



Tipos de Dados & Variáveis & Constantes

Prof. Nelson bEllincanta

Tipos de dados primitivos



Tipos de dados primitivos

- ▶ Um tipo de dados estabelece um conjunto particular de valores que podem ser representados dentro de um programa e para os quais podem ser definidas operações específicas;
- ▶ Quando essas operações fazem parte da linguagem, é dito que constituem um tipo primitivo ou predefinido;
- ▶ No Java antes de utilizar variáveis é necessário definir que tipo de dado elas vão armazenar.

Tipos de dados primitivos

Categoria	Tipo	bits	Faixa de valores
Inteiro	byte	8	-128 a + 127
	short	16	-32.768 a + 32.767
	int	32	-2.147.483.648 a +2.147.483.647
	long	64	-9.223.372.036.854.775.808 a +9.223.372.036.854.775.807
Real	float	32	Valores positivos: de +1.40129846432481707e-45 a +3.40282346638528860e+38 Valores negativos: de -3.4028234663852886e+38 a -1.40129846432481707e-45
	double	64	Valores positivos: de +4.94065645841246544e-324 a +1.79769313486231570e+308 Valores negativos: de -1.7976931348623157e+308 a -4.94065645841246544e-324
Caractere	char	16	'\u0000' a '\uFFFF'
Lógico	boolean	8	false e true

Cadeia de caracteres

- ▶ A combinação de um ou mais caracteres em sequência é uma cadeia de caracteres ou uma *string*;
- ▶ É representada no Java por objetos da classe *String*, ou seja, não é um tipo primitivo;
- ▶ As cadeias de caracteres podem ser vazias (como " "), ou conter um ou mais caracteres, até o limite dado por Integer.MAX_VALUE, ou seja 2³¹ caracteres.

```
1  /*
2  Classe exemplo uso do Integer.MAX_VALUE: MaximumInteger.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 MaximumInteger {
10
11      public static void main(String[] args) {
12          //retorna o maior valor que um int pode representar, ou seja, (2 elevado a 31) – 1, que é igual a 2147483647.
13          System.out.println(Integer.MAX_VALUE);
14      }
15  }
```

Tipos de dados cadeia de caracteres

- ▶ Valores do tipo String são instâncias desta classe;
- ▶ Tipo de classe especial onde instância pode ser declarada como tipos simples:
 - ▶ Ex.: String nome = “Aluno”;
- ▶ Comparação
 - ▶ <string1>.equals(<string2>);
 - ▶ <string1>.equalsIgnoreCase(<string2>);

```
1  /*
2  Classe exemplo cadeia de caracteres: CadeiaCaracteres.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 CadeiaCaracteres {
11     public static void main(String args[]){
12         //Declaração e inicialização de variáveis
13         String nome = "Nelson",
14             sobrenome = " Bellincanta";
15         int teste = 10;
16         String nomeCompleto = nome + sobrenome;
17         // Exibe string Nome completo: concatenado (+) com valor da variável nomeCompleto
18         System.out.println("Nome completo: " + nomeCompleto);
19         // Exibe string Nome completo: concatenado (+) com valor das variáveis nomeCompleto e teste
20         System.out.println("Nome completo: " + nomeCompleto + teste);
21         // Exibe string Tamanho String: concatenado (+) com o tamanho da variável nomeCompleto
22         System.out.println("Tamanho String: " + nomeCompleto.length());
23         // Exibe o resultado da comparação das strings
24         System.out.println("Igual a Nelson Mandela: "
25             + nomeCompleto.equals("Nelson Mandela"));
26         // Exibe o resultado da comparação das strings desconsiderando letras maiúsculas e minúsculas
27         System.out.println("Igual a NELSON BELLINCANTA: "
28             + nomeCompleto.equalsIgnoreCase("NELSON BELLINCANTA"));
29     }
30 }
```


Caracteres Especiais

- ▶ Um caractere especial ou caractere de controle é um símbolo que produz um efeito próprio, mas que não possui uma forma visual;
- ▶ São usados na formatação de mensagens e dados e representados pela combinação de uma \ (barra invertida) com um caractere extra, formando as sequências de escape (escape sequences).

Caracteres Especiais

Representação	Significado
\n	Pula linha(new line ou line feed)
\r	Retorno de carro (carrige return)
\b	Retrocesso (backspace)
\t	Tabulação (horizontal tabulation)
\f	Nova página (form feed)
\'	Apóstrofe
\"	Aspas
\\	Barra invertida

