

FIAP GRADUAÇÃO

Agenda

- Sobrecarga de métodos
- Construtores

Sobrecarga de métodos

Sobrecarga de método, ou overload, é o conceito de utilizar o mesmo nome para mais de um método.

Para que isso ocorra estes métodos precisam receber parâmetros diferentes (tipo ou quantidade).

Sobrecarga de métodos

Sobrecarga de método, ou overload, é o conceito de utilizar o mesmo nome para mais de um método.

Para que isso ocorra estes métodos precisam receber parâmetros diferentes (tipo ou quantidade).

Sobrecarga de métodos

```
public class Carro {  
    public void mover(String direcao){  
        ...  
    }  
    public void mover(int metros){  
        ...  
    }  
    public void mover(String direcao, int metros){  
        ...  
    }  
}
```

Sobrecarga de métodos

É importante ressaltar que no Java o nome do parâmetro não faz diferença, apenas o tipo:

```
public void mover(String direcao){  
    ...  
}
```

```
public void mover(String metros){  
    ...  
}
```

ERRO!!

(método já definido na classe)

Construtores

- O construtor é a forma que temos de criar um objeto (uma instância) de uma classe
- Para utilizar o construtor precisamos da palavra reservada “new”
- Toda classe na Java tem pelo menos um construtor.

Construtor

Para criar um construtor temos que utilizar o mesmo nome da classe:

```
[modificador de visibilidade] [NomeDaClasse] (  
    [parâmetros]  
) {  
    [corpo do construtor]  
}
```


Construtores

```
private int idade; ➡ Algum atributo
```

```
public Pessoa(int idade) {  
    this.idade = idade; ➡ Construtor que recebe  
    // alguma idade como  
    // parâmetro  
}
```

Construtores

Para criar um objeto (instância) de Pessoa, agora podemos utilizar o construtor que acabamos de declarar:

```
Pessoa pessoa = new Pessoa(20);
```

Construtores

É importante perceber que, caso existam, os construtores declarados são a ÚNICA forma de criar um objeto. No caso da classe Pessoa, se declararmos apenas o construtor que recebe idade como parâmetro, não é mais possível instanciar objetos sem informar a idade:

```
Pessoa pessoa = new Pessoa();
```

Chamada de construtor inválida.
(cannot be applied)

Construtor Default

Cada classe pode ou não ter construtores declarados. Se declararmos um construtor, o construtor default não irá mais funcionar.

Ou seja, se não declararmos nenhum construtor explicitamente, podemos chamar o construtor default:

```
Pessoa pessoa = new Pessoa();
```

Construtores

E se quisermos utilizar algum construtor específico (como a classe Pessoa recebendo a idade) e o construtor default?

Construtores

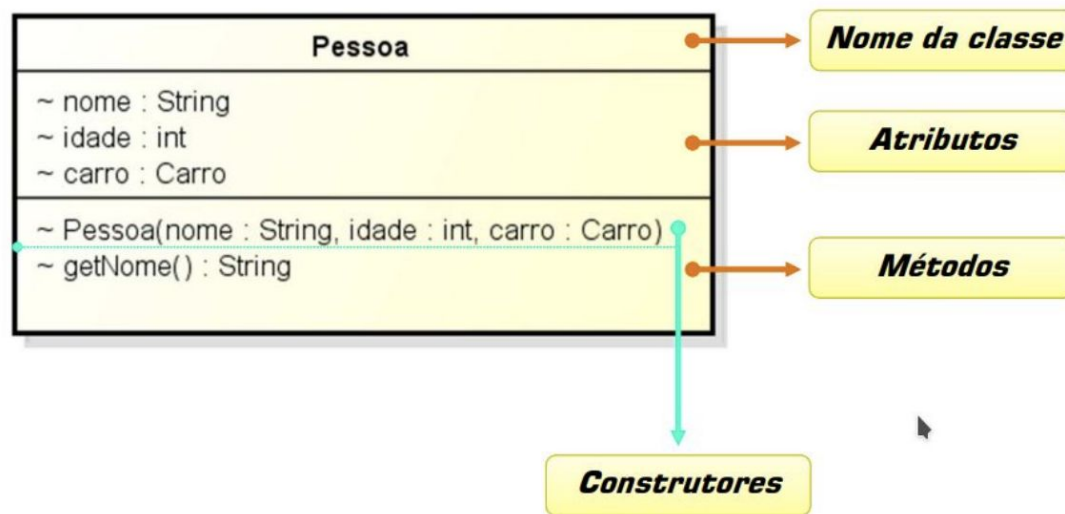
Múltiplos construtores:

```
public Pessoa(int idade) {  
    this.idade = idade;  
}
```

```
public Pessoa() {}
```

Construtores

Representação de construtores em UML



Palavra reservada - this

- Tanto em métodos quanto em construtores, se quisermos acessar a instância (referência) da própria classe, podemos utilizar a palavra reservada “this”

```
public class Carro {  
  
    private String modelo;  
    private int quilometragem;  
  
    public Carro(String modelo){  
        this.modelo = modelo;  
    }  
  
    public void mover(int quilometragem){  
        this.kilometragem = this.kilometragem + quilometragem ;  
    }  
}
```


OBRIGADO

FIAP

Copyright © 2020 | Professor Fabio Tadashi Miyasato
Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente
proibido sem consentimento formal, por escrito, do professor/autor.