

# Sobrecarga

Prof<sup>a</sup>. Rachel Reis rachel@inf.ufpr.br



## Sobrecarga

- Tipos:
  - Sobrecarga de construtores
  - Sobrecarga de métodos



#### Sobrecarga de Construtores

- Uma classe pode ter vários construtores sobrecarregados permitindo que objetos dessa classe sejam inicializados de diferentes maneiras.
- Para sobrecarregar construtores, basta fornecer múltiplas declarações de construtor com assinaturas diferentes.
- As regras para sobrecarga de construtores são:
  - A lista de parâmetros tem que mudar
  - O modificador de acesso pode mudar

```
public class ContaBancaria{
                                        Três construtores
  private String nomeT;
  private double saldo;
  // Construtor 1
  public ContaBancaria() { }
  // Construtor 2
  public ContaBancaria(double saldo) {
     this.saldo = saldo;
  // Construtor 3
  public ContaBancaria(String nomeT, double saldo){
      this.nomeT = nomeT;
      this.saldo = saldo;
```

```
public class ContaBancaria{
                                      Lista de parâmetros
  private String nomeT;
                                        tem que mudar
  private double saldo;
  // Construtor 1
  public ContaBancaria() { }
  // Construtor 2
  public ContaBancaria(double saldo) {
     this.saldo = saldo;
  // Construtor 3
  public ContaBancaria(String nomeT, double saldo) {
      this.nomeT = nomeT;
      this.saldo = saldo;
```

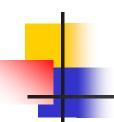
```
Modificador de acesso
public class ContaBancaria{
                                        pode mudar
  private String nomeT;
  private double saldo;
  // Construtor 1
  public ContaBancaria() { }
  // Construtor 2
  public ContaBancaria(double saldo) {
     this.saldo = saldo;
  // Construtor 3
  public ContaBancaria(String nomeT, double saldo){
      this.nomeT = nomeT;
      this.saldo = saldo;
```



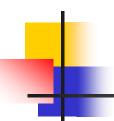
#### Construtores - referência this

- Um construtor pode ser chamado dentro de outro construtor utilizando a palavra-chave this.
- A instrução this deve vir <u>sempre</u> na primeira linha do construtor.

```
Referência this
public class ContaBancaria{
  private String nomeT;
  private double saldo;
  // Construtor 1
  public ContaBancaria() {
      this("", 0.0);
  // Construtor 2
  public ContaBancaria(double saldo) {
      this("", saldo);
  // Construtor 3
  public ContaBancaria(String nomeT, double saldo) {
      this.setNomeT(nomeT);
      this.setSaldo(saldo);
```



- É a capacidade de possuir <u>métodos diferentes</u> com mesmo nome, mas com parâmetros diferentes.
- O interpretador determinará qual método deve ser invocado pelo tipo de parâmetro passado.



- As regras para sobrecarga de métodos são:
- 1) O nome do método **tem** que ser o mesmo.
- 2) A lista de parâmetros **tem** que mudar.
- 3) O tipo de retorno **pode** mudar.
- 4) O modificador de acesso **pode** mudar.

Exemplo 1

```
public void imprimir(int i) { ... }

public void imprimir(float f) { ... }

public void imprimir(String s) { ... }
```

nome do método: tem que ser o mesmo.

Exemplo 1

```
public void imprimir(int i) { ... }

public void imprimir(float f) { ... }

public void imprimir(String s) { ... }
```

lista de parâmetros: tem que mudar

Exemplo 1

```
public void imprimir(int i) { ... }

public void imprimir(float f) { ... }

public void imprimir(String s) { ... }
```

tipo de retorno: <u>pode</u> mudar

Exemplo 1

```
public void imprimir(int i) { ... }

public void imprimir(float f) { ... }

public void imprimir(String s) { ... }
```

modificador de acesso: <u>pode</u> mudar

```
public int soma(int x, int y){
                                            nome do método
   return x + y;
                                          tem que ser o mesmo
public double soma(double x, double y){
   return x + y;
public String soma(String x, String y){
      return x + y;
```

```
public int soma(int x, int y){
                                             lista de parâmetros
   return x + y;
                                               tem que mudar
public double soma(double x, double y){
   return x + y;
public String soma(String x, String y){
      return x + y;
```

```
public int soma(int x, int y){
                                                   tipo de retorno
   return x + y;
                                                    pode mudar
public double soma(double x, double y){
   return x + y;
public String soma(String x, String y){
       return x + y;
```

```
public int soma(int x, int y){
                                         modificador de acesso
                                               pode mudar
   return x + y;
public double soma(double x, double y){
   return x + y;
public String soma(String x, String y){
      return x + y;
```



#### Para praticar...

 Escreva 3 exemplos válidos de sobrecarga para o método abaixo:

```
public void teste(int x, String y, float z){ }
```



#### Referências

- Deitel, P. J.; Deitel, H. M. (2017). Java como programar.
   10a edição. São Paulo: Pearson Prentice Hall.
- Barnes, D. J. (2009). Programação orientada a objetos com Java: uma introdução prática usando o BlueJ (4. ed.). São Paulo, SP: Prentice Hall.
- Boratti, I. C. (2007). Programação orientada a objetos em Java. Florianópolis, SC: Visual Books.