

The logo for Oracle Academy. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is centered on a light gray background, which is framed by dark gray horizontal bars at the top and bottom.

ORACLE

Academy

Java Fundamentals

2-13

Tipos de Dados e Variáveis Java

ORACLE
Academy



Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

Objetivos

- Esta lição abrange os seguintes objetivos:
 - Descrever variáveis
 - Descrever tipos Java simples
 - Descrever operadores aritméticos
 - Descrever operadores relacionais e lógicos
 - Descrever operadores de atribuição



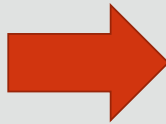
Alice 3 Versus Java

Alice 3	Java
O ambiente de programação 3D que usa representações visuais da linguagem de programação Java	Linguagem de programação; a sintaxe pode ser editada usando o IDE (Integrated Development Environment)
Usado para criar animações ou jogos interativos enquanto trabalha-se com construções de programação	Usado para criar aplicativos que são executados em qualquer plataforma, inclusive na web, usando a sintaxe Java
Interface de arrastar e soltar projetada para reduzir erros de sintaxe e fazer com que seja mais fácil aprender a programar	O IDE ajuda você a modelar objetos do mundo real, permitindo o reuso e facilitando a manutenção

Esta tabela mostra alguns dos recursos de criação de programas no Alice 3 comparados com o modo como eles são descritos em um ambiente puro Java.

Variáveis no Alice 3

- As variáveis são declaradas (criadas) no seu código
- Um aplicativo típico usa diversos valores que estão sempre mudando durante a execução do programa
- Por exemplo, no Alice 3, um carro é programado para capotar determinado número de vezes
- O valor inserido por um usuário pode ser diferente do valor inserido por outro usuário



ORACLE
Academy

JF 2-13
Tipos de Dados e Variáveis Java

Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

5

Variáveis são espaços reservados para valores que estão armazenados na memória.

Variáveis Gerenciam Valores

- As variáveis gerenciam valores inseridos por diferentes usuários no Alice 3

Uma variável é um local na memória em que dados de um tipo específico podem ser armazenados para serem recuperados e usados posteriormente pelo programa. Cada variável recebe um nome exclusivo para facilitar sua localização.

Quando você cria variáveis, lembre-se de que o nome delas deve ser descritivo e exclusivo. O uso do mesmo nome para diferentes variáveis pode gerar confusão e erros dentro do código.

Propriedades do Objeto

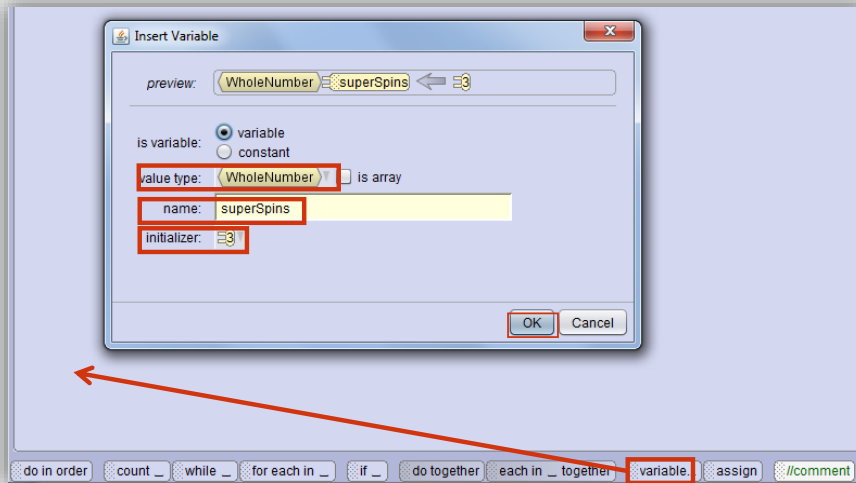
- Propriedades do objeto são variáveis que armazenam informações sobre o objeto, como cor, altura e profundidade



As propriedades de um objeto são consideradas como variáveis porque você pode alterar o valor delas no código a qualquer momento.

