

Fundamentos de Java

2-13: Variáveis de Java e Tipos de Dados

Atividades Práticas

Objetivos:

- Descrever variáveis
- Descrever tipos simples de Java
- Definir operadores aritméticos
- Descrever operadores relacionais e lógicos
- Descrever operadores de atribuição

Vocabulário:

Identifique a palavra do vocabulário para cada definição a seguir.

	Unidade lexical usada para expressar uma relação, como igualdade ou maior que, entre duas expressões.
	Conjunto de dados com valores que têm características predefinidas.
	Operadores booleanos (AND, OR e NOT).
	Local na memória onde os dados de um tipo específico podem ser armazenados para recuperação e uso posteriores.
	Unidade lexical usada para executar operações matemáticas básicas usando dois operadores e retornando o resultado do cálculo matemático.

Tente/solucione:

1. No Alice 3, você criará uma animação de uma criança se exercitando. Você também declarará variáveis.
 - a. Crie um mundo com uma criança. Salve o projeto como "Criança se Exercitando".
 - b. Faça com que a criança acene. Atribua o valor do aceno a "Estou feliz".
 - c. Faça com que a criança diga "Gostaria de me exercitar hoje."
 - d. Faça com que a criança faça vários exercícios (alongamentos laterais, alongamentos frontais, saltos). Depois do exercício, faça com que a criança pare e diga: "Acabei de me exercitar."
 - e. Altere o código para que antes de a criança se exercitar, você declare uma variável do tipo Inteiro denominada numSets. Defina o valor padrão como 3. Salve sua animação.
 - f. Use esse valor para controlar quantos grupos de exercícios a criança faz.
 - g. Faça com que a criança acene e diga adeus no final da animação.

- No Alice 3, crie uma animação com um alienígena dirigindo um veículo de sua escolha. Salve o projeto como "Pouso do Alienígena". Programe o alienígena para voar e girar à medida que se aproxima lentamente do chão. Declare uma variável de local para contar o número de vezes que o alienígena deve girar e se aproximar do chão.
- No Alice 3, crie uma animação intitulada "Pássaros Voadores". Adicione dois pássaros à cena, cada um em um local aleatório, e uma árvore no centro da cena. Usando variáveis, uma instrução if-else e operadores relacionais, teste a distância entre cada pássaro e a árvore programando os pássaros para agirem de acordo com o seguinte storyboard textual (Revise os slides sobre operadores relacionais para obter dicas de programação):

```

If bird 1 is currently a shorter distance to the tree than bird 2, then,
    bird 1 flies to the tree
else
    bird 2 flies to the tree

```

- Preencha a seguinte Folha de Revisão de Sintaxe de Java:

Constructo	Sintaxe
Operadores de atribuição	
Operadores aritméticos	
Operadores de igualdade	
Operadores relacionais	
Operadores lógicos	

- Quais são os resultados do código a seguir?

```

class basicOperators2 {
    //using arithmetic operators and variables
    public static void main(String[] args) {
        int a = 1+ 3;
        int b = a * 3;
        int c = b / 4;
        int d = c - a;
        int e = -d;
        System.out.println("a = " + a);
        System.out.println("b = " + b);
        System.out.println("c = " + c);
        System.out.println("d = " + d);
        System.out.println("e = " + e);
    }
}

```

6. Quais são os resultados do código a seguir?

```
class Test {  
  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 20;  
        System.out.println("a == b = " + (a == b) );  
        System.out.println("a != b = " + (a != b) );  
        System.out.println("a > b = " + (a > b) );  
        System.out.println("a < b = " + (a < b) );  
        System.out.println("b >= a = " + (b >= a) );  
        System.out.println("b <= a = " + (b <= a) );  
    }  
}
```

7. Esse exemplo demonstra o operador NOT. Revise o código, em seguida, preencha as lacunas abaixo com "verdadeiro" ou "falso".

```
class BoolNotDemo {  
  
    public static void main(String[] args){  
        int x = 2;  
        int y = 1;  
        boolean bl;  
        bl = !(x > y); // bl is false  
        System.out.println("x is not greater than y:"+bl);  
        bl = !(y > x); // bl is true  
        System.out.println("y is not greater than x:"+bl);  
    }  
}
```

Preencha as lacunas:

x não é maior que y:

y não é maior que x:

8. Esse exemplo demonstra a sintaxe de atribuição. Revise o código, em seguida, preencha as lacunas abaixo com os resultados.

```
class AssignmentDemo2{

    public static void main(String[] args) {
        int x=5;
        int y=10;
        x += y;
        System.out.println("The += result is:"+ x);
        x -= y;
        System.out.println("The -= result is:"+ x);
        x *= y;
        System.out.println("The *= result is:"+ x);
        x /= y;
        System.out.println("The /= result is"+ x);
    }
}
```

Preencha as lacunas:

O resultado de += é:

O resultado de -= é:

O resultado de *= é:

O resultado de /= é: