



SÃO
PAULO
TECH
SCHOOL

Linguagem de Programação

Vida Útil do Objeto e Construtor

Profa. Célia Taniwaki

Prof. Diego Brito

Profa. Giuliana Miniguiti

Vida útil do objeto

Vida útil do objeto

- Objeto é criado, utilizado e depois destruído durante a execução do sistema
- Programador sabe o momento em que o objeto é criado (quando se usa o **new**), mas não sabe exatamente quando é destruído
- **Garbage Collector** ("coletor de lixo")
 - Processo do Java RunTime que é executado periodicamente
 - Verifica se existem objetos na memória que não estão sendo usados, se encontrar objetos não referenciados, libera a área da memória que eles estão ocupando

Construtor vs. Método

Tem sua forma parecida com a de um método, porém **não é um método**, pois o método pode ser chamado várias vezes

- Construtor é chamado e executado **apenas no momento da criação do objeto**
- Nome do construtor: **mesmo nome da classe**
- **Não retorna nada** mas não se deve especificar void
- Por padrão seu **modificador sempre será public**

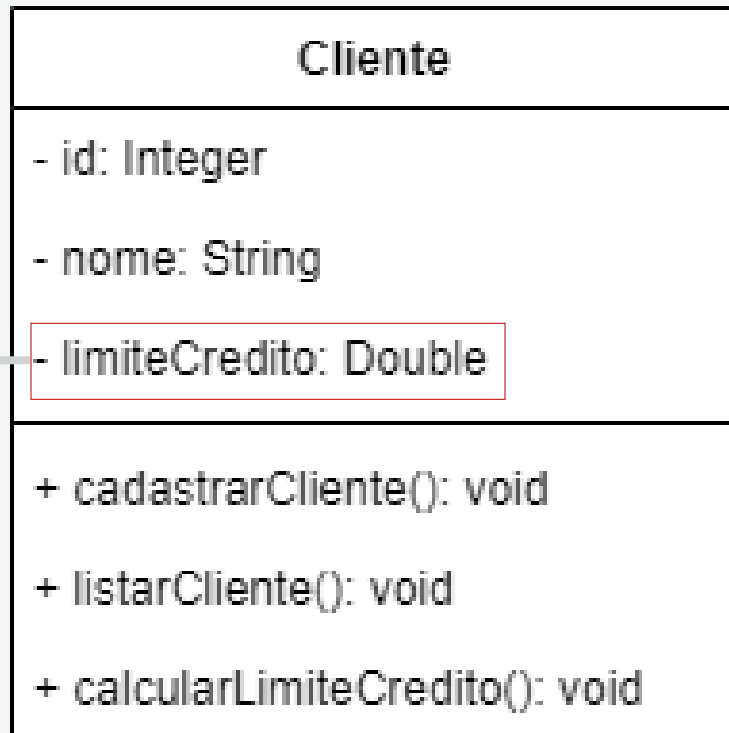
Porque usar Construtor

Executar algum código no momento da criação do objeto, como por exemplo, **inicialização dos valores do atributo**

Sua execução é sempre obrigatória... e porque o código funciona sem criar um construtor ?

- Quando a definição da classe não contém um construtor, o compilador Java cria automaticamente um construtor default, que não recebe parâmetros e não faz nada

Exemplo de construtor



```
Cliente(Integer id, String nome, Double limiteCredito){
    this.id = id;
    this.nome = nome;
    this.limiteCredito = limiteCredito;
}
```

Construtores sobrecarregados

É possível haver mais de um construtor em uma mesma classe, com lista de parâmetros diferentes

Dá-se o nome de **Construtores Sobrecarregados**

```
Cliente(Integer id, String nome, Double limiteCredito){  
    this.id = id;  
    this.nome = nome;  
    this.limiteCredito = limiteCredito;  
}
```

```
Cliente(){  
    this.id = 0;  
    this.nome = "";  
    this.limiteCredito = 0.0;  
}
```

Palavra-chave this

A palavra-chave **this** é utilizada para:

- Diferenciar a variável atributo de outra variável com mesmo nome:

```
Cliente(Integer id, String nome, Double limiteCredito){  
    this.id = id;  
    this.nome = nome;  
    this.limiteCredito = limiteCredito;  
}
```

- Se os parâmetros tivessem nome diferente do atributo, o this não seria necessário:

```
Cliente(Integer id1, String nome1, Double limiteCredito1){  
    id = id1;  
    nome = nome1;  
    limiteCredito = limite1;  
}
```


Palavra-chave this

A palavra-chave **this** também é utilizada para:

- Construtor chamar outro construtor:

```
Cliente(){  
    this( 0, "", 0.0 );  
}
```

- Nesse caso, existem algumas regras:
 - Somente construtores podem chamar outros construtores
 - Se um construtor for chamado a partir de outro, a chamada deve ser a primeira linha de código dentro do construtor
 - Construtores de uma mesma classe são sempre chamados por this



Agradeço

a sua atenção!

Obrigado!

Em caso de dúvidas, entre em contato com:

diego.lima@bandtec.com.br

giuliana.franca@sptech.school

**SÃO
PAULO
TECH
SCHOOL**