



PОО com Java - 02

Classe → Objeto

- A Classe é um modelo, ou tipo de objeto, já o Objeto é a instância desse modelo.
 - Exemplo clássico: Molde e Ovo de Páscoa.





Atributos

- ▢ Definem e denotam o estado do objeto em dado momento.
 - Atributos mudam, as vezes não.
 - Peso, cor da pele, gênero, DNA.
 - Atributos também indicam, ou definem, o estado de uma ação desempenhada pelo Objeto (E.g.: velocidade).



Métodos

- Indicam, denotam ou determinam os comportamentos do Objeto.
 - Pessoa: dormindo, andando, moscando...
 - Determinam o estado de alguma ação que o objeto execute.
 - Denotam, ou retornam, o estado de uma ação (verbo) ou de um atributo (velocidade).



Métodos vs Funções

- ❑ Em linhas gerais são a mesma coisa.
- ❑ Métodos executam algo sem, necessariamente, retorno. (Jogar lixo)
- ❑ Funções executam algo com necessariamente um retorno. (Pagar conta da Celpe)



Tipos de dados

- Primitivos: são os tipos de dados mais simples que dão origem a tipos de dados herdados.
 - Considerando Java.
 - E.g.: boolean, byte, short, int, long, float, double, char.



Variáveis

- São espaços em memória RAM utilizados para armazenar dados.
 - Dados possuem tipo.
 - Tipos possuem tamanho.
 - O tamanho depende do tipo e o tipo depende da arquitetura da CPU e também da linguagem utilizada.



Variáveis

- ▣ Operações com variáveis seguem em ordem, não necessariamente, a sequência:
 - Declaração, define-se nome, tipo e tamanho. Exemplo.
 - Inicialização, define-se o valor.
 - Redefinição, atribui-se um novo valor à variável.
 - Acesso à variável, recupera seu valor.



Variáveis

- A variável é na verdade um endereço de memória que aponta para os dados.
 - Em “C” chamaríamos o endereço de ponteiro.
 - Em Java cada variável é um objeto.
 - Em Java não se opera com ponteiros.



Variáveis

□ Cópia de variável por referência em Java é feita de modo implícito.

- `int varA = 10;`
- `int varB;`
- `varB = varA;`
- `System.out.println(varB);`
- Terei 2 Objetos, `varA` e `varB`.

Variáveis Java vs “C”

- Em “C” poderia ser feita a cópia pelo ponteiro.
 - `int varA = 10;`
 - `int *ponteiro = &varA; // Ponteiro`
 - `int varB;`
 - `varB = *ponteiro;`
- Referência à variável original `varA`, seu endereço, e não a uma cópia completa. De certo modo, economiza memória.
- Tem-se apenas uma variável e uma referência.




Casting ou Type Casting

- Mudança do tipo de dado, primitivo, da variável para outro tipo de dado também primitivo.
 - Conversão implícita, controlada pelo core do Java.
 - Conversão explícita, necessária alguma atenção do programador.



Casting ou Type Casting

- Widening. Conversão implícita, controlada pelo core do Java.
 - De um tipo “menor” para um “maior”. O dado cabe por completo.
 - byte → short → char → int → long → float → double



Casting ou Type Casting Widening (amplicação)

- ❑ `int menor = 1853674;`
- ❑ `double maior = (double) menor;`
- ❑ `System.out.println(menor);`
- ❑ `System.out.println(maior);`



Casting ou Type Casting

- Narrowing. Conversão explícita, precisa de algum entendimento do programador.
 - De um tipo “maior” para um “menor”. O dado não cabe. Haverá truncagem.
 - double → float → long → int → char → short → byte



Casting ou Type Casting Narrowing (estreitamento)

- ❑ `int maior = 1253684;`
- ❑ `short menor = (short) maior;`
- ❑ `System.out.println(maior);`
- ❑ `System.out.println(menor);`