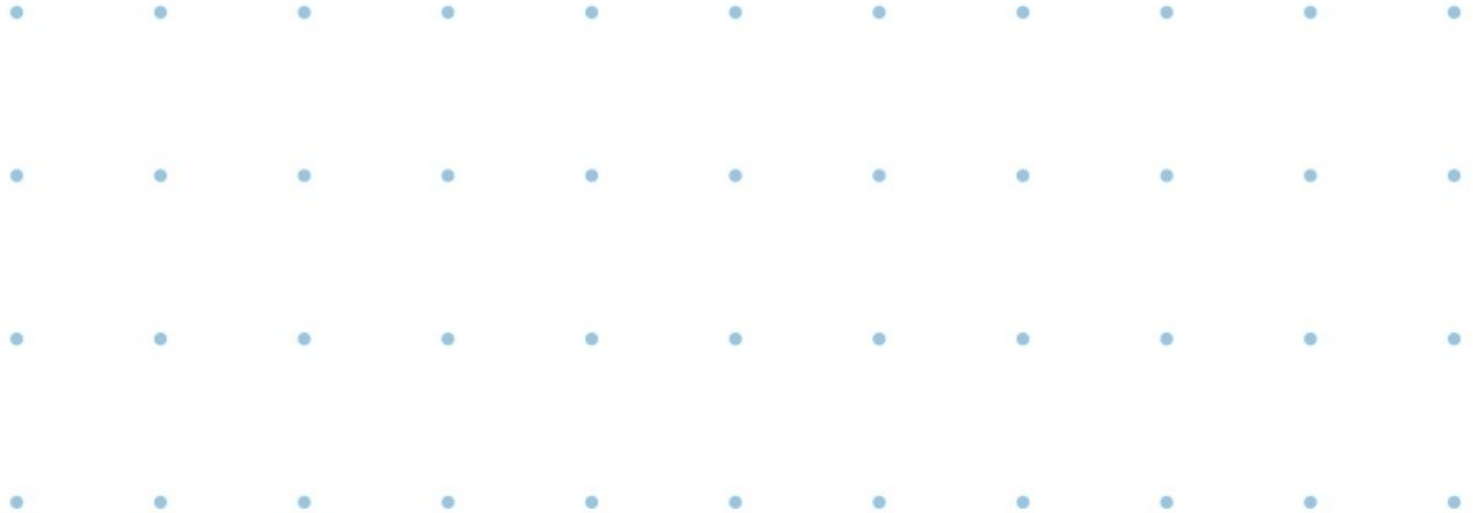




**INSTITUTO
FEDERAL**
Norte de Minas Gerais



P OO - Sobrecarga

SOBRECARGA

- Revisão:
 - Uma classe possui atributos e métodos. Também possui um construtor padrão que pode ser substituído por um construtor alterado.
 - Os objetos se comunicam por troca de mensagens que são as chamadas e os retornos dos métodos.

SOBRECARGA

- Uma classe pode possuir mais de um construtor, bem como possui um método que dê um retorno diferente dependendo de como é chamado.
- Essa característica é chamada de “Sobrecarga”.
- O que diferencia os construtores e os métodos na sobrecarga é a sua assinatura.

SOBRECARGA

- Exemplo:
 - Vamos utilizar a classe ContaBancária:
 - Ela possui um construtor que inicializa o atributo saldo com o valor R\$ 0,00.
 - Mas podemos criar uma classe que possibilite inicializar o saldo um determinado valor

```
1 public class ContaBancaria {  
2  
3     private int numero;  
4     private Double saldo;  
5  
6     public int getNumero(){  
7         return numero;  
8     }  
9  
10    public void setNumero(int numero){  
11        this.numero = numero;  
12    }  
13  
14    public ContaBancaria(){  
15        this.saldo = 0.0;  
16    }  
17  
18    public void depositar(Double valor){  
19        saldo += valor;  
20    }  
21  
22    public void sacar (Double valor){  
23        saldo -= valor;  
24    }  
25 }  
26  
27
```

SOBRECARGA

- Exemplo:

```
1 public class ContaBancaria {  
2  
3     private int numero;  
4     private Double saldo;  
5  
6     public int getNumero(){  
7         return numero;  
8     }  
9  
10    public void setNumero(int numero){  
11        this.numero = numero;  
12    }  
13  
14    public ContaBancaria(){  
15        this.saldo = 0.0;  
16    }  
17  
18    public ContaBancaria(Double saldo){  
19        this.saldo = saldo;  
20    }  
21  
22    public void depositar(Double valor){  
23        saldo += valor;  
24    }  
25  
26    public void sacar (Double valor){  
27        saldo -= valor;  
28    }  
29  
30 }
```

