



Técnicas de Programação e Algoritmos - TPA

– Tipos de Dados
– JAVA –



Operadores

Veremos agora os operadores da linguagem Java, que agregam importantes funcionalidades aos programas.

Eles possuem uma ordem de precedência na execução da expressão.

Para garantir a ordem de precedência desejada, agrupe as expressões com parênteses.

Operadores Aritméticos

Multiplicação e Divisão: * e /

```
int um = 3 / 2;           // divisão de inteiros gera um inteiro
float umEmeio = (float) 3 / 2; // ocorre promoção aritmética para float
double xyz = umEmeio * um; // ocorre promoção aritmética para float
```

Módulo: %

```
int resto = 7 % 2;           // resto = 1
```

Adição e Subtração: + e -

```
long l = 1000 + 4000;
double d = 1.0 - 0.01;
```

Concatenação:

```
long var = 12345;
String str = "O valor de var é " + var;
```

Na concatenação de Strings, as variáveis ou literais são promovidos a String antes:

```
String str = "O valor de var é " + Long.toString( var );
```

Operadores Lógicos de Curto Circuito: && e ||

Estes operadores não precisam testar toda a expressão.
Ele pára assim que uma das condições o satisfaça.
O retorno da expressão é um boolean

```
if( (a>10) && (b<5) ) {  
    // isso  
}  
  
if( (x==y) || (b<5) ) {  
    // aquilo  
}  
  
boolean b = x && y || z;
```

Operadores de Atribuição

Estes operadores atribuem um novo valor a uma variável ou expressão.

O operador = apenas atribui um valor.

Os operadores +=, -=, *= e /= calculam e atribuem um novo valor.

```
int i = 10;

int dois = 1;
dois += 1;      // dois = dois + 1;

int cinco = 7;
cinco -= 2;     // cinco = cinco - 2;

int dez = 5;
dez *= 2;       // dez = dez * 2;

int quatro = 12;
quatro /= 3;    // quatro = quatro / 3;
```

Algumas Funções

Potência

```
Math.pow(x, y);
```

Legenda:

x = base

y = potência

Raiz Quadrada

```
int x = 2;
```

```
Math.sqrt(x);
```

ou pode ser o próprio numero:

```
Math.sqrt(2);
```

Math é uma “biblioteca” de funções matemáticas. Nela encontramos funções para cálculos de:

-- seno, coseno, tangente, logaritmos, raiz cúbica, etc.

Variáveis de Tipos Primitivos

```
public class TiposPrimitivos {
    public static void main( String[] args ) {
        //declara um int e atribui um valor
        int idade = 25;
        //declara um float e, depois, atribui um valor
        float valor;
        valor = 1.99f;
        //declarando um boolean
        boolean verdadeiroOuFalso = false;
        verdadeiroOuFalso = true;
        //declarando um char
        char letraA = 'A';
        letraA = 65;           //valor ASCII para 'A'
        letraA = '\u0041';     //valor Unicode para 'A'
        //declarando um byte
        byte b = 127;
        //declarando um short
        short s = 1024;
        //declarando um long
        long l = 1234567890;
        //declarando um double
        double d = 100.0;
        //declaração múltipla
        int var1=0, var2=1, var3=2, var4;
    }
}
```



String

String é uma classe que manipula cadeias de caracteres
A classe String possui métodos para essas manipulações
Trabalha com Pool de Strings para economizar memória

```
String str = "Isto é uma String do Java";  
String xyz = new String("Isto é uma String do Java");  
  
if( str == xyz ) System.out.println("IGUAL");  
else System.out.println("DIFERENTE");  
  
if( str.equals( xyz ) ) {  
    //MANEIRA CORRETA DE SE COMPARAR O CONTEÚDO DAS STRINGS  
}  
  
System.out.println( "Tamanho da String: " + str.length() );  
  
System.out.println( "SubString: " + str.substring(0, 10) );  
  
System.out.println( "Caracter na posição 5: " + str.charAt(5) );
```



String - continuação

Outros métodos úteis da classe String:

```
String str = "Isto é uma String do Java";

// O método split quebra a String e várias outras,
// pelo separador desejado
String[] palavras = str.split(" ");

int i = str.indexOf("uma"); //retorna o índice da palavra na String

if( str.startsWith("Olá") || str.endsWith("Mundo!") ) {
    // testa o começo e o fim da String - retorna boolean
}

str = str.trim(); // elimina os espaços em branco no início e fim

str = str.replace('a','@'); // substitui os caracteres

// substitui uma palavra (usa expressões regulares)
str = str.replaceAll("String","Cadeia de caracteres");
```



```

public class Testando {
    public static void main(String[] args)
    {
        String str = "Isto é uma string do Java";
        String xyz = new String ("Isto é uma string do Java");
        // maneira errada de comparar string, portanto dará diferente
        if(str==xyz)
            System.out.println("IGUAL");
        else
            System.out.println("DIFERENTE");
        // maneira correta de comparar string
        if (str.equals(xyz))
            System.out.println("Tamanho: " + str.length());
            System.out.println("Substring: " + str.substring(0,10));
            System.out.println("Caracter na posição 5: " + str.charAt(5));
        // outros métodos
        //Quebra sempre que encontrar o espaço
        String palavras[]=str.split(" ");
        System.out.println("palavras: " + palavras[0]);
        System.out.println("palavras: " + palavras[1]);
        System.out.println("palavras: " + palavras[2]);
        //aponta o index
        int i = str.indexOf("uma");
        System.out.println("Indice: " + i); // o indice conta a partir do 0
        //compara o início e o fim da string
        boolean teste =(str.startsWith("Olá") || str.endsWith("Mundo"));
        System.out.println("Resultado: " + teste);
        // elimina espaços vazios no início e fim da string
        str=str.trim();
        System.out.println(str);
        //substitui caracteres
        str=str.replace('a','@');
        System.out.println(str);
        // substitui palavras
        str=str.replaceAll("string","Cadeia de caracteres");
        System.out.println(str);
    }
}

```