

MÉTODOS



"Nunca deixe ninguém dizer que você não é capaz"

Até aqui em todos os exemplos que programamos sempre foi declarado o método `main()`, obrigatório em todas as classes executáveis. Também já utilizamos diversos métodos (matemáticos, de strings etc.). Agora vamos ver os conceitos básicos para elaboração de métodos, se tudo ocorrer estas mentes brilhantes se abrirem vamos aprender a criar nossos próprios métodos.

Métodos são trechos de código que permitem modularizar um sistema, isto é, são pequenos blocos que, juntos, compõem um sistema maior. Os métodos recebem um determinado nome e podem ser chamados várias vezes durante a execução de uma classe, ou seja, é uma sub-rotina que pode ser invocada toda vez que sua funcionalidade for necessária em um trecho da classe ou ainda a partir de uma outra classe.

Os principais motivos que levam a utilizar métodos se referem à redução do tamanho total de código de um sistema, à melhoria da modularização do sistema (cada trecho de código realiza uma tarefa) e à facilitação e agilização do processo de manutenção. Considere o método um programa autossuficiente que possui internamente todas as definições necessárias para seu funcionamento, no entanto ele compõe um sistema maior do qual é membro. O método pode ser comparado ao setor de uma empresa especializado numa determinada tarefa

Vamos ver a sintaxe da construção de um método.

```
qualificador tipo-do-retorno nome-do-método ([lista-de-parâmetros]) {  
    códigos do corpo  
    .....  
    .....  
}
```

Qualificador: conhecido também pelo nome de modificador, define a visibilidade do método. A Oracle o define como "nível de acesso" (`accessLevel`) do método. Trata-se de uma forma de especificar se o método é visível apenas à própria classe em que está declarado ou pode ser visualizado (e utilizado) por classes externas. O qualificador do método pode ser do tipo:

public: o método é visível por qualquer classe. É o qualificador mais aberto no sentido de que qualquer classe pode usar esse método.

private: o método é visível apenas pela própria classe. É o qualificador mais restritivo.

protected: o método é visível pela própria classe, por suas subclasses e pelas classes do mesmo pacote.

Tipo-de-retorno: refere-se ao tipo de dado retornado pelo método. Métodos que não retornam valores devem possuir nesse parâmetro a palavra `void`. Sempre que `void` for usada em uma declaração de método, nenhum valor é retornado após sua execução, isto é, o método atua como uma procedure de outras linguagens de programação. Um método pode ter como retorno qualquer tipo primitivo (`int`, `float`, `double` etc.), um array ou ainda um objeto qualquer.

Nome-do-método: pode ser qualquer palavra ou frase, desde que iniciada por uma letra. Se o nome for uma frase, não podem existir espaços em branco entre as palavras. Como padrão da linguagem Java, o nome de um método sempre inicia com uma palavra com letras minúsculas. Se outras palavras forem necessárias, elas devem iniciar com maiúsculas. São exemplos de nomes de métodos de acordo com o padrão da linguagem: `imprimir`, `imprimirFrase`, `gravarArquivoTexto`. É importante dar aos métodos nomes sugestivos, ou seja, que identificam facilmente a tarefa executada por eles.

Lista-de-parâmetros: trata-se de uma lista de variáveis opcionais, que podem ser recebidas pelo método para tratamento interno. Quando um método é invocado (chamado), ele pode receber valores de quem o chamou. Esses valores podem ser manipulados internamente e devolvidos ao emissor da solicitação.

```
public class Metodos_Sem_Retorno {
    public static void main(String[] args) {
        /*****
        imprimir();
        *****/
        String texto = JOptionPane.showInputDialog("Digite um texto qualquer: ");

        /*****
        imprimirTexto(texto);
        *****/

        int num1 = Integer.parseInt(JOptionPane.showInputDialog("Digite o primeiro número"));
        int num2 = Integer.parseInt(JOptionPane.showInputDialog("Digite o segundo número"));

        /*****
        somaNumeros(num1,num2);
        *****/

        num1 = Integer.parseInt(JOptionPane.showInputDialog("Digite a base: "));
        num2 = Integer.parseInt(JOptionPane.showInputDialog("Digite o expoente: "));

        /*****
        mostraExponenciacao(num1,num2);
        *****/

        char sexo = JOptionPane.showInputDialog("Digite o sexo: ").toUpperCase().charAt(0);

        /*****
        mostraSexoExtenso(sexo);
        *****/

        int numeros[] = {10,20,100,15,25,70,80,60,78,1,2,5};

        /*****
        mostraMaiorNumero(numeros);
        *****/

        String nome[] = {"Zico", "Tereza", "Carlos", "Ana", "Bruno"};
        JOptionPane.showMessageDialog(null, "Nome no vetor " + nome[0]
            + " | " + nome[1]
            + " | " + nome[2]
            + " | " + nome[3]
            + " | " + nome[4]);

        /*****
        ordenaNomes(nome);
        *****/
    }
    /***** Criação dos Métodos *****/

    public static void imprimir() {
        JOptionPane.showMessageDialog(null, "Aprendendo Java 151 ");
    }

    public static void imprimirTexto(String txt) {
        JOptionPane.showMessageDialog(null,txt);
    }

    public static void somaNumeros(int n1, int num2){
        int soma = n1 + num2;
        JOptionPane.showMessageDialog(null,"A soma de "+n1 + " mais o número " + num2
            + " o resultado é " + soma);
    }

    public static void mostraExponenciacao(int base,int expoente){
        JOptionPane.showMessageDialog(null,base + " elevado por " + expoente +
            " o resultado é: " + Math.pow(base,expoente));
    }
}
```