Declarar Variáveis no Alice 3

- Arraste o mosaico Variable até o Code editor para declarar uma nova variável



Declarar Variáveis no Java

- No Java, você declara variáveis que podem, então, ser referenciadas em outras partes do programa

```
public class Print{  
    public static void main(String[] args){  
        //variable declaration section  
        int i=1;  
        int j;  
  
        while(i<=7){  
            for(j=1; j<=i; j++)  
                System.out.println("*");  
            i=i+2;  
            System.out.println();  
        }//end while  
    }//fim do método main  
}//fim da classe Print
```

i e j são variáveis que
são declaradas aqui

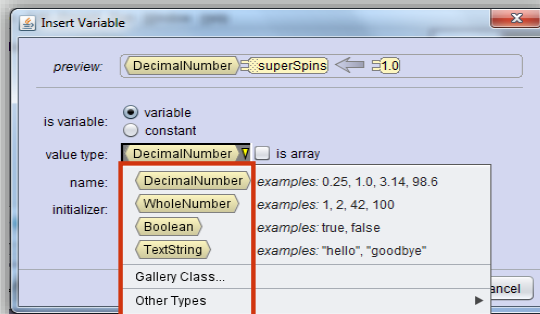
i e j são chamadas no
corpo do programa

É assim que você declara variáveis no Java fora do ambiente Alice 3. Você criará variáveis como esta quando usar o Greenfoot e o Eclipse mais adiante neste curso.

Tipos de Dados no Alice 3

- Quando você declara uma variável no Alice, precisa definir o tipo de dados

Um tipo de dados de uma variável define o tipo de valores que uma variável pode armazenar



O tipo de dados é o que define o tipo de informação que pode ser armazenada na variável.

DecimalNumber – Um número que tem uma parte decimal nele.

WholeNumber – Um número não decimal

Boolean – verdadeiro ou falso

TextString – Pode conter quaisquer caracteres alfanuméricos (letras e números). Qualquer número armazenado como String de Texto é armazenado como um caractere, e não como um valor numérico.

Tipos de Dados de Variáveis no Alice 3

Tipo de Dados	Descrição
Número Decimal	<ul style="list-style-type: none">• Executa um cálculo aritmético e define o valor dos argumentos de um procedimento• Exemplos: 0.1, 2.25, 98.6
Número Inteiro	<ul style="list-style-type: none">• Executa um cálculo aritmético e define o valor dos argumentos de um procedimento• Exemplos: 1, 459, 30
Booleano	<ul style="list-style-type: none">• Um destes dois valores: true ou false• Geralmente é o resultado de testes que compara uma coisa com a outra
Objeto	<ul style="list-style-type: none">• Qualquer objeto no Alice 3, como um gato, uma pessoa, etc
Classes	<ul style="list-style-type: none">• As classes de objetos na sua animação• Exemplos: Bípede, Cena, Quadrúpede
String de texto	<ul style="list-style-type: none">• Uma string de caracteres, como "oi" e "tchau"
Outro	<ul style="list-style-type: none">• Sons, cores e outros valores especiais

Tipos de Dados no Java - Exemplo

```
class PrintText{  
    public static void main(String[] args){  
        //variable declaration section  
        byte aByte = -10;  
        int aNumber = 10;  
        char aChar = 'b';  
        boolean isBoolean = true;  
  
        //print variables alone  
        System.out.println(aByte);  
        System.out.println(aNumber);  
        //print variables with text  
        System.out.println("aChar = " + aChar);  
        System.out.println("The boolean var is:" + isBoolean);  
    } //fim do método main  
} //fim da classe PrintText
```

Este código define variáveis usando quatro tipos Java diferentes

As variáveis são declaradas e inicializadas

As variáveis são impressas

O tipo de dados sempre deve vir antes do nome da variável na declaração. Você não pode acessar um valor por meio do tipo de dados; somente pelo nome da variável.

Tipos Java Simples

- No Java, existem oito tipos de dados básicos, denominados "tipos simples" ou "primitivos"

Tipo de Dados	Tamanho	Exemplo de Dados	Descrição dos Dados
boolean	1 bit	True, false	Armazena flags true/false
byte	1 byte (8 bits)	12, 128	Armazena números inteiros de -127 a 128
char	2 bytes	'A', '5', '#'	Armazena um único caractere Unicode
short	2 bytes	6, -14, 2345	Armazena números inteiros de -32.768 a 32.767

Pelo tipo de dados que eles armazenam, você pode ver como eles se comparam os tipos de dados do Alice 3.

