul, para isso basta atribuírmos chamadas para os métodos "tampar" e fazer uma atribuição padrão para o azul. <i><b>Se por um acaso a nossa Classe tiver setters, deveremos fazer a atribuição por através dos setters.</b></i></p>

MÉTODOS ESPECIAIS		MÉTODOS ESPECIAIS
Como seria a <b>sintáxe</b> de um método Setter convencional?		<p>Para criar um método Construct, basta usar achamada o método construtor, em algumas linguagens é identificado pela palavra reservada "construct", mas essa chamada vai depender de linguagem para linguagem. Em seguida criamos o método construtor segundo o nosso desejo. <i><b>Lembrando que, todas as instâncias criadas vão obedecer o construtor, e se tiver setters na classe o construtor só poderá fazer mudanças por através dos setters.</b></i></p> <p>A sintáxe na prática é assim:</p> <pre>class Caneta  private char modelo private float ponta private boolean tampa public char cor  public caneta construct(modelo, ponta, cor){     setModelo(modelo)     setPonta(ponta)     cor = "Azul"     tampar( ) }  public char setModelo( valor ){     modelo = valor }  public float setPonta( valor ){     ponta = valor }  private tampar( ){     tampa = True }</pre> <p>Para usar o método construtor basta criar uma instância...</p> <pre>c1 = Caneta('Bic', 0.5) # perceba que a cor, nem a tampa precisam ser declarados # o construtor já vai atribuir um padrão a eles</pre> <p><i><b>OBS: Lembre-se que toda instância criada vai obrigatoriamente obedecer ao padrão do método construtor.</b></i></p>
- reservado para a questão acima -		
- reservado para a questão acima -		
- reservado para a questão acima -		
- reservado para a questão acima -		