<http://docs.oracle.com/javase/specs/jls/se7/html/jls-3.html#jls-3.10.6>

Conversão de tipos



Conversão de tipos

Supondo a variável x	Converter em	y recebe o valor convertido
int x = 10	float	float y = (float) x
int x = 10	double	double y = (double) x
float x = 10.5	int	int y = (int) x
String x = "10"	int	int y = Integer.parseInt(x)
String x = "20.54"	float	float y = Float.parseFloat(x)
String x = "20.54"	double	double y = Double.parseDouble(x)
String x = "Java"	Vetor de bytes	byte b[] = x.getBytes()
int x = 10	String	String y = String.valueOf(x)
float x = 10.35	String	String y = String.valueOf(x)
double x = 254.34	String	String y = String.valueOf(x)
byte x[] - (x é um vetor de bytes)	String	String y = new String(x)

```
1  /*
2  Classe exemplo para realizar a conversão de tipos de dados primitivos: ConversaoDeTipos.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 ConversaoDeTipos
10 {
11     public static void main(String[] args)
12     {
13         String s1 = "10"; //Declaração e inicialização de variáveis
14         int v = Integer.parseInt(s1); //Converte a variável s1 em um tipo inteiro
15         float x = Float.parseFloat(s1); //Converte a variável s1 em um tipo float
16         double y = Double.parseDouble(s1); //Converte a variável s1 em um tipo double
17         int w = (int) x; //A variável w do tipo int recebe a parte inteira da variável x do tipo float
18         String s2 = String.valueOf(v); //Converte a variável v do tipo inteiro em String
19         System.out.println(s2 + " " + w); //Exibe os valores das variáveis concatenadas
20     }
21 }
```

```
1  /*
2  Classe exemplo para realizar a conversão de tipos de dados primitivos: ConversaoDeTipos2.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 ConversaoDeTipos2
10 {
11     public static void main(String[] args)
12     {
13         String s1 = "10"; //Declara e inicializa a variável s1
14         int v = Integer.parseInt(s1); //Converte a variável s1 em inteiro
15         float x = Float.parseFloat(s1); //Converte a variável s1 em float
16         double y = Double.parseDouble(s1); //Converte a variável s1 em double
17         int w = (int) x; //A variável w do tipo int recebe a parte inteira da variável x do tipo float
18         int z = (int) y; //A variável z do tipo int recebe a parte inteira da variável y do tipo double
19         String s2 = String.valueOf(v); //Converte a variável v do tipo inteiro em String
20         System.out.println(s2 + " " + " " + w + " " + " " + z); //Exibe os valores das variáveis concatenadas
21     }
22 }
```

Variáveis



Definição de variáveis

- ▶ É um nome definido pelo programador ao qual se associa um valor que pertence a um tipo de dados particular, armazenado em um conjunto de posições da memória do computador;
- ▶ O nome, ou identificador de variável, deve descrever seu propósito e serve como representação simbólica do local da memória onde se encontra seu valor;

Definição de variáveis

- ▶ Todo identificador possui:
 - ▶ Um tipo;
 - ▶ Um nome;
 - ▶ Um valor (seu conteúdo).

```
char sexo = 'f';  
byte idade = 37;
```



Definição de variáveis

- ▶ Palavras reservadas:
 - ▶ O Java possui um conjunto de palavras que indicam os nomes dos tipos primitivos, as diretivas da linguagem, os especificadores e modificadores e outros elementos pertencentes à sintaxe dessa linguagem;
 - ▶ Essas palavras são reservadas e não podem ser utilizadas com outro fim.

Variáveis - Palavras reservadas

abstract	default	goto	package	this
assert	do	if	private	throw
boolean	double	implements	protected	throws
break	else	import	public	transient
byte	enum	instanceof	return	try
case	extends	int	short	true
catch	false	interface	static	var
char	final	long	strictfp	void
class	finally	native	super	volatile
const	float	new	switch	while
continue	for	null	synchronized	yield

Denominação de variáveis



Denominação de variáveis

- ▶ Os nomes de variáveis podem ser compostos de uma ou mais letras, números e também os símbolos _ (sublinhado ou underscore) e \$ (cifrão);
- ▶ Os nomes de variáveis podem começar com letra, caractere de sublinhado (_) ou cifrão (\$);
- ▶ Os nomes de variáveis e identificadores em geral no Java não podem começar com caracteres especiais e nem com números e, apesar de válido, devem-se evitar os caracteres underscore "_" e cifrão "\$".

Denominação de variáveis

- ▶ Conforme a convenção de codificação da linguagem Java, os nomes são iniciados com letras minúsculas;
- ▶ O nome das variáveis devem ser pequenos e de fácil entendimento;
- ▶ Caso o nome seja composto de mais de uma palavra, a primeira inicial deve ser minúscula e as demais iniciais devem ser letras maiúsculas;
- ▶ Caso o nome seja constituído de uma única letra, é sugerida a inicial de seu próprio tipo (i para int ou d para double), acompanhada eventualmente de números para caracterizar variáveis diferentes (como i1, i2 etc.);

Denominação de variáveis

► Ex.:

i, d, total, sinal09, _media, \$total, valorVenda, codigoFornecedor,
valorTaxaJuros, posAbsoluta, _mediaParcial, possuiCPF, totalNumRows,
actionClassName, serialVersionUID.

Declaração de variáveis



Declaração de variáveis

- ▶ Declarar uma variável é associar um tipo, um nome e um valor. Para isso deve ser observada a sintaxe:

```
<tipo> <nome>[, