

EducaCiência FastCode

Fala Galera,

Neste artigo, abordaremos um tema muito interessante.

- Artigo: 72/2022 Data: Maio/2022
- Público-alvo: Desenvolvedores – Iniciantes ao Avançado
- Linguagem: Java
- Tema: Artigo-72 - Java Entendendo Método de Sobrecarga
- Link: https://github.com/perucello/Artigos-EducaCiencia_FastCode

Desta vez , escolhi um tema interessante, vamos abordar polimorfismo.

Métodos de Sobrecarga ou “overload” é um conceito que podemos dizer que consiste basicamente em criarmos variações objetivas de nossos Métodos (Funções) em nossa classe.

Isso proporciona a você criar métodos com os mesmos nomes que podem ser utilizados para os fins pertinentes ao desenvolvimento e função do método.

Estes fins podem ter as mesmas ou assinaturas diferentes o que teremos com variação , é o(s) parâmetros de entrada.

Não devemos confundir “Sobrecarga” com “Sobrescrita” , porém , vamos nesse artigo nos atentar à “Sobrecarga” e em um próximo artigo , abordaremos “Sobrescrita”.

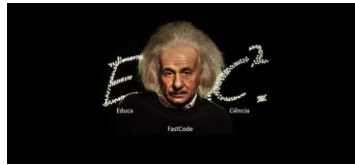
Para saber:

- Método Sobrecarga -> ou **Overload**, permite criar métodos de mesmos nomes, porém com assinaturas/argumentos diferentes.
- Método Sobrescrita -> ou **Override**, permite criar um método na nossa classe “filha” onde contém a mesma assinatura e tipo de retorno (retorno do método sobrescrito).

No Livro SCJP Sun® Certified Programmer for Java? 6 Study Guide Exam (310-065) - Kathy Sierra and Bert Bates. Pág. 109-110 faz uma abordagem extremamente interessante onde você pode ler nesse link de domínio público ou adquiri-lo.

- https://firozstar.tripod.com/_darksiderg.pdf





Vamos à pratica !

Em nosso cenário, vamos abordar método para calcular valores, vamos supor que temos que em determinado cenário calcularmos a soma, subtração, divisão e multiplicação entre dois, três, quatro e cinco atributos de entrada.

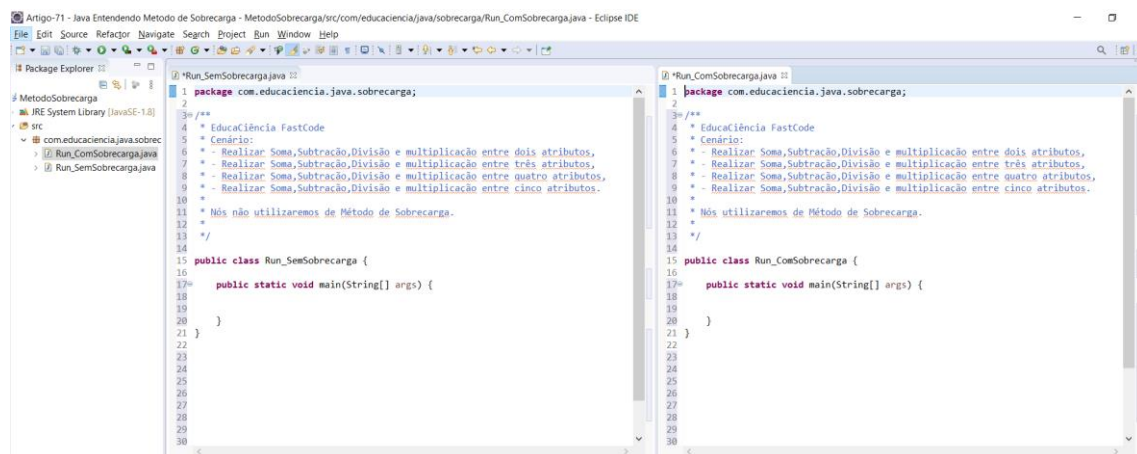
Para melhor didática de explicação, iremos criar nomes sugestivos para o seu entendimento até avançarmos ao método de Sobrecarga onde teremos funções/métodos com o mesmo nome para que possa realizar a atividade.

