**ESTUDO PARA DESENVOLVIMENTO DE SOBRECARGA DE MÉTODOS NO PROJETO CAMBIOCAPITAIS – POR: CAROLINE RESENDE SILVEIRA 165921**

* A sobrecarga, ou overload em inglês, permite a existência de vários métodos com o mesmo nome. Veremos como funciona a sobrecarga de métodos e a sua relação com lista de argumentos, tipos de retorno e modificadores de acesso.
* Métodos sobrecarregados são escritos com o mesmo nome, mas com uma lista de argumentos diferente. São geralmente usados dentro de uma mesma classe.
* Lista de argumentos diferentes:
  + Números de parâmetros diferentes

Exemplo:

public void media(Moeda venda, Outra compra)

  public void media(Moeda venda)

* + Tipos de parâmetros diferentes

Exemplo:

public void media(Moeda venda)

  public void media(Outra tipo)

* + Posições dos tipos dos parâmetros diferentes

Exemplo

public void media(Moeda venda, Outra compra)

public void media(Outra compra, Moeda venda)

\*\*Em DEITEL (2005, pág. 232), encontra-se que essa utilização das reticências, juntamente com seu tipo de dados, só pode ocorrer uma única vez e sempre no final da lista de parâmetros, como pode ser observado nos cabeçalhos dos métodos mostrarValores() e somarValores() da Listagem 2.

Listagem 2: Implementando listas de argumentos (parâmetros) de comprimento variável.

public class ExemploDois {

public static void main(String args[]) {

int nro1 = 10, nro2 = 20, nro3 = 30, nro4 = 40;

mostrarValores("1a. Chamada = ", nro1, nro2);

System.out.printf("%d\n", somarValores(nro1, nro2));

mostrarValores("2a. Chamada = ", nro1, nro2, nro3);

System.out.printf("%d\n", somarValores(nro1, nro2, nro3));

mostrarValores("3a. Chamada = ", nro1, nro2, nro3, nro4);

System.out.printf("%d\n", somarValores(nro1, nro2, nro3, nro4));

}

public static void mostrarValores(String s, int... param) {

int i, n = param.length;

System.out.printf("%s", s);

for (i=0; i<n; i++) {

if (i < (n-1))

System.out.printf("%d + ", param[i]);

else System.out.printf("%d = ", param[i]); // último valor mostrado

}

}

public static int somarValores(int... param) {

int sm = 0;

// estrutura "for" aprimorada (o identificador "nro" representa

// os valores sucessivos da lista de parâmetros "param")

for (int nro: param) {

sm = sm + nro;

}

return(sm); // resposta do método "somarValores"

}

}

**Conclusão:**

* As regras para usarmos métodos sobrecarregados são simples. O que você precisa lembrar para a usar sobrecarga de métodos é o seguinte:
* Uma classe pode fazer a sobrecarga nos métodos que foram declarados dentro dela e também nos métodos herdados.
* Métodos sobrecarregados devem possuir o mesmo nome.
* Métodos sobrecarregados devem possuir listas de parâmetros diferentes.
* Métodos sobrecarregados podem ter modificadores de acesso e tipos de retorno diferentes, mas ele não podem ser usados para diferenciar um método do outro.