

Classes, atributos e construtores em Java

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

Java respeita a estrutura definida pelo paradigma "Orientado a objetos"

Paradigma

=====

(gr. paradigma) Modelo, padrão, norma;

Objeto

=====

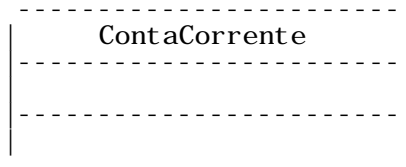
Modelo de um conceito do mundo real que pode possuir características e comportamento.

Classe

=====

Projeto de um objeto

Exemplo através de representação UML



Implementação em Java

```
class ContaCorrente {  
  
}
```

Membros de uma classe

=====

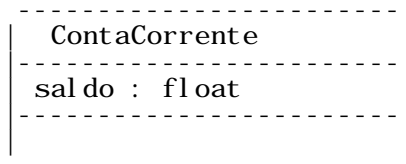
Atributos
Métodos (Veremos mais adiante)

Atributos

Representam características dos objetos, ou seja, informações que definem os objetos.

Exemplo: Uma ContaCorrente possui um saldo. Cada ContaCorrente existente possui o seu saldo correspondente.

Representação UML



Todo atributo necessita de um tipo.
No exemplo acima, o atributo saldo é do tipo float.

Os atributos que representam características, normalmente são especificados através de "tipos primitivos".

Tipos primitivos

Tipos Inteiros

byte	->	-128 a 127 (1 byte)
short	->	-32.768 a 32.767 (2 bytes)
int	->	-2.147.483.648 a 2.147.483.647 (4 bytes)
long	->	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807

Tipos de ponto flutuante

float	->	-/+ 3.40282347E38 (4 bytes)
double	->	-/+ 1.79769313486231570E308 (8 bytes)

Tipo caracter

char	->	'a', 'b' ... etc
------	----	------------------

Tipo estado lógico

boolean	->	true ou false
---------	----	---------------

Implementação em Java

```
class ContaCorrente {  
    float saldo;  
}
```

Construtores

=====

Método com o mesmo nome da classe.
É executado automaticamente quando criamos um objeto.
Permite inicializar o objeto criado.
Podem existir vários construtores com assinaturas diferentes.

Ex:

```
class ContaCorrente {  
    float saldo;  
  
    ContaCorrente() {  
        saldo=0;  
    }  
  
    ContaCorrente(float saldoInicial) {  
        saldo=saldoInicial;  
    }  
}
```

A expressão entre parênteses significa um argumento (ou parâmetro). Todo argumento deve possuir um tipo e um nome. Este nome será utilizado para referenciar o parâmetro passado dentro do método construtor.

O tempo de vida de um argumento corresponde ao tempo de execução do construtor, ou seja, ao finalizar a execução do construtor, o argumento e seu valor são descartados.

No exemplo acima, o valor do argumento é armazenado no atributo saldo. Esta é uma técnica amplamente utilizada para preservar o valor dos argumentos passados.

Exercícios

=====

- 1) Utilizando o programa BlueJ, crie um novo projeto chamado "Banco"
- 2) Dentro desse projeto, crie uma classe chamada "ContaCorrente" que possua os seguintes atributos:

Nome atributo	Tipo
-----	-----
saldo	float
numeroConta	int
statusContaAtiva	boolean

Obs: Elimine todas as linhas desnecessárias, que foram automaticamente inseridas pelo Bluej durante a criação da classe.

- 3) Crie dois objetos a partir da classe ContaCorrente, nomeando-os como "contaJose" e "contaJoao" (Lembre-se, nomes de objetos devem ser escritos em letra minúscula). Perceba que os dois objetos possuem os mesmos valores em seus atributos.
- 4) Monte um construtor que permita configurar o número da conta durante a sua criação.
- 5) Modifique o construtor de modo que o status da conta possa ser passado como parâmetro.
- 6) Crie 3 objetos do tipo ContaCorrente com a seguinte configuração:

Nome do objeto (instância)	Número Conta	Conta ativa
-----	-----	-----
contaBranqs	1234	true
contaLula	5678	false
contaPapa	2222	true

- 7) Permita a especificação "opcional" de um saldo inicial para a conta. Caso o saldo não seja passado, atribua o valor R\$50,00 como default.
- 8) Chame o Branqs para verificar se o resultado foi atingido

Final do documento

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