Isso fará que você como desenvolvedor tenha maior controle sobre a regra de negócio ao invés de ter inúmeros nomes de métodos que em algum momento poderá trazer problema para sua aplicação.

Utilize de sua IDE preferida para reproduzir o código que disponibilizaremos, porém, no nosso cenário , utilizarei da IDE Eclipse para realizar nossa atividade , lembrando que os códigos bem como o projeto estarão disponíveis no GitHub.

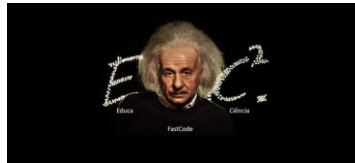
Portanto, crie seu Projeto Java e as seguintes classes como ilustrado abaixo:

- Run_SemSobrecarga
- Run_ComSobrecarga



Na classe Run_**SemSobrecarga** criaremos métodos que receberão os atributos para que possamos realizar os cálculos como proposto, utilizaremos de nomes de métodos sugerimos como mencionamos anteriormente para facilitar o entendimento.





Código:

```
Artigo-71 - Java Entendendo Metodo de Sobrecarga - MetodosSobrecarga/src/com/educaciencia/java/sobrecarga/Run_SemSobrecarga.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer
MetodosSobrecarga
  JRE System Library [JavaSE-1.8]
  src
    com.educaciencia.java.sobrecarga
      Run_SemSobrecarga.java
      Run_ComSobrecarga.java

Run_SemSobrecarga.java
14
15 public class Run_SemSobrecarga {
16
17     public static void main(String[] args) {
18
19         int n1 = 100;
20         int n2 = 20;
21         calculaDoisAtributos(n1, n2);
22     }
23
24     /** Nesse metodo, iremos realizar calculos aritmeticos de dois atributos que receberemos externamente */
25     public static void calculaDoisAtributos(int n1, int n2) {
26
27         int soma = n1 + n2;
28         int subtracao = n1 - n2;
29         int divisao = n1 / n2;
30         int multiplicacao = n1 * n2;
31         System.out.println("***** Dois atributos de Entrada *****");
32         System.out.println("A soma dos numeros é: " + soma);
33         System.out.println("A subtração dos numeros é: " + subtracao);
34         System.out.println("A divisao dos numeros é: " + divisao);
35         System.out.println("A multiplicacao dos numeros é: " + multiplicacao);
36         System.out.println("*****");
37     }
38
39
40
41
42 }
43
44

Problems Javadoc Declaration Console
<terminated> Run_SemSobrecarga [Java Application] C:\Program Files\RedHat\java-1.8.0-openjdk-1.8.0.212-3\bin\javaw.exe (7 de mai de 2022 22:35:26 - 22:35:29)
***** Dois atributos de Entrada *****
A soma dos numeros é: 120
A subtração dos numeros é: 80
A divisao dos numeros é: 5
A multiplicacao dos numeros é: 2000
*****
```