Tipos Java Simples

Tipo de Dados	Tamanho	Exemplo de Dados	Descrição dos Dados
int	4 bytes	6, -14, 2345	Armazena números inteiros de - 2.147.483.648 a 2.147.483.647
long	8 bytes	3459111, 2	Armazena números inteiros de -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807
float	4 bytes	3.145, .077	Armazena um número decimal positivo ou negativo de $1,4023 \times 10^{-45}$ a $3,4028 \times 10^{+38}$
double	8 bytes	.0000456, 3.7	Armazena um número decimal positivo ou negativo de $4,9406 \times 10^{-324}$ a $1,7977 \times 10^{308}$

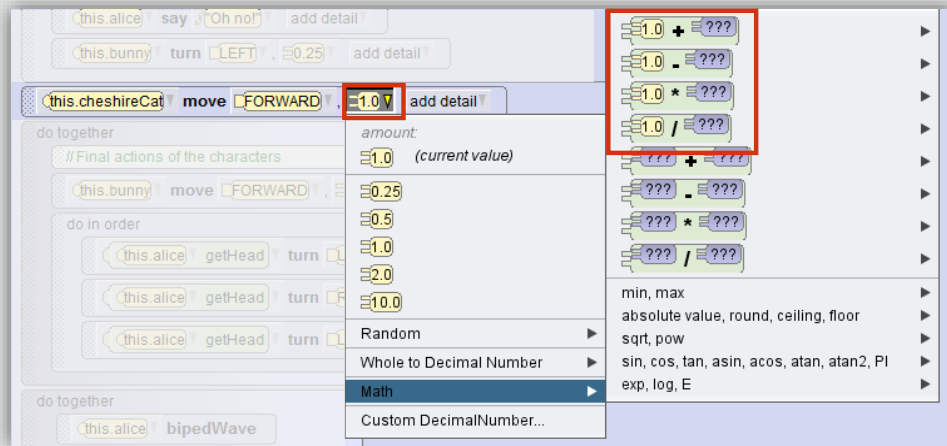
Operadores Aritméticos no Alice 3

- Use somar (+), subtrair (-), multiplicar (*) e dividir (/) para criar expressões em:
 - Argumentos de Quantidade e Duração
 - Funções getDistance
- O valor de um operador aritmético pode ser armazenado em uma variável

Operadores aritméticos podem executar operações matemáticas básicas. Eles pegam dois operandos e retornam o resultado do cálculo matemático.

Operadores Aritméticos no Argumento de Distância

- O uso da seta drop-down ao lado do valor do argumento dá acesso à função matemática



O Alice 3 pode realizar expressões matemáticas simples e complexas.

Operadores Aritméticos no Java

- Os operadores aritméticos funcionam da mesma maneira no Alice e no Java, mas são acessados de maneira diferente
- No Java, você inclui operadores aritméticos diretamente no código

```
class BasicOperators {  
    //using arithmetic operators  
    public static void main(String[] args) {  
        //variable declaration section  
        int a = 1 + 1;  
        int b = 3 * 3;  
        int c = 1 + 8 / 4;  
        int d = -2;  
        System.out.println("a = " + a);  
        System.out.println("b = " + b);  
        System.out.println("c = " + c);  
        System.out.println("d = " + d);  
    } //fim do método main  
} //fim da classe BasicOperators
```

A maneira como as equações matemáticas são usadas no Java é a mesma que na maioria das linguagens de programação. Por isso, compreender como elas funcionam dá a você um conjunto de habilidades de programação transferíveis.

Operadores Relacionais

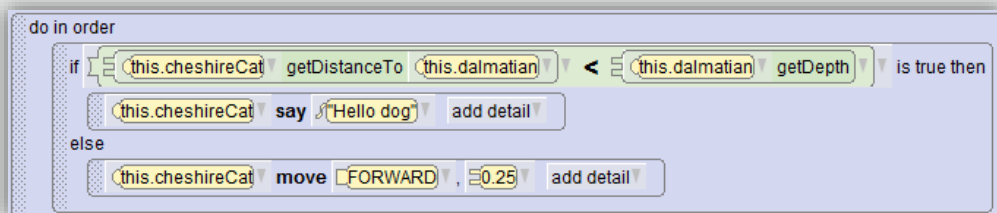
- Os operadores relacionais incluem o seguinte:
 $=$ \neq $>$ $<$ \geq \leq
- Expressões com operadores relacionais produzem valores verdadeiros ou falsos

Um operador relacional é uma unidade léxica usada para expressar uma relação, como igualdade ou maior que, entre duas expressões. Duas expressões adequadas combinadas com um operador relacional geralmente formam uma expressão relacional ou uma condição em uma linguagem de programação.

Os operadores relacionais são usados dentro das instruções de controle para avaliar a condição a fim de controlar o fluxo do programa.

Operadores Relacionais no Alice 3

- O exemplo a seguir mostra um operador relacionado usado para testar a distância entre o gato Cheshire e o Dálmata
- Se a distância do gato até o cão for menor que a profundidade do cão, o gato dirá: "Olá, cão!"
- Se a distância for menor, o gato se moverá 0,25 m para frente



Este exemplo mostra que, se a distância entre o gato Cheshire e o Dálmata for menor que a profundidade do Dálmata, o gato dirá "Olá". Se a distância não for menor que a profundidade, o gato se moverá para frente.

