



Características Avançadas das Classes – **Operador Static**

LPOO

Prof. Fabrício Martins Lopes

fabricao@utfpr.edu.br

[Uso do especificador **static**]

- O especificador **static** pode ser aplicado a **atributos, métodos e inicializadores**.
- As classes podem ter tanto atributos quanto métodos estáticos.
- **obs:** apenas as classes internas podem ser estáticas.

[Atributos estáticos]

- Quanto aplicado o operador static a um atributo, o mesmo passa a **pertencer a classe**, passando a ser conhecido como “atributo da classe”.
- Em outras palavras, todos os objetos instanciados da classe passam a **compartilhar o mesmo atributo**, similarmente ao que ocorre com **variáveis globais** em linguagens de programação estruturadas.

[Exemplos de atributos estáticos]

- Vários atributos constantes são definidos em Java como **public static final**.
 - E (2.71828...) e PI (3.14159...) da classe Math.
- Outro exemplo de variável public static final é a variável **out** da classe **System**.

[System.out]

- O atributo **out** está associado à apresentação de caracteres na saída padrão, ou seja, na tela do monitor.
 - `public static final PrintStream out`
- Uma simples análise dessa declaração mostra que:
- `out` é um atributo de classe e, portanto, pode ser acessado na forma **System.out**;
 - Esse atributo é público
 - O atributo é final
 - O Atributo é do tipo `PrintStream`.

[Métodos estáticos]

- Os métodos estáticos pertencem a classe e não operam sobre instâncias dessa classe.
- Como os métodos estáticos não funcionam com uma instância da classe, eles só podem acessar membros estáticos (atributos e métodos) da classe.

[Inicializador estático]

- Um atributo estático pode ser inicializado com uma atribuição no momento de sua declaração ou usando um bloco de inicialização estático.
 - Atribuição inicial
 - `public static double fatorreducao = 1.3;`

[Inicializador estático]

- Quando a inicialização não é possível em uma expressão. Usa-se um bloco de inicialização estático.

```
public static double [] aleatorios = new double [100];
static{
    int i;
    for (i = 0; i < 100; i++){
        aleatorios[i] = java.lang.Math.random();
    }
}
```


[Inicializador estático]

- Inicialização estática ocorre quando a classe é inicialmente carregada.
 - Primeiro: os atributos estáticos são inicializados na ordem em que estão declarados.
 - Segundo: os blocos de inicialização estáticos são executados na ordem em que aparecem na declaração da classe.

```
public class Hello{  
    static{  
        System.out.println("Olá Java");  
    }  
}
```

[Considerações...]

- Normalmente os **atributos estáticos** não são inicializados no método construtor.
- Os **atributos estáticos** normalmente são utilizados na padronização de valores (constantes) dentro do projeto/sistema.
- **Métodos estáticos** não podem ser abstratos.
- Os **métodos estáticos** normalmente são usados em rotinas isoladas dentro do projeto/sistema.

[Referências Consultadas]

- **DEITEL, P.J. Java - Como Programar. Porto Alegre: Bookman, 2003.**
- **HORSTMANN, Cay. Big Java. Porto Alegre: Bookman, 2004.**
- **HORSTMANN, Cay, S. e CORNELL, Gary. Core Java 2. São Paulo: Makron Books, 2001 v.1. e v.2.**
- **MORGAN, Michael. Java 2 para Programadores Profissionais. Rio de Janeiro: Ciência Moderna, 2000.**