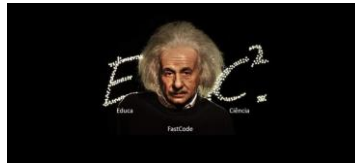
```
Artigo-71 - Java Entendendo Metodo de Sobrecarga - MetodosSobrecarga/src/com/educaciencia/java/sobrecarga/Run_SemSobrecarga.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer
MetodosSobrecarga
  JRE System Library [JavaSE-1.8]
  src
    com.educaciencia.java.sobrecarga
      Run_SemSobrecarga.java
      Run_ComSobrecarga.java

Run_SemSobrecarga.java
14
15 public class Run_SemSobrecarga {
16
17     public static void main(String[] args) {
18
19         int n1 = 100;
20         int n2 = 20;
21         int n3 = 2;
22         calculaTresAtributos(n1, n2, n3);
23     }
24
25     /** Nesse metodo, iremos realizar calculos aritmeticos de tres atributos que receberemos externamente */
26     public static void calculaTresAtributos(int n1, int n2, int n3) {
27
28         int soma = n1 + n2 + n3;
29         int subtracao = n1 - n2 - n3;
30         int divisao = n1 / n2 / n3;
31         int multiplicacao = n1 * n2 * n3;
32         System.out.println("***** Tres atributos de Entrada *****");
33         System.out.println("A soma dos numeros é: " + soma);
34         System.out.println("A subtração dos numeros é: " + subtracao);
35         System.out.println("A divisao dos numeros é: " + divisao);
36         System.out.println("A multiplicacao dos numeros é: " + multiplicacao);
37         System.out.println("*****");
38     }
39
40
41
42
43 }
44

Problems Javadoc Declaration Console
<terminated> Run_SemSobrecarga [Java Application] C:\Program Files\RedHat\java-1.8.0-openjdk-1.8.0.212-3\bin\javaw.exe (7 de mai de 2022 22:36:29 - 22:36:32)
***** Tres atributos de Entrada *****
A soma dos numeros é: 122
A subtração dos numeros é: 78
A divisao dos numeros é: 2
A multiplicacao dos numeros é: 4000
*****
```





Artigo-71 - Java Entendendo Metodo de Sobrecarga - MetodosSobrecarga/src/com/educaciencia/java/sobrecarga/Run_SemSobrecarga.java - Eclipse IDE

```

13 14
15 public class Run_SemSobrecarga {
16
17     public static void main(String[] args) {
18
19         int n1 = 100;
20         int n2 = 20;
21         int n3 = 2;
22         int n4 = 1;
23         calculaQuatroAtributos(n1, n2, n3, n4);
24     }
25
26     /** Nesse metodo, iremos realizar calculos aritmeticos de quatro atributos que receberemos externamente*/
27     public static void calculaQuatroAtributos(int n1, int n2, int n3, int n4) {
28
29         int soma = n1 + n2 + n3 + n4;
30         int subtracao = n1 - n2 - n3 - n4;
31         int divisao = n1 / n2 / n3 / n4;
32         int multiplicacao = n1 * n2 * n3 * n4;
33         System.out.println("Quatro atributos de Entrada *****");
34         System.out.println("A soma dos numeros é: " + soma);
35         System.out.println("A subtracao dos numeros é: " + subtracao);
36         System.out.println("A divisao dos numeros é: " + divisao);
37         System.out.println("A multiplicacao dos numeros é: " + multiplicacao);
38         System.out.println("*****");
39     }
40
41
42 }
43

```

Problems Javadoc Declaration Console

```

<terminated> Run_SemSobrecarga [Java Application] C:\Program Files\RedHat\java-1.8.0-openjdk-1.8.0.212-3\bin\javaw.exe (7 de mai de 2022 22:37:17 - 22:37:20)
***** Quatro atributos de Entrada *****
A soma dos numeros é: 123
A subtracao dos numeros é: 77
A divisao dos numeros é: 2
A multiplicacao dos numeros é: 4000
*****

```

Writable Smart Insert 26:5:662

Artigo-71 - Java Entendendo Metodo de Sobrecarga - MetodosSobrecarga/src/com/educaciencia/java/sobrecarga/Run_SemSobrecarga.java - Eclipse IDE

```

13 14
15 public class Run_SemSobrecarga {
16
17     public static void main(String[] args) {
18
19         int n1 = 100;
20         int n2 = 20;
21         int n3 = 2;
22         int n4 = 1;
23         int n5 = 1;
24         calculaCincoAtributos(n1, n2, n3, n4, n5);
25     }
26
27     /** Nesse metodo, iremos realizar calculos aritmeticos de cinco atributos que receberemos externamente*/
28     public static void calculaCincoAtributos(int n1, int n2, int n3, int n4, int n5) {
29
30         int soma = n1 + n2 + n3 + n4 + n5;
31         int subtracao = n1 - n2 - n3 - n4 - n5;
32         int divisao = n1 / n2 / n3 / n4 / n5;
33         int multiplicacao = n1 * n2 * n3 * n4 * n5;
34         System.out.println("***** Cinco atributos de Entrada *****");
35         System.out.println("A soma dos numeros é: " + soma);
36         System.out.println("A subtracao dos numeros é: " + subtracao);
37         System.out.println("A divisao dos numeros é: " + divisao);
38         System.out.println("A multiplicacao dos numeros é: " + multiplicacao);
39         System.out.println("*****");
40     }
41
42 }
43

