

ILP-010 – Linguagem de Programação

ADS – Análise e Desenvolvimento de Sistemas
Prof. Vagner Macedo

Aula 03
Variáveis, tipos de dados e operadores

Variáveis

- Variáveis são nomes atribuídos a endereços na memória de um computador onde se guardam dados.
- A declaração de uma variável consiste em:
 - especificar qual tipo de dado será guardado;
 - dar um nome para a variável.

Declarando uma variável

- Sintaxe:
 - [Tipo] [NomeDaVariável];
- Exemplo:
 - int Idade;

Declarando uma variável

- Podemos também declarar e ao mesmo tempo atribuir um valor a uma variável.
- Exemplo:
 - `int idade=10;`

[NomeDaVariavel]

- Regras para a definição de um [NomeDaVariavel]
 - não pode ser uma palavra-reservada;
 - não pode ser “true” nem “false” - literais que representam os tipos lógicos (booleanos);
 - não pode ser “null” - literal que representa o tipo nulo;
 - não pode conter espaços em brancos ou outros caracteres de formatação;

[NomeDaVariavel]

- Regras para a definição de um [NomeDaVariavel]
 - deve ser a combinação de uma ou mais letras e dígitos UNICODE-16
 - letras de A a Z
 - letras de a a z sublinha _
 - cifrão \$
 - dígitos de 0 a 9

[NomeDaVariavel]

- Regras para a definição de um [NomeDaVariavel]
 - Caracteres compostos (acentuados) são interpretados igualmente aos não compostos (não acentuados).
 - Por exemplo, “História” e “Historia” são o mesmo identificador.
 - Letras maiúsculas e minúsculas diferenciam os identificadores.
 - Por exemplo, “a” é um identificador diferente de “A”, História é diferente de história, etc.

[NomeDaVariavel]

- Regras para a definição de um [NomeDaVariavel]
 - Para variáveis com nomes compostos, costuma-se utilizar a “regra do camelo” (“camel case”), concatenando as palavras e deixando as iniciais de cada uma em maiúsculo.
 - Exemplos: dataPagamento, numeroCartaoCredito etc

[NomeDaVariavel]

- Lista de palavras reservadas (*keywords*):

abstract	continue	for	new	switch
assert	default	goto *	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const *	float	native	super	while
* not used				

Obs: de acordo com a *Java Language Specification*, as palavras *null*, *true* e *false* são consideradas valores literais reservados, e não keywords. Porém, assim como as keywords, também não podem ser utilizados para criar identificadores.

Tipo de dados

- Tipos Primitivos:
 - boolean
 - Pode assumir o valor true ou o valor false.
 - boolean c;
 - boolean ligado = true;

Tipo de dados

- Tipos Primitivos:
 - char
 - Caractere em notação Unicode de 16 bits. Serve para a armazenagem de dados alfanuméricos. Também pode ser usado como um dado inteiro com valores na faixa entre 0 e 65535.
 - `char letra = 'A';`
 - `char letra = '\u0041';`

Tipo de dados

- Tipos Primitivos:
 - byte
 - É o tipo de dado capaz de armazenar 8 bits de informação, ou seja, um número inteiro entre -128 e 127. Sua utilização é recomendada em caso de economia de memória.
 - byte a;
 - byte b = '1';
 - byte z = 111;
 - byte pato = 0xA;
 - byte seven = 07;

Tipo de dados

- Tipos Primitivos:
 - short
 - É o tipo de dado que é capaz de armazenar números inteiros de 16 bits, ou seja, um número inteiro entre -32.768 e 32.767.
 - short a;
 - short by1 = -32;
 - short by2 = 0XBB;

Tipo de dados

- Tipos Primitivos:
 - int
 - Inteiro de 32 bits. Pode assumir valores entre -2.147.483.648 e 2.147.483.647.
 - int a;
 - int by1 = -32;
 - int by2 = 0XBB;

Tipo de dados

- Tipos Primitivos:
 - long
 - Inteiro de 64 bits. Pode assumir valores entre
 - -9.223.372.036.854.775.808 e
 - 9.223.372.036.854.775.807.
 - long a;
 - long bwy1 = -32L;
 - long byz2 = 32l;

Tipo de dados

- Tipos Primitivos:

- float

- É o tipo de dado capaz de armazenar números reais de precisão simples, ou seja, 32 bits de informação representando um número real.

