



Iniciando com Ruby e Orientação a Objetos



5.

Encapsulamento






O que é encapsulamento?





**Se refere a esconder os dados
(atributos) dentro do objeto;**

- Produz duas visões do objeto:
 - Visão Interna (COMO o objeto faz)
 - Visão Externa (O QUE o objeto faz)
- 


- Ex.: O funcionamento interno de um telefone é escondido (ou encapsulado) do usuário. Para o usuário realizar uma chamada telefônica (visão externa) ele não precisa conhecer os detalhes da eletrônica utilizados no telefone (visão interna)



- 
- Em se tratando de classes, o em encapsulamento é obtido através da visibilidade, conhecida também como "modificadores de acesso".
- 




Os modificadores de acessos mais comuns são:

- **public / público:** Representado por um sinal de mais "+" na notação UML. Elementos declarados como público podem ser acessados a partir de qualquer outra classe;
- 




Os modificadores de acessos mais comuns são:

- **private / privado:** Representado por um sinal de menos "-" na notação UML. Elementos private só podem ser acessados pela própria classe;
- 



Os modificadores de acessos mais comuns são:

- **protected / protegido:** Representado por um sinal de sustenido "#" na notação UML. Elementos **protected** só podem ser acessados pela própria classe e por suas descendentes;
- 



Com o que vimos até agora, já podemos identificar facilmente o que o diagrama abaixo representa.

Conta
- saldo: float = 0.0 - numero: String
+ debitar(float: valor) + creditar(float: valor) + transferir(float: valor) + consultarSaldo()