```

Problems Javadoc Declaration Console

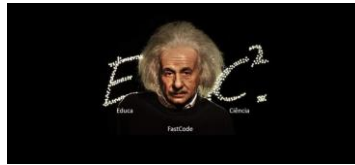
```

<terminated> Run_SemSobrecarga [Java Application] C:\Program Files\RedHat\java-1.8.0-openjdk-1.8.0.212-3\bin\javaw.exe (7 de mai de 2022 22:41:16 - 22:41:18)
***** Cinco atributos de Entrada *****
A soma dos numeros é: 125
A subtracao dos numeros é: 75
A divisao dos numeros é: 1
A multiplicacao dos numeros é: 8000
*****

```

Writable Smart Insert 27:1:677





Note que criamos quatro métodos para que pudesse receber nossos parâmetros e realizar o cálculo proposto.

```
*Run_SemSobrecarga.java
14
15 public class Run_SemSobrecarga {
16
17     public static void main(String[] args) {
18
19         int n1 = 100;
20         int n2 = 20;
21         int n3 = 2;
22         int n4 = 1;
23         int n5 = 2;
24         calculaDoisAtributos(n1, n2);
25         calculaTresAtributos(n1, n2, n3);
26         calculaQuatroAtributos(n1, n2, n3, n4);
27         calculaCincoAtributos(n1, n2, n3, n4, n5);
28
29
30         /** Nesse metodo, iremos realizar calculos aritmeticos de dois atributos que receberemos externamente*/
31         public static void calculaDoisAtributos(int n1, int n2) {}
32
33
34         /** Nesse metodo, iremos realizar calculos aritmeticos de tres atributos que receberemos externamente*/
35         public static void calculaTresAtributos(int n1, int n2, int n3) {}
36
37
38         /** Nesse metodo, iremos realizar calculos aritmeticos de quatro atributos que receberemos externamente*/
39         public static void calculaQuatroAtributos(int n1, int n2, int n3, int n4) {}
40
41
42         /** Nesse metodo, iremos realizar calculos aritmeticos de cinco atributos que receberemos externamente*/
43         public static void calculaCincoAtributos(int n1, int n2, int n3, int n4, int n5) {}
44
45     }
46 }
```

Para um projeto de desenvolvimento, poderia parecer ótimo trabalho, pois bastaria invocar o método sugestivo para que realizasse o processo.

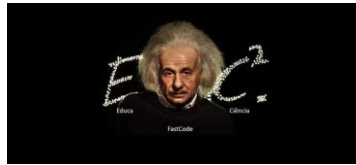
Agora, isso imaginemos que teríamos inúmeros cálculos, enfim, teríamos que ter inúmeros nomes de métodos e se por algum motivo a pessoa não conhecesse o processo, certamente causaria problema, concorda?

Portanto, é isso que iremos realizar, quando usamos do método de sobrecarga, podemos ter um mesmo nome para a função, onde facilitaria e muito para o nosso projeto!

Imagina que você tenha um único método chamado cálculo, e que independente de quantos parâmetros você passa, ele reconheceria os atributos de entrada e realizaria o cálculo proposto, não parece ótimo?

Vamos agora então, com o método de sobrecarga, alterar todo o nome que criamos no projeto e deixaremos apenas o nome “cálculo”, não é mágica, isso é sobrecarga, estaremos sobrecarregando vários métodos com o mesmo nome, porém, para cada método “cálculo” o que diferenciá-los será a quantidade (ou quais) os atributos serão passados na entrada.

Para melhor organizarmos o nosso código, iremos realizar essa atividade na classe “Run_ComSobrecarga” que criamos anteriormente.



Código:

```

/** Nesse metodo, iremos realizar calculos aritmeticos de dois atributos que receberemos externamente*/
public static void calculo(int n1, int n2) {

    int soma = n1 + n2;
    int subtracao = n1 - n2;
    int divisao = n1 / n2;
    int multiplicacao = n1 * n2;
    System.out.println("***** Dois atributos de Entrada *****");
    System.out.println("A soma dos numeros é: " + soma);
    System.out.println("A subtração dos numeros é: " + subtracao);
    System.out.println("A divisao dos numeros é: " + divisao);
    System.out.println("A multiplicacao dos numeros é: " + multiplicacao);
    System.out.println("*****");
}

/** Nesse metodo, iremos realizar calculos aritmeticos de tres atributos que receberemos externamente*/
public static void calculo(int n1, int n2, int n3) {

    int soma = n1 + n2 + n3;
    int subtracao = n1 - n2 - n3;
    int divisao = n1 / n2 / n3;
    int multiplicacao = n1 * n2 * n3;
    System.out.println("***** Tres atributos de Entrada *****");
    System.out.println("A soma dos numeros é: " + soma);
    System.out.println("A subtração dos numeros é: " + subtracao);
    System.out.println("A divisao dos numeros é: " + divisao);
    System.out.println("A multiplicacao dos numeros é: " + multiplicacao);
    System.out.println("*****");
}

/** Nesse metodo, iremos realizar calculos aritmeticos de quatro atributos que receberemos externamente*/
public static void calculo(int n1, int n2, int n3, int n4) {

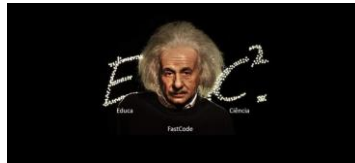
    int soma = n1 + n2 + n3 + n4;
    int subtracao = n1 - n2 - n3 - n4;
    int divisao = n1 / n2 / n3 / n4;
    int multiplicacao = n1 * n2 * n3 * n4;
    System.out.println("***** Quatro atributos de Entrada *****");
    System.out.println("A soma dos numeros é: " + soma);
    System.out.println("A subtração dos numeros é: " + subtracao);
    System.out.println("A divisao dos numeros é: " + divisao);
    System.out.println("A multiplicacao dos numeros é: " + multiplicacao);
    System.out.println("*****");
}

/** Nesse metodo, iremos realizar calculos aritmeticos de cinco atributos que receberemos externamente*/
public static void calculo(int n1, int n2, int n3, int n4, int n5) {

    int soma = n1 + n2 + n3 + n4 + n5;
    int subtracao = n1 - n2 - n3 - n4 - n5;
    int divisao = n1 / n2 / n3 / n4 / n5;
    int multiplicacao = n1 * n2 * n3 * n4 * n5;
    System.out.println("***** Cinco atributos de Entrada *****");
    System.out.println("A soma dos numeros é: " + soma);
    System.out.println("A subtração dos numeros é: " + subtracao);
    System.out.println("A divisao dos numeros é: " + divisao);
    System.out.println("A multiplicacao dos numeros é: " + multiplicacao);
    System.out.println("*****");
}

```





Note que criamos quatro métodos com o mesmo nome e nenhum deles apresentou ou retornou erro, isso apenas foi permitido porque para cada métodos, temos uma quantidade diferente de atributos de entrada (n^*).

```

12  *
13  */
14
15 public class Run_ComSobrecarga {
16
17     public static void main(String[] args) {
18
19
20     }
21
22     /** Nesse metodo, iremos realizar calculos aritmeticos de dois atributos que receberemos externamente*/
23     public static void calculo(int n1, int n2) {}
24
25     /** Nesse metodo, iremos realizar calculos aritmeticos de tres atributos que receberemos externamente*/
26     public static void calculo(int n1, int n2, int n3) {}
27
28     /** Nesse metodo, iremos realizar calculos aritmeticos de quatro atributos que receberemos externamente*/
29     public static void calculo(int n1, int n2, int n3, int n4) {}
30
31     /** Nesse metodo, iremos realizar calculos aritmeticos de cinco atributos que receberemos externamente*/
32     public static void calculo(int n1, int n2, int n3, int n4, int n5) {}
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Agora, vamos passar os mesmos parâmetros de entrada e vamos executar o processo completo, ou seja, vamos rodar nossa aplicação e pedir para que execute todos os cálculos dos 4 métodos que criamos.