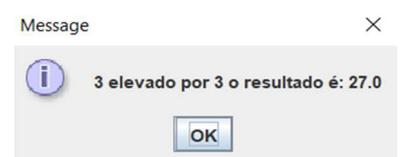
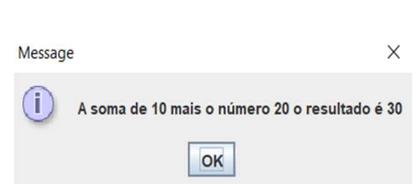
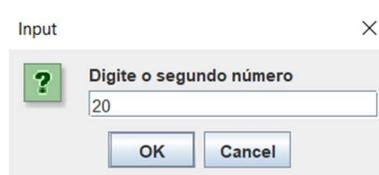
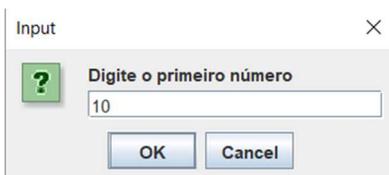
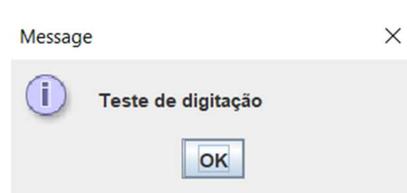
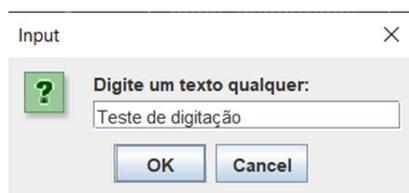
```

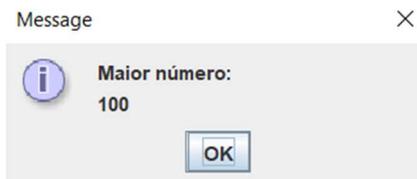
public static void mostraSexoExtenso(char sexo){
    if(sexo == 'M'){
        JOptionPane.showMessageDialog(null,"Sexo Masculino");
    } else if (sexo == 'F') {
        JOptionPane.showMessageDialog(null,"Sexo Feminino");
    } else{
        JOptionPane.showMessageDialog(null,"Sexo Desconhecido");
    }
}

public static void mostraMaiorNumero(int[] num){
    int maior=0;
    for(int i=0 ; i < num.length;i++){
        maior = Math.max(maior,num[i]);
    }
    JOptionPane.showMessageDialog(null,"Maior número: "
    +"\n"+ maior);
}

public static void ordenaNomes(String nome[]){
    Arrays.sort(nome);
    JOptionPane.showMessageDialog(null,"Nome ordenados " + nome[0]
    + " | " + nome[1]
    + " | " + nome[2]
    + " | " + nome[3]
    + " | " + nome[4]
    );
}
}
}

```

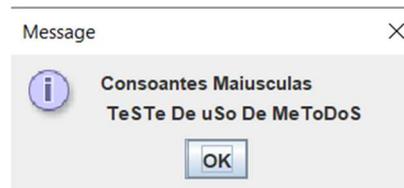
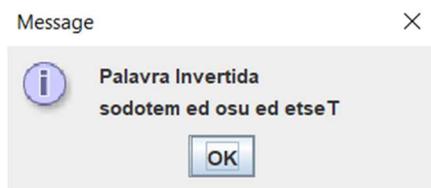
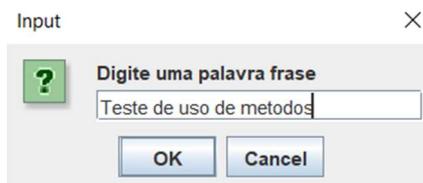




Exercício de Fixação

1 – Crie um programa onde você irá digitar uma frase/palavra qualquer a entrada será usando JOptionPane, depois você deve criar dois métodos um chamado **invertePalavra()** e outra chamada **consoantesMaiusculos()**, obvio que o primeiro método deve retornar a palavra frase invertida e o segundo método deve retornar a palavra toda escrita em minúsculas menos as consoantes independente do usuário digitar tudo maiúsculo ou minúsculos. As saídas devem ser usado também o JOptionPane.

Veja a imagem do programa rodando abaixo .



2 – Crie um programa onde o usuário irá digitar 5 nomes e 5 salários, para ser práticos vamos usar a classe Scanner(). No final dever ter 3 métodos, mostraFuncionarios(), maiorSalario() e menorSalario(). A primeira mostra o nome e salários dos funcionários, o segundo método mostra o maior salário e o terceiro claro o menor salário.

```
Digite nome funcionário: Joao da Silva  
Digite salário funcionário: 2500  
Digite nome funcionário: Carlos Alberto  
Digite salário funcionário: 2560  
Digite nome funcionário: Maria Eduarda  
Digite salário funcionário: 5600  
Digite nome funcionário: Maria Tereza  
Digite salário funcionário: 3560  
Digite nome funcionário: Pedro Luiz  
Digite salário funcionário: 3200
```

```
***** Método mostraFuncionarios() *****  
Nome : JOAO DA SILVA | Salário 2500.0  
Nome : CARLOS ALBERTO | Salário 2560.0  
Nome : MARIA EDUARDA | Salário 5600.0  
Nome : MARIA TEREZA | Salário 3560.0  
Nome : PEDRO LUIZ | Salário 3200.0
```

```
***** Método mostraMaior() *****  
Funcionário com maior salário  
Nome : MARIA EDUARDA  
Salário: 5600.0
```

```
***** Método mostraMenor() *****  
Funcionário com menor salário  
Nome : JOAO DA SILVA  
Salário: 2500.0
```