- float a;
 - float by1 = -32.0;
 - float bz2 = 32.2F;
 - float bz = 32.455f;
 - float bze = 1.32455e4f;

Tipo de dados

- Tipos Primitivos:

- double

- É o tipo de dado capaz de armazenar números reais de precisão dupla, ou seja, 64 bits de informação em forma de número real. É usado para representar valores nos quais é preciso uma precisão maior que a de float.

- double a;

- double by1 = -32.0;

- double bz2 = 32.2d;

- double bz = 32.455D;

- double bze = 1.32455e4D;

Operadores

- Os operadores são sinais que representam atribuições, cálculos e ordem dos dados.
- As operações seguem uma ordem de prioridades, ou seja, alguns cálculos (ou outros) são processados antes de outros.
 - 1. Divisão e multiplicação;
 - 2. Soma e subtração.
- Separadores
 - Os separadores são sinais que indicam/modificam a ordem das operações.
 - Em Java, temos o “()” parênteses como separador.

Operadores

- Operador de atribuição
 - =
- Operadores de igualdade
 - ==, !=
- Operadores relacionais
 - <, <=, >, >=
- Operadores de adição, subtração, multiplicação, divisão e módulo
 - +, -, *, /, %
- Atribuição com adição, subtração, multiplicação, divisão e módulo
 - +=, -=, *=, /=, %=
- Incremento e Decremento
 - ++, --

Operadores

- Operador de atribuição
 - =
- Operadores de igualdade
 - ==, !=
- Operadores relacionais
 - <, <=, >, >=
- Operadores de adição, subtração, multiplicação, divisão e módulo
 - +, -, *, /, %
- Atribuição com adição, subtração, multiplicação, divisão e módulo
 - +=, -=, *=, /=, %=

Operadores

- Incremento e decremento
 - `++`, `--`
- Operador lógico de negação
 - `!`

Operadores

- Exemplos:

- $x += 5 \rightarrow x = x + 5$
- $x -= y \rightarrow x = x - y$
- $x *= 2 \rightarrow x = x * 2$
- $z /= 4 \rightarrow z = z / 4$
- $x++ \rightarrow x = x + 1$

Exercícios

- 3.1. Crie um novo projeto e defina algumas variáveis. Tente nomear alguma variável de forma ilegal, ou seja, usando uma palavra reservada e veja que erro é produzido.
- 3.2. No projeto do exercício 1, corrija o nome das variáveis. Agora exiba o valor de alguma variável que não teve seu conteúdo definido e verifique o que ocorre.
- 3.3. Teste os códigos abaixo e entenda o que está ocorrendo. Qual a diferença entre eles?
 - Código 1
 - int i = 10;
 - int n = i++;
 - System.out.println(n);
 - Código 2
 - int i = 10;
 - int n = ++i;
 - System.out.println(n)

Exercícios

- 3.4. Declare variáveis de todos os tipos primitivos e as utilize com vários operadores. Em cada etapa de operação, imprima o valor da variável utilizada, deixando claro o que está sendo feito.
 - Exemplo:
 - int altura = 5;
 - int largura = 10;
 - System.out.println("Aqui a altura vale " + altura + " e a largura " + largura);
 - System.out.println("altura x largura = " + altura*largura);

Exercícios

- 3.5. Faça um programa que declare uma variável inteira que armazena o ano de seu nascimento. Declare uma variável e atribua a ela o ano atual. Mostre na tela a sua idade utilizando o operador de subtração e as duas variáveis declaradas.
- 3.6. Faça um programa que declare três variáveis: largura, altura e profundidade. Mostre na tela o volume, definido como a multiplicação das três variáveis.
- 3.7. Faça um programa que declare uma variável que será inicializada com o ano atual. Mostre na tela se o ano é bissexto ou não utilizando o que aprendeu sobre operadores. Pesquise na internet como calcular se um ano é bissexto.
- 3.8. Faça um programa que declare as três notas que você espera obter em LPG. Mostre na tela a média das três notas.

Obrigado!