```

Artigo-71 - Java Entendendo Metodo de Sobrecarga - MetodoSobrecarga/src/com/educaciencia/java/sobrecarga/Run_ComSobrecarga.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
MetodoSobrecarga
JRE System Library [JavaSE-1.8]
src
com.educaciencia.java.sobrecarga
Run_ComSobrecarga.java
Run_SemSobrecarga.java
12  *
13  */
14
15 public class Run_ComSobrecarga {
16
17     public static void main(String[] args) {
18
19         int n1 = 100;
20         int n2 = 20;
21         int n3 = 2;
22         int n4 = 1;
23         int n5 = 2;
24         calculo(n1, n2);
25         calculo(n1, n2, n3);
26         calculo(n1, n2, n3, n4);
27         calculo(n1, n2, n3, n4, n5);
28     }
29
30     /** Nesse metodo, iremos realizar calculos aritmeticos de dois atributos que receberemos externamente*/
31     public static void calculo(int n1, int n2) {}
32
33     /** Nesse metodo, iremos realizar calculos aritmeticos de tres atributos que receberemos externamente*/
34     public static void calculo(int n1, int n2, int n3) {}
35
36     /** Nesse metodo, iremos realizar calculos aritmeticos de quatro atributos que receberemos externamente*/
37     public static void calculo(int n1, int n2, int n3, int n4) {}
38
39     /** Nesse metodo, iremos realizar calculos aritmeticos de cinco atributos que receberemos externamente*/
40     public static void calculo(int n1, int n2, int n3, int n4, int n5) {}
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

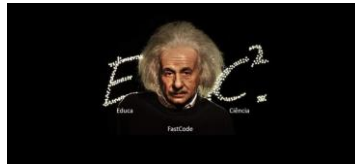
```

```

<terminated> Run_ComSobrecarga [Java Application] C:\Program Files\RedHat\java-1.8.0-openjdk-1.8.0.212-3\bin\javaw.exe (7 de mai de 2022 22:59:18 - 22:59:19)
***** Cinco atributos de Entrada *****
A soma dos numeros é: 125
A subtracao dos numeros é: 75
A divisao dos numeros é: 1
A multiplicacao dos numeros é: 8000
*****

```





Output:

```
***** Dois atributos de Entrada *****
A soma dos numeros é: 120
A subtração dos numeros é: 80
A divisao dos numeros é: 5
A multiplicacao dos numeros é: 2000
*****
***** Tres atributos de Entrada *****
A soma dos numeros é: 122
A subtração dos numeros é: 78
A divisao dos numeros é: 2
A multiplicacao dos numeros é: 4000
*****
***** Quatro atributos de Entrada *****
A soma dos numeros é: 123
A subtração dos numeros é: 77
A divisao dos numeros é: 2
A multiplicacao dos numeros é: 4000
*****
***** Cinco atributos de Entrada *****
A soma dos numeros é: 125
A subtração dos numeros é: 75
A divisao dos numeros é: 1
A multiplicacao dos numeros é: 8000
*****
```

Agora , podemos dizer que entendemos a real proposta de usarmos método de sobrecarga em Java.

Códigos estão disponíveis no link:

Link do Projeto:

- S

Link dos Artigos EducaCiência FastCode

- https://github.com/perucello/Artigos-EducaCiencia_FastCode

Espero ter colaborado de alguma maneira !

Abraços e até mais !

