

Lógica de Programação

Técnico em Desenvolvimento de Sistemas

OPERADORES DE ATRIBUIÇÃO

- O operador de atribuição é utilizado para definir o valor inicial ou sobrescrever o valor de uma variável.
- Em seu uso, o operando à esquerda representa a variável para a qual desejamos atribuir o valor informado à direita.

=

OPERADORES ARITMÉTICOS

- Os operadores aritméticos são operadores binários, ou seja, funcionam com dois operandos.
- Por exemplo, a expressão " $a + 1$ " contém o operador binário "+" (mais) e os dois operandos " a " e " 1 ".

OPERADORES ARITMÉTICOS

| | |
|---|--|
| + | operador de adição |
| - | operador subtração |
| * | operador de multiplicação |
| / | operador de divisão |
| % | operador de módulo (ou resto da divisão) |

OPERADORES ARITMÉTICOS

```
int a = 2 + 2; //4
```

```
int b = 5 - 2; //3
```

```
int c = 4 * 2; //8
```

```
int d = 64 / 4; //16
```

```
int e = 5 % 2; //1
```

```
// 5 dividido por 2 dá 2 e tem resto 1;
```

```
// o operador % pega o resto da divisão inteira
```

PRECEDÊNCIA DE OPERADORES

- Os operadores possuem regras que são aplicadas nas expressões aritméticas do Java, que são as mesmas seguidas em álgebra.
- Quando dizemos que os operadores são aplicados da esquerda para a direita, estamos nos referindo à sua associatividade.

PRECEDÊNCIA DE OPERADORES

- Operadores de multiplicação, divisão e módulo são aplicadas primeiro. Por exemplo, quando aparecer uma expressão com várias dessas operações, elas serão aplicadas da esquerda para a direita.
- As operações de adição e subtração são aplicadas em seguida.

PRECEDÊNCIA DE OPERADORES

| Operador | Operação | Ordem de avaliação(precedência) |
|----------|--------------------------------|--|
| * / % | Multiplicação Divisão Resto | Avaliado primeiro. Se houver vários operadores desse tipo serão avaliados da esquerda para a direita |
| + - | Adição Subtração | Avaliado em seguida. Se houver vários operadores desse tipo, serão avaliados da esquerda para a direita. |
| = | Atribuição | Avaliado por último |

OPERADORES DE INCREMENTO E DECREMENTO

- Basicamente temos dois deles:
 - ++
 - --
- Podem ser declarados antes ou depois da variável .
- ++ incrementa em 1 o valor da variável.
- -- decrementa em 1 o valor da variável.

OPERADORES DE IGUALDADE

- Verificam se o valor ou o resultado da expressão lógica à esquerda é igual (“==”) ou diferente (“!=”) ao da direita, retornando um valor booleano.

| | |
|----|--|
| == | Utilizado quando desejamos verificar se uma variável é igual a outra. |
| != | Utilizado quando desejamos verificar se uma variável é diferente de outra. |

OPERADORES DE IGUALDADE

- Tem que ser prestado muita atenção ao operador de igualdade, o qual possui 2 sinais de igual (==) , sendo bem diferente do que possui 1 igual (=) que apenas atribui valores.

OPERADORES RELACIONAIS

- Assim como os de igualdade, avaliam dois operandos.
- Neste caso, mais precisamente, definem se o operando à esquerda é **menor**, **menor ou igual**, **maior** ou **maior ou igual** ao da direita, retornando um valor booleano.
- Todos os operadores relacionais têm o mesmo nível de precedência e são associados da esquerda para a direita.

OPERADORES RELACIONAIS

| | |
|----|--|
| > | Utilizado quando desejamos verificar se uma variável é maior que outra. |
| >= | Utilizado quando desejamos verificar se uma variável é maior ou igual a outra |
| < | Utilizado quando desejamos verificar se uma variável é menor que outra. |
| <= | Utilizado quando desejamos verificar se uma variável é menor ou igual a outra. |

OPERADORES LÓGICOS

- Representam o recurso que nos permite criar expressões lógicas maiores a partir da junção de duas ou mais expressões.
- Para isso, aplicamos as operações lógicas E (representado por “&&”) e OU (representado por “||”).

OPERADORES LÓGICOS

- Símbolo && é chamado de E. Este operador retorna true somente se os dois operandos forem true.

Tabela verdade:

| Operando1 | Operando2 | Resultado |
|-----------|-----------|-----------|
| true | true | true |
| true | false | false |
| false | true | false |
| false | false | false |

OPERADORES LÓGICOS

- Símbolo `||` é chamado de OU. Este operado retorna `true` caso tenha pelo menos um operando com o valor `true`.

Tabela verdade:

| Operando1 | Operando2 | Resultado |
|-----------|-----------|-----------|
| true | true | true |
| true | false | true |
| false | true | true |
| false | false | false |

OPERADOR DE NEGAÇÃO

- Símbolo ! é chamado de negação.
- Este operador retorna true se o operando tem o valor false, e retorna false se o operando o valor true.

Tabela verdade:

| Operando | Resultado |
|----------|-----------|
| false | true |
| true | false |

OPERADORES LÓGICOS E DE NEGAÇÃO

- <operando1> && <operando2>
- <operando1> || <operando2>
- ! <operando>

CONDICIONAL IF

- Uma condição é uma expressão que pode ser verdadeira ou falsa, ou seja um valor do tipo Booleano.
- A instrução de seleção if (se) permite a um programa tomar uma decisão com base no valor de uma condição.

CONDICIONAL IF

- As condições nas instruções if podem ser formadas utilizando os operadores de igualdade (== e !=) e operadores relacionais (>, <, >=, <=).
- Pode conter um segundo bloco composto pela instrução else (senão).

LENDO ENTRADA DO TECLADO

- Para armazenar em uma variável algo que foi digitado no teclado, utilizamos a biblioteca `java.util.Scanner`
- Para isso deve ser utilizada, antes de declarar a classe, a importação dessa biblioteca:
- `import java.util.Scanner;`

LENDO ENTRADA DO TECLADO

- Deve-se instanciar e criar um objeto Scanner dentro do método que lerá essa entrada:
- `Scanner ler = new Scanner(System.in);`

LENDO ENTRADA DO TECLADO

- Lendo um valor inteiro:

```
int n;
```

```
System.out.printf("Informe um número inteiro: ");
```

```
n = ler.nextInt();
```

LENDO ENTRADA DO TECLADO

- Lendo um valor real:

```
float preco;
```

```
System.out.printf("Informe o preço da mercadoria = R$ ");
```

```
preco = ler.nextFloat();
```


LENDO ENTRADA DO TECLADO

- Lendo um valor real:

```
double salario;
```

```
System.out.printf("Informe o salário do Funcionário = R$ ");
```

```
salario = ler.nextDouble();
```

LENDO ENTRADA DO TECLADO

- Lendo uma String, usado na leitura de palavras compostas, por exemplo, Pato Branco:

```
String s;
```

```
System.out.printf("Informe uma cadeia de caracteres:\n");
```

```
s = ler.nextLine();
```

LENDO ENTRADA DO TECLADO

- Lendo uma String, usado na leitura de palavras compostas, por exemplo, Pato Branco:

```
String s;
```

```
System.out.printf("Informe uma cadeia de caracteres:\n");
```

```
s = ler.nextLine();
```

EXERCÍCIOS

Vamos praticar!!!