

The slide features abstract green geometric shapes. On the left, a thin green triangle points downwards. On the right, a complex arrangement of overlapping green triangles and polygons in various shades of green (from light lime to dark forest green) creates a dynamic, layered effect. A thin, light gray line extends diagonally from the bottom left towards the right, passing through the green shapes.

# Operadores

Prof. Nelson Bellincanta

# Operadores e expressões

- ▶ A linguagem Java oferece um amplo conjunto de operadores destinados à realização de operações aritméticas, lógicas e relacionais, com a possibilidade de formar expressões de qualquer tipo;
- ▶ As expressões são construções que podem ser constituídas de um único valor literal, constante, variável ou chamada de método, mas também podem combinar vários desses elementos por meio dos operadores para obtenção de um resultado numérico ou lógico.

Operador de atribuição simples



# Operador de atribuição simples

- ▶ Atribuição é a operação que define o valor de uma variável por meio da avaliação do resultado de uma expressão envolvendo literais, constantes, variáveis, métodos e operações diversas;
- ▶ A atribuição simples armazena o resultado da avaliação de uma expressão em uma variável e usa a sintaxe:

`<variável> = <expressão>;`

# Operadores aritméticos



# Operadores aritméticos

Função	Sinal	Exemplo
Adição	+	$x + y$
Subtração	-	$x - y$
Multiplicação	*	$x * y$
Divisão	/	$x / y$
Resto da divisão inteira	%	$x \% y$
Sinal negativo	-	-x
Sinal positivo	+	+x
Incremento unitário	++	++x ou x++
Decremento unitário	--	--x ou x--

```
1  /*
2  Classe exemplo operadores aritméticos: OperadoresAritmeticos.java
3  IFPR – Campus Cascavel
4  Disciplina: Programação Orientada à Objetos
5  Professor: Nelson Bellincanta
6  Data da criação: 21/03/2023
7  */
8
9  public class OperadoresAritmeticos {
10     public static void main(String args []) {
11         //declaração e inicialização de variáveis
12         int a = 10;
13         int b = 3;
14         int x = 0, y = 0;
15
16         //várias operações com as variáveis
17         System.out.println("A = " + a); // Exibe o texto entre aspas concatenado com (+) o conteúdo da variável a
18         System.out.println("Y = " + b); // Exibe o texto entre aspas concatenado com (+) o conteúdo da variável b
19         System.out.println("-A = " + (-a)); // Exibe a variável a com vaor negativo
20         System.out.println("A/B = " + (a/b)); // Exibe o texto entre aspas concatenado com (+) com o resultado da expressão a/b
21         System.out.println("Resto de X por Y = " + (a%b)); //Exibe o resto da divisão
22         System.out.println("Inteiro de X por Y = " + (int) (a/b)); //Exibe apenas a parte inteira do resultado da expressão
23
24         y = ++x; // Pré-incremento: x incrementado ANTES de seu uso
25
26         System.out.println("Valor de x = " + x + ", valor y = " + y); // Exibe x=1, y=1
27
28         y = x++; // Pós-incremento: x incrementado DEPOIS de seu uso
29
30         System.out.println("Valor de x = " + x + ", valor de y = " + y); // Exibe x=2, y=1
31         System.out.println("Valor da expressão é: " + (64 * a - 12 * b) / 5.4321); //Exibe o valor da variável expressão
32     }
33 }
```

# Operadores relacionais





# Operadores relacionais

- ▶ Os operadores relacionais comparam valores literais, variáveis ou o resultado de expressões, retornando um resultado lógico verdadeiro ou falso;
- ▶ Os operadores relacionais duplos (definidos por meio de dois caracteres) não podem conter espaços em branco entre seus símbolos;

# Operadores relacionais

Função	Caractere(s) utilizado(s)	Exemplo
Igual	==	$x == y$
Diferente	!=	$x != y$
Maior que	>	$x > y$
Maior ou igual a	>=	$x >= y$
Menor que	<	$x < y$
Menor ou igual a	<=	$x <= y$

```
1  /*
2  Classe exemplo operadores relacionais: OperadoresRelacionais.java
3  IFPR – Campus Cascavel
4  Disciplina: Programação Orientada à Objetos
5  Professor: Nelson Bellincanta
6  Data da criação: 21/03/2023
7  */
8
9
10 public class OperadoresRelacionais {
11     public static void main (String args[]) {
12         int a = 15, b = 12; // Declaração e inicialização variáveis
13         System.out.println("Valores: a = " + a + ", b = " + b); //Exibe os valores das variáveis a e b
14         System.out.println("a == b -> " + (a == b)); //Exibe a operação relacional igual
15         System.out.println("a != b -> " + (a != b)); //Exibe a operação relacional diferente
16         System.out.println("a < b -> " + (a < b)); //Exibe a operação relacional menor que
17         System.out.println("a > b -> " + (a > b)); //Exibe a operação relacional maior que
18         System.out.println("a <= b -> " + (a <= b)); //Exibe a operação relacional menor igual a
19         System.out.println("a >= b -> " + (a >= b)); //Exibe a operação relacional maior igual a
20     }
21 }
```

# Operadores lógicos



# Operadores lógicos

- ▶ Os operadores lógicos conectam logicamente o resultado de diferentes expressões aritméticas ou relacionais, o que permite construir expressões compostas de várias partes e, portanto, mais complexas;
- ▶ Os operadores lógicos duplos (definidos por dois caracteres) também não podem conter espaços em branco entre seus símbolos.