Operadores Relacionais no Java

Símbolo	Nome do Operador	Exemplo
==	Igual A	(A == B) é falso
!=	Diferente De	(A == B) é verdadeiro
>	Maior Que	(A > B) é falso
>=	Maior Que ou Igual A	(A >= B) é falso
<	Menor Que	(A < B) é verdadeiro
<=	Menor Que ou Igual A	(A <= B) é verdadeiro

Estes são os símbolos matemáticos tradicionais que novamente são usados na maioria das linguagens de programação.

Operadores Relacionais no Java - Exemplo

- Os operadores relacionais no Java e no Alice funcionam da mesma maneira
 - Veja a seguir um exemplo de código Java que contém operadores relacionais

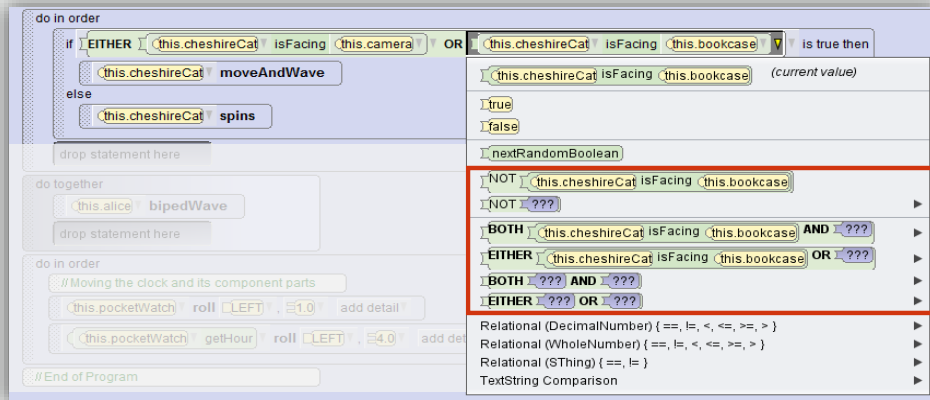
```
class Test {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 20;  
  
        System.out.println("a == b = " + (a == b) );  
        System.out.println("a != b = " + (a != b) );  
        System.out.println("a > b = " + (a > b) );  
        System.out.println("a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

Este slide usa uma instrução de saída (System.out.println()) para exibir o valor do resultado de uma expressão.

Operadores Lógicos no Alice 3

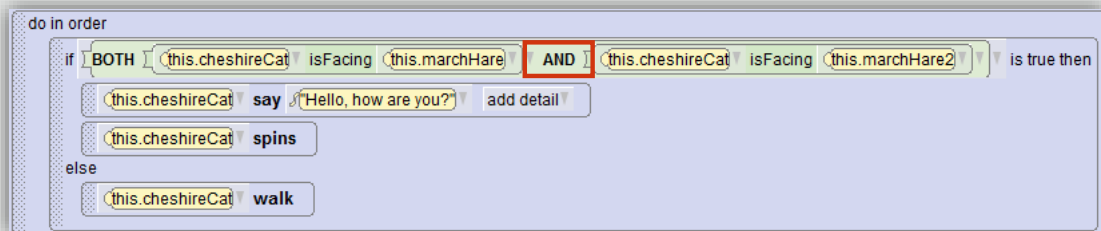
- Essas são variações de operadores lógicos no Alice 3

Os operadores lógicos são os operadores booleanos AND, OR e NOT. As expressões gravadas com operadores lógicos resultam em true ou false.



Operadores Lógicos no Alice 3 - Exemplo

- O exemplo usa os dois operadores lógicos a e b para combinar as duas condições em uma única condição
- Essas duas condições devem ser verdadeiras para que toda a condição seja verdadeira
- Caso contrário, ela será falsa



Operadores Lógicos no Java

- Os operadores lógicos funcionam da mesma maneira no Alice 3 e no Java, mas são acessados de maneira diferente
- No Java, os operadores lógicos são digitados usando símbolos

Símbolo	Nome do Operador
&&	AND Condicional ou "Operadores de curto-circuito"
 	OR Condicional ou "Operadores de curto-circuito"
!	NOT

Operadores Lógicos no Java - Exemplo

- Este exemplo demonstra o operador NOT