SOBRECARGA

- Exemplo:
 - A classe agora possui dois construtores que se diferenciam pelos parâmetros.

```
13  
14  public ContaBancaria(){  
15      this.saldo = 0.0;  
16  }  
17  
18  public ContaBancaria(Double saldo){  
19      this.saldo = saldo;  
20  }  
21
```

SOBRECARGA

- Exemplo:
 - O que determina qual construtor será chamado é a passagem de parâmetros:

```
1 public class Banco {  
2  
3     public static void main (String[] args) {  
4  
5         ContaBancaria conta1 = new ContaBancaria();  
6  
7         ContaBancaria conta2 = new ContaBancaria(1000.0);  
8  
9  
10  
11     }  
12 }  
13
```

SOBRECARGA

- Exemplo:
 - O que determina qual construtor será chamado é a passagem de parâmetros:

```
1 public class Banco {  
2  
3     public static void main (String[] args) {  
4  
5         ContaBancaria conta1 = new ContaBancaria();  
6  
7         ContaBancaria conta2 = new ContaBancaria(1000.0);  
8  
9  
10  
11     }  
12 }  
13
```

Saldo conta1 = R\$ 0,00
Saldo conta2 = R\$ 1000,00

SOBRECARGA

- Sobrecarga de método
 - Assim como no caso do construtor os métodos também podem ter mais de uma assinatura.
 - A sobrecarga pode ser definida pelo tipo, pela quantidade ou pela ordem dos parâmetros.

SOBRECARGA

- Exemplo
 - Vamos criar uma classe CalculaPotencia que possui um método faz o cálculo do quadrado de um número:

```
1  
2 public class CalculaPotencia {  
3  
4     public float calculo(int numero) {  
5         return numero*numero;  
6     }  
7  
8 }  
9
```

SOBRECARGA

- Exemplo
 - Criando uma classe com o método *main* para teste.

```
1
2 public class Principal {
3
4     public static void main (String[] args) {
5
6         CalculaPotencia calculaPotencia = new CalculaPotencia();
7
8         float resultado = calculaPotencia.calculo(3);
9
10        System.out.printf("O quadrado de 3 é %.0f \n", resultado);
11
12    }
```

SOBRECARGA

- Exemplo
 - Vamos modificar a classe CalculaPotencia para calcular qualquer potência:

```
1  
2 public class CalculaPotencia {  
3  
4     public float calculo(int numero) {  
5         return numero*numero;  
6     }  
7  
8     public float calculo(int numero, int potencia) {  
9         return numero ^ potencia;  
10    }  
11
```

SOBRECARGA

- Exemplo
 - Modificando o método *main*.

```
1
2 public class Principal {
3
4     public static void main (String[] args) {
5
6         CalculaPotencia calculaPotencia = new CalculaPotencia();
7
8         float resultado = calculaPotencia.calculo(3);
9
10        System.out.printf("O quadrado de 3 é %.0f \n", resultado);
11
12        resultado = calculaPotencia.calculo(3, 3);
13
14        System.out.printf("O cubo de 3 é %.0f \n", resultado);
15
16    }
```

SOBRECARGA

- Exemplo
 - O método calculo a ser executado será escolhido pela quantidade de parâmetros.

```
1  
2 public class CalculaPotencia {  
3  
4     public float calculo(int numero) {  
5         return numero*numero;  
6     }  
7  
8     public float calculo(int numero, int potencia) {  
9         return numero ^ potencia;  
10    }  
11
```



**INSTITUTO
FEDERAL**
Norte de Minas Gerais

Dúvidas?



Exercício

1- Crie uma classe `Emprestimo` para calcular o valor que pode ser emprestado a uma pessoa. A classe deve possuir os atributos `nomeCliente`, `Renda`, os métodos `getter` e `setter`, dois construtores, um sem parâmetros e outro com os parâmetros `nomeCliente` e `Renda`. Implementar o método `calculaEmprestimo` que deve retornar um valor de acordo com os parâmetros enviados:

Parâmetro	Retorno
Sem parâmetro	A renda * 4
Um inteiro entre 2 e 10	A renda * o parâmetro
Um Double (Uma taxa entre 1 e 15%)	A renda * 5 - a taxa(parâmetro)

Implemente uma classe `Financeira` que instancia um cliente, faz a leitura dos dados e mostra o valor disponível para empréstimo deste.