



FUNDAMENTOS DE PROGRAMAÇÃO

Conceitos Iniciais (Variáveis,
Tipos Primitivos, Operadores)

VARIAVEIS

Variaveis servem para guardar dados, sejam caracteres(literais), numéricos ou lógicos.

Variavel em programação é um espaço alocado na memória de um dispositivo, ou seja é um local reservado em algum local na memória do computador para um certo dado ou valor.

Exemplo declaração:

- <tipo primitivo> <identificador>;
- <tipo primitivo> numero;
- <tipo primitivo> nome;

IDENTIFICADOR

Identificadores são nomes que usamos para identificar as nossas variáveis. Para que um identificador seja válido é necessário seguir algumas regras:

- Pode:
 - Podem começar com letras;
 - É possível usar letras com números;
- Não Pode:
 - Não pode começar com números;
 - Não podemos usar acentos;
 - Não pode conter espaços;
 - Não pode ser palavra reservada.

IDENTIFICADOR(CONTINUAÇÃO)

Dicas para escolha de identificadores:

- Tente escolher nomes coerentes para as variáveis;
- Maísculas e minúsculas fazem diferença;

TIPOS PRIMITIVOS

Tipos Primitivos é tipo de valor que uma determinada variavel declarada pode ter. Temos três tipos primitivos que são:

- Números(byte, short, int, long, float, double);
- Caracteres(char, string);
- Lógicos(boolean).

Os **tipos numéricos** são valores que podem ser efetuadas operações de adição, subtração, multiplicação, divisão, etc.

Os tipos caracteres são dados que representam uma sequência de símbolos, como nomes, instituições, locais, endereços, classificação alfabetica, etc.

TIPOS PRIMITIVOS(CONTINUAÇÃO)

Tipo Caractere

Para o tipo caractere temos:

Char que aceita apenas um valor(letra) que está delimitado por aspas simples e **String** que aceita valores como frases ou sequências de caracteres que são delimitados por aspas duplas.

Tipo Boolean(Lógico)

É um tipo que aceita apenas dois bits que são verdadeiro ou falso(true ou false).

TIPOS PRIMITIVOS — TIPOS NÚMERICOS

Tipo	Tamanhos (bits)	Valores Mínimos	Valores Máximos
byte	8	-128	127
short	16	-32768	32767
int	32	-2,147,483,648 (-2^{31})	2,147,483,647 ($2^{31} - 1$)
long	64	-9,223,372,036,854,775,808 (-2^{63})	9,223,372,036,854,775,807 ($2^{63} - 1$)

TIPOS PRIMITIVOS — TIPOS NÚMERICOS (COM PONTO FLUTUANTE)

Float	32	24	8	6-7
Double	64	53	11	15-16

OPERADORES

Operadores Aritméticos – são operadores utilizados para fazer alguma operação entre valores numéricos.

Operadores Relacionais ou Comparação – são operadores utilizados para fazer comparação entre valores do mesmo tipo. O seu retorno é um valor booleano.

Operadores Lógicos – são operadores utilizados geralmente com os operadores relacionais comparando valores booleano(true ou false) de forma a obter um único valor booleano.

OPERADOR DE ATRIBUIÇÃO

O operador de atribuição usamos para atribuir valores a uma variável. Maior parte das linguagens de programação usa o operador (=) para atribuir valores a variáveis e a linguagem Java não é exceção.

Exemplo:

```
int num1 = 10;
```

```
String nome = "Diana";
```

```
int soma = 1 + 1 2
```

Um sinal de igual em java(=) lê-se recebe ou atribuímos um valor tal a uma variável declarada.

Exercício

Declare uma variável do tipo char e atribua um valor a essa variável

OPERADORES ARITMETICOS

Operadores	Descrição
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo(Resto da divisão)

OPERADORES ARITMETICOS EXEMPLOS

`int soma = 5 + 2;` \longrightarrow 7

`int subtrai = 5 - 2;` \longrightarrow 3

`int multiplica = 5 * 2;` \longrightarrow 10

`float divide = 5/2;` \longrightarrow 2.5


`int restoDivisao = 5%2;` \longrightarrow 1

Operadores de incremento e decremento:

`int i = i + 1` \longrightarrow `i++`

`int i = i - 1` \longrightarrow `i--`

OPERADORES ARITMETICOS(PRECEDÊNCIA)



Ordem	Operadores
primeiro	()
segundo	*, /, %
terceiro	+, -

OPERADORES RELACIONAIS

Operadores	Descrição
==	Igual
>	Maior
<	Menor
>=	Maior ou Igual
<=	Menor ou Igual
!=	Diferente

OPERADORES RELACIONAIS EXEMPLOS

$5 > 2 \longrightarrow$ `true`

$7 < 4 \longrightarrow$ `false`

$8 \geq 8 \longrightarrow$ `true`

$9 \leq 7 \longrightarrow$ `false`

$5 == 5 \longrightarrow$ `true`

$4 != 4 \longrightarrow$ `false`

NEGAÇÃO

p	Não(!)
false	true
true	false


CONJUNÇÃO(E)

p	q	E(&&)
false	false	false
false	true	false
true	false	false
true	true	true

DISJUNÇÃO(OU)

p	q	Ou()
false	false	false
false	true	true
true	false	true
true	true	true

OPERADORES LÓGICOS(PRECEDÊNCIA)



Ordem	Operadores
primeiro	Negação(!)
segundo	Conjunção(&&)
terceiro	Disjunção()



FIM

Feito por:

- Anselmo Nhamage