

Orientação a Objetos 1

# Sobrecarga

Prof. MSc. Vinícius Camargo Andrade

[vcandrade@utfpr.edu.br](mailto:vcandrade@utfpr.edu.br)

Departamento Acadêmico de Informática  
Universidade Tecnológica Federal do Paraná

# Sobrecarga

Overloading

# Sobrecarga

*Quando um **identificador** é utilizado em diversas outras operações (métodos) em uma mesma classe ocorre uma sobrecarga.*

# Sobrecarga

*Consistem em criarmos o mesmo método com **parâmetros diferentes** em termos de:*

- *Tipos;*
- *Quantidade de parâmetros;*
- *Posições dos tipos.*

# Exemplo 1

Sobrecarga

# Exemplo 1

```
public class Data {  
  
    private String data;  
  
    public void atribuirData() {  
  
        Date hoje = new Date();  
        SimpleDateFormat df = new SimpleDateFormat("dd/MM/yyyy");  
        this.data = df.format(hoje);  
    }  
  
    public void atribuirData(String data) {  
  
        this.data = data;  
    }  
  
    public void atribuirData(int dia, int mes, int ano) {  
  
        this.data = dia + "/" + mes + "/" + ano;  
    }  
}
```

## Data

- data : String

+ atribuirData( ) : void

+ atribuirData(String) : void

+ atribuirData(Integer, Integer, Integer) : void

# Exemplo 2

Sobrecarga

# Exemplo 2

```
public class Mensageiro {  
    public void imprimirMensagem(String texto) {  
        System.out.println("Mensagem: " + texto);  
    }  
  
    public void imprimirMensagem(int numero) {  
        System.out.println("Número: " + numero);  
    }  
  
    public void imprimirMensagem(String texto, int numero) {  
        System.out.println(numero + ", escreveu: " + texto);  
    }  
}
```

```
public class MensageiroTeste {  
    public static void main(String[] args) {  
        Mensageiro mensageiro = new Mensageiro();  
        mensageiro.imprimirMensagem("Olá, Tudo bem?");  
        mensageiro.imprimirMensagem(99887766);  
        mensageiro.imprimirMensagem("Orientação a Objetos", 11223344);  
    }  
}
```

Mensageiro

+ imprimirMensagem(String) : void  
+ imprimirMensagem(Integer) : void  
+ imprimirMensagem(String, Integer) : void



# Exemplo 3

Sobrecarga

# Sobrecarga

*Geralmente utiliza-se o polimorfismo **sobrecarga** em **métodos construtores**, pois necessita-se iniciar os objetos de uma mesma classe de diferentes maneiras.*

# Exemplo 3

Conta
- saldo : Double
+ Conta( ) + Conta(saldo : Double)

```
public class Conta {  
  
    private Double saldo;  
  
    public Conta() {  
  
    }  
  
    public Conta(Double saldo) {  
  
        this.saldo = saldo;  
  
    }  
}
```

```
public class ContaTeste {  
  
    public static void main(String[] args) {  
  
        Conta conta1 = new Conta();  
        Conta conta2 = new Conta(1000.00);  
  
    }  
}
```

# Exemplo 4

Sobrecarga

# Exemplo 4

Conversor
+ converterValor(Long) : Integer + converterValor(Float) : Integer + converterValor(Double) : Integer + converterValor(String) : Integer

```
public class Conversor {  
    public int converterValor(long valor) {  
        return (int) valor;  
    }  
    public int converterValor(float valor) {  
        return (int) valor;  
    }  
    public int converterValor(double valor) {  
        return (int) valor;  
    }  
    public int converterValor(String valor) {  
        return Integer.parseInt(valor);  
    }  
}
```

# Exercício

# Exercício

*Implemente os métodos somar de uma calculadora.*

- *o primeiro recebe dois valores inteiros e retorna o resultado inteiro;*
- *o segundo recebe dois valores double e retorna o resultado double*
- *o terceiro recebe um arraylist de N valores inteiros, soma todos eles e retorna o resultado, que será um valor inteiro.*

Calculadora
+ somar(Integer, Integer) : Integer + somar(Double, Double) : Double + somar(List<Integer>) : Integer