```
class BoolNotDemo {  
    public static void main(String[] args){  
        //variable declaration section  
        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);  
    } //fim do método main  
} //fim da classe BoolNotDemo
```

A leitura do operador NOT pode ser confusa. A linha `bl = !(x > y);` `x` é maior que `y`, o que é verdadeiro por causa do operador NOT que você está procurando é o oposto disso. Quando o valor verdadeiro é passado para o operador NOT, ele define esse valor para seu valor oposto, ou seja, falso.

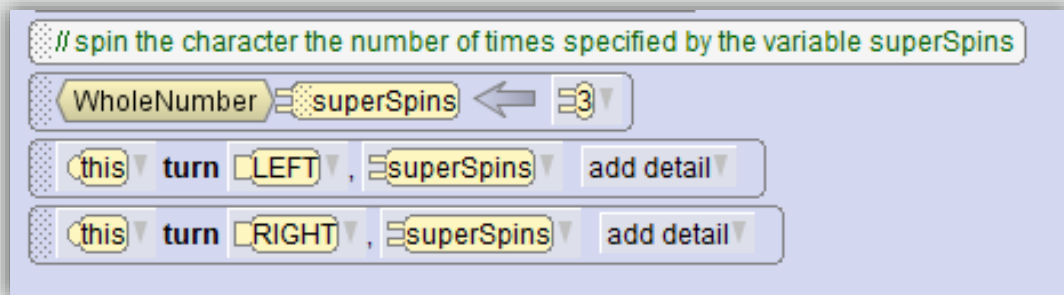
Operadores de Atribuição no Alice 3

- Os operadores de atribuição mudam o valor de uma variável
- Na interface do Alice 3, você atribui valores a propriedades e variáveis
- As propriedades de cor, opacidade, posição, veículo e tamanho são valores atribuídos nas Propriedades dos Objetos



Operadores de Atribuição no Alice 3 - Exemplo

- Neste exemplo, é atribuído um valor inicial 3 à variável local superSpins
- O número de vezes em que o personagem gira é atribuído a superSpins



Operadores de Atribuição no Java

- No Java, use o sinal de igualdade ("=") para atribuir um valor a outro
 - Neste exemplo, o valor de y é 5 o valor de z é 25

```
class AssignmentDemo{
    public static void main(String[] args) {
        //variable declaration section
        int x=5;
        int y=10;
        int z=20;
        y = x;
        z = y + z;

        System.out.println("The value of y is:"+ y);
        System.out.println("The value of z is:"+ z);
    } //fim do método main
} //fim da classe AssignmentDemo
```

No código acima, a instrução `y = x;` significa que o valor de `x` será armazenado em `y`. A variável à esquerda do sinal `=` sempre contém o valor da expressão à direita. Neste caso, embora seja atribuído o valor 10 a `y`, ele assume o valor de `x` quando essa linha é executada.

`z` recebe o valor do resultado da expressão `y + z;` e é calculado como `5 + 20`. Assim, `z` terá um valor igual a 25 quando esta linha for executada.

Lidando com Operações Padrão no Java

- O Java fornece uma sintaxe especial para lidar com operações padrão como `z = y + z;`
- A sintaxe é `z += y;`
- Este código salva alguns pressionamentos de tecla, mas atribui a `z` o valor de `y` mais `z`

A instrução `z += y;` pode ser lida como `z` sendo igual ao valor atual de `z` mais o valor de `y`.

Sintaxe de Atribuição para Outras Operações

Símbolo	Exemplo	Equivalente A
<code>+=</code>	<code>x += y</code>	<code>x = x + y;</code>
<code>- =</code>	<code>x -= y</code>	<code>x = x - y;</code>
<code>* =</code>	<code>x *= y</code>	<code>x = x * y;</code>
<code>/ =</code>	<code>x /= y</code>	<code>x = x / y;</code>

Qualquer um dos métodos resultará na mesma resposta, mas a sintaxe de atribuição faz com que você precise digitar menos.

Código da Sintaxe de Atribuição - Exemplo

```
class AssignmentDemo2{
    public static void main(String[] args) {
        //variable declaration section
        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);
    } //fim do método main
} //fim da classe AssignmentDemo2
```

Terminologia

- Estes são os principais termos usados nesta lição:
 - Operadores aritméticos
 - Operadores de atribuição
 - Tipo de dados
 - Operadores lógicos
 - Operadores relacionais
 - Variável

Resumo

- Nesta lição, você deverá ter aprendido a:
 - Descrever variáveis
 - Descrever tipos Java simples
 - Descrever operadores aritméticos
 - Descrever operadores relacionais e lógicos
 - Descrever operadores de atribuição



