

## Strings e associação entre classes

---

Prof. Fernando J. C. Branquinho  
<http://br.geocities.com/branqs>

---

### Strings

=====

Classe para armazenamento de séries de caracteres

Exemplo:

```
String nome="Branquinho";
```

Concatenação

```
String primeiroNome = "Fernando"
String segundoNome = "Branquinho"
String nomeCompleto = primeiroNome + " " + segundoNome
```

Métodos úteis

Ex: String nome = "Branqs"

Assinatura	Exemplo uso	Retorno
-----	-----	-----
char charAt(int)	nome.charAt(2) nome.charAt(0)	'a' 'B'
boolean equals(String)	nome.equals("Bronqs") nome.equals("Branqs")	false true
int length()	nome.length()	6
String substring(int, int)	nome.substring(1, 3)	"ran"

### Associação entre classes

=====

O uso acima da classe "String" é um exemplo de associação entre classes.

Considerando que ao criarmos uma nova classe estamos definindo um "novo tipo", podemos utilizá-la para definir o tipo de um atributo.

Exemplo:

Suponha a criação da seguinte classe Cliente:

```
class Cliente{
    String nome;
    String senha;
}
```

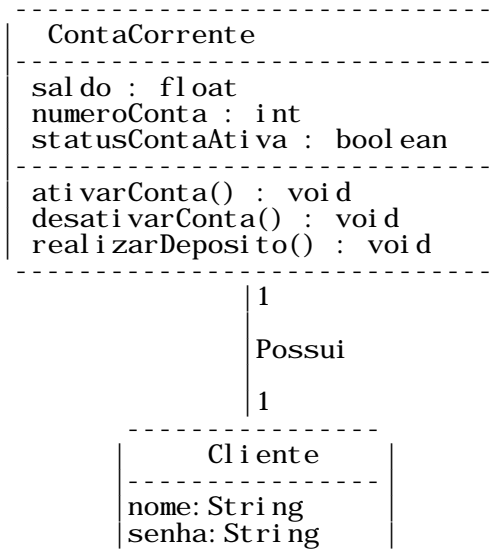
Podemos criar um atributo em ContaCorrente que informa que uma conta corrente possui um cliente associado, caracterizado como "correntista".

Exemplo:

```
class ContaCorrente{
    float saldo;
    int numeroConta;
    boolean statusContaAtiva;
    Cliente correntista;
    .
    .
    .
}
```

No exemplo acima, o atributo "correntista" pode armazenar um objeto do tipo "Cliente". Um objeto do tipo "Cliente" possui "nome" e "senha".

Podemos dizer que essa construção equivale à seguinte representação de classes em UML:



A partir da classe ContaCorrente podemos ter acesso aos membros da classe Cliente através do uso do operador "." (ponto). Este operador permite acessarmos os atributos e os métodos de um objeto. Sendo assim, dentro de um método qualquer de ContaCorrente podemos executar a seguinte operação:

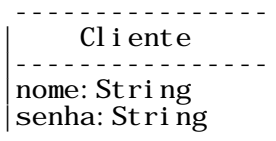
```
String nomeCorrentista = correntista.nome;
```

No exemplo acima, estamos criando uma variável do tipo String que receberá o nome do correntista associado à conta corrente.

Na explicação sobre Strings podemos ver vários exemplos de utilização do operador "." agindo sobre os métodos do objeto "nome".

#### Exercícios

- 1) Acrescente a seguinte classe "Cliente" ao projeto Banco criado nas últimas aulas. No momento da criação de um objeto Cliente, deverá ser informado seu nome e senha (Utilize o método construtor para isso).



- 2) Crie um atributo do tipo Cliente na classe ContaCorrente. Modifique os métodos construtores da ContaCorrente, de modo que um cliente seja passado no momento da criação de uma conta corrente.
- 3) Permita obter o nome do correntista a partir da classe ContaCorrente
- 4) Crie 5 contas e somente 2 clientes. Associe a 1a e 3a conta ao 1o correntista, e a 2a, 4a e 5a conta, ao 2o correntista.

Final do documento

Fernando J. C. Branquinho  
<http://br.geocities.com/branqs>