# Operadores lógicos

Função	Caractere(s) utilizado(s)	Exemplo
E lógico ou AND	&&	x && y
Ou lógico ou OR		x    y
Negação ou NOT	!	!x

```
1  /*
2  Classe exemplo operadores lógicos: OperadoresLogicos.java
3  IFPR – Campus Cascavel
4  Disciplina: Programação Orientada à Objetos
5  Professor: Nelson Bellincanta
6  Data da criação: 21/03/2023
7  */
8
9
10 public class OperadoresLogicos {
11     public static void main (String args[]) {
12
13         double k = 1.5, m = -1.9;
14
15         System.out.println("Valores: k = " + k + ", m = " + m); //Exibe os valores das variáveis k e m
16         System.out.println("(k > 0) && (m < 0) -> " + ((k > 0) && (m < 0))); //Exibe a operação lógica &&
17         System.out.println("(k == m) || (m > 0) -> " + ((k == m) || (m > 0))); //Exibe a operação lógica ||
18         System.out.println("!(k == m) || (m > 0) -> " + (!(k == m) || (m > 0))); //Exibe a operação lógica !
19     }
20 }
```

# Operadores de atribuição composta





# Operadores de atribuição composta

- ▶ `x = x + 15;`
- ▶ `y = y - posAbsoluta;`
- ▶ `total = total * (valor/referencia);`

# Operadores de atribuição composta

- ▶ `x = x + 15;`
- ▶ `y = y - posAbsoluta;`
- ▶ `total = total * (valor/referencia);`

# Operadores de atribuição composta

Operador	Significado	Exemplo
+=	Adição e atribuição	a += expressão
-=	Subtração e atribuição	a -= expressão
*=	Multiplicação e atribuição	a *= expressão
/=	Divisão e atribuição	a /= expressão
%=	Divisão indireta e atribuição	a %= expressão

```
1  /*
2  Classe exemplo para operadores de atribuicao composta: OperadoresAtribuicaoComposta.java
3  IFPR – Campus Cascavel
4  Disciplina: Programação Orientada à Objetos
5  Professor: Nelson Bellincanta
6  Data da criação: 21/03/2023
7  */
8
9  public class OperadoresAtribuicaoComposta {
10     public static void main(String[] args) {
11         int x = 5; //Declaração e inicialização da variável x
12
13         //Operador de atribuição composta
14         x += 3; // x agora é 8
15         System.out.println("x = " + x); //Exibe o valor atual da variável x
16         //Operador de atribuição composta
17         x -= 2; // x agora é 6
18         System.out.println("x = " + x); //Exibe o valor atual da variável x
19         //Operador de atribuição composta
20         x *= 4; // x agora é 24
21         System.out.println("x = " + x); //Exibe o valor atual da variável x
22         //Operador de atribuição composta
23         x /= 3; // x agora é 8
24         System.out.println("x = " + x); //Exibe o valor atual da variável x
25         //Operador de atribuição composta
26         x %= 2; // x agora é 0
27         System.out.println("x = " + x); //Exibe o valor atual da variável x
28     }
29 }
```

# Operador ternário



# Operador ternário

- ▶ É um avaliador de condições que permite utilizar o valor de uma entre duas expressões diferentes, conforme o resultado da condição avaliada;
- ▶ Sua sintaxe é:
  - ▶ `<condição> ? <expressão_true> : <expressão_false>`
- ▶ Quando a condição avaliada resulta em verdadeiro (true), o operador retorna o resultado de `expressão_true`, caso contrário, quando o resultado é falso (false), é retornado o resultado de `expressão_false`.

# Operador ternário

- ▶ Sua utilização é conveniente para selecionar o valor de atribuições condicionadas a outras variáveis ou para impressão de mensagens do tipo é/não é, como nestes exemplos:
- ▶ `// dois valores reais quaisquer`
- ▶ `double x = 12.34, y = 56.78;`
- ▶ `// armazena em max o maior entre x e y`
- ▶ `double max = x>y ? x : y;`
- ▶ `// valor inteiro qualquer int a = 100;`
- ▶ `// exibe se valor a é ou não é maior que 50`
- ▶ `System.out.println("A= " + a + (a>50 ? "é" : "não é") + " maior que 50.");`

```
1  /*
2  Classe exemplo do uso operador ternario: OperadorTernario.java
3  IFPR – Campus Cascavel
4  Disciplina: Programação Orientada à Objetos
5  Professor: Nelson Bellincanta
6  Data da criação: 21/03/2023
7  */
8
9  public class OperadorTernario {
10     public static void main(String[] args) {
11         int idade = 18; //Declaração inicialização da variável idade
12
13         // Uso do operador ternário para determinar se a pessoa é maior de idade
14         String status = (idade >= 18) ? "Maior de idade" : "Menor de idade";
15
16         System.out.println(status); // Exibe o resultado da expressão
17     }
18 }
```



# Referências

- ▶ DEITEL, Harvey M.; DEITEL, Paul J. **Java: como programar**. 10ª ed. Editora Pearson Education do Brasil, 2017.
- ▶ FURGERI, Sérgio. **Java 8: Ensino Didático**. 1ª ed., São Paulo: Érica, 2015.
- ▶ JUNIOR, Peter Jandl. **Java Guia do Programador**. 4ª Edição: Atualizado para Java 16. Novatec Editora, 2021.