

# Curso Java COMPLETO

**Capítulo: Outros tópicos básicos sobre Java**

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

## Restrições e convenções para nomes

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

## Restrições para nomes de variáveis

- Não pode começar com dígito: use uma letra ou \_
- Não usar acentos ou til
- Não pode ter espaço em branco
- Sugestão: use nomes que tenham um significado

Errado:

```
int 5minutes;  
int salário;  
int salario do funcionario;
```

Correto:

```
int _5minutes;  
int salario;  
int salarioDoFuncionario;
```

## Convenções

- Camel Case: lastName
  - pacotes
  - atributos
  - métodos
  - variáveis e parâmetros
- Pascal Case: ProductService
  - classes

```
package entities;

public class Account {

    private String holder;
    private Double balance;

    public Account(String holder, Double balance) {
        this.holder = holder;
        this.balance = balance;
    }

    public String getHolder() {
        return holder;
    }

    public void deposit(double amount) {
        balance += amount;
    }

    public void withdraw(double amount) {
        balance -= amount;
    }
}
```

# Operadores bitwise

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

## Operadores bitwise

Operador	Significado
<b>&amp;</b>	Operação "E" bit a bit
<b> </b>	Operação "OU" bit a bit
<b>^</b>	Operação "OU-exclusivo" bit a bit

C1	C2	C1 E C2
F	F	F
F	V	F
V	F	F
V	V	V

C1	C2	C1 OU C2
F	F	F
F	V	V
V	F	V
V	V	V

C1	C2	C1 XOR C2
F	F	F
F	V	V
V	F	V
V	V	F

## Demo

(89) 0101 1001 }  
(60) 0011 1100 }  
                    &: 0001 1000 (24)  
                    |: 0111 1101 (125)  
                    ^: 0110 0101 (101)

```
int n1 = 89;  
int n2 = 60;  
System.out.println(n1 & n2);  
System.out.println(n1 | n2);  
System.out.println(n1 ^ n2);
```

## Aplicação comum: verificar bit

(89)	0101	1001	}	&: 0000 0000	(0)
(32)	0010	0000			
(113)	0111	0001	}	&: 0010 0000	(32)
(32)	0010	0000			

```
package course;

import java.util.Scanner;

public class Program {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int mask = 0b100000;
        int n = sc.nextInt();

        if ((n & mask) != 0) {
            System.out.println("6th bit is true!");
        }
        else {
            System.out.println("6th bit is false");
        }

        sc.close();
    }
}
```

# Funções interessantes para String

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

## Checklist

- Formatar: `toLowerCase()`, `toUpperCase()`, `trim()`
- Recortar: `substring(inicio)`, `substring(inicio, fim)`
- Substituir: `Replace(char, char)`, `Replace(string, string)`
- Buscar: `IndexOf`, `LastIndexOf`
- `str.Split(" ")`

```

String original = "abcde FGHIJ ABC abc DEFG ";

String s01 = original.toLowerCase();
String s02 = original.toUpperCase();
String s03 = original.trim();
String s04 = original.substring(2);
String s05 = original.substring(2, 9);
String s06 = original.replace('a', 'x');
String s07 = original.replace("abc", "xy");
int i = original.indexOf("bc");
int j = original.lastIndexOf("bc");

System.out.println("Original: -" + original + "-");
System.out.println("toLowerCase: -" + s01 + "-");
System.out.println("toUpperCase: -" + s02 + "-");
System.out.println("trim: -" + s03 + "-");
System.out.println("substring(2): -" + s04 + "-");
System.out.println("substring(2, 9): -" + s05 + "-");
System.out.println("replace('a', 'x'): -" + s06 + "-");
System.out.println("replace('abc', 'xy'): -" + s07 + "-");
System.out.println("Index of 'bc': " + i);
System.out.println("Last index of 'bc': " + j);

```

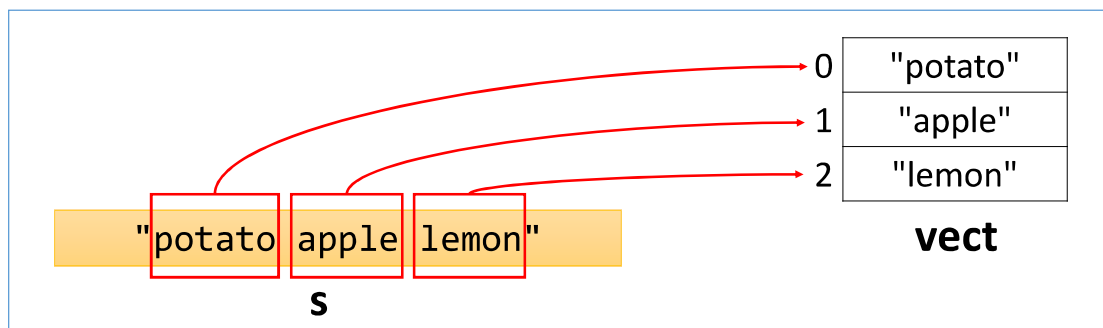
## Operação split

```

String s = "potato apple lemon";

String[] vect = s.split(" ");
String word1 = vect[0];
String word2 = vect[1];
String word3 = vect[2];

```



# Comentários em Java (básico)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

```
package course;

import java.util.Locale;
import java.util.Scanner;

/*
Este programa calcula as raízes de uma equação do segundo grau

Os valores dos coeficientes devem ser digitados um por linha
*/
public class Program {
    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        double a, b, c, delta;

        System.out.println("Digite os valores dos coeficientes:");
        a = sc.nextDouble();
        b = sc.nextDouble();
        c = sc.nextDouble();

        delta = b * b - 4 * a * c; // cálculo do valor de delta
```



# Funções (sintaxe)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

## Funções

- Representam um processamento que possui um significado
  - `Math.sqrt(double)`
  - `System.out.println(string)`
- Principais vantagens: modularização, delegação e reaproveitamento
- Dados de entrada e saída
  - Funções podem receber dados de entrada (parâmetros ou argumentos)
  - Funções podem ou não retornar uma saída
- Em orientação a objetos, funções em classes recebem o nome de "métodos"

# Problema exemplo

Fazer um programa para ler três números inteiros e mostrar na tela o maior deles.

## Exemplo:

```
Enter three numbers:  
5  
8  
3  
Higher = 8
```

```
package course;  
  
import java.util.Scanner;  
  
public class Program {  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("Enter three numbers:");  
        int a = sc.nextInt();  
        int b = sc.nextInt();  
        int c = sc.nextInt();  
  
        if (a > b && a > c) {  
            System.out.println("Higher = " + a);  
        } else if (b > c) {  
            System.out.println("Higher = " + b);  
        } else {  
            System.out.println("Higher = " + c);  
        }  
  
        sc.close();  
    }  
}
```

```
package course;

import java.util.Scanner;

public class Program {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter three numbers:");
        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();

        int higher = max(a, b, c);

        showResult(higher);

        sc.close();
    }

    public static int max(int x, int y, int z) {
        int aux;
        if (x > y && x > z) {
            aux = x;
        } else if (y > z) {
            aux = y;
        } else {
            aux = z;
        }
        return aux;
    }

    public static void showResult(int value) {
        System.out.println("Higher = " + value);
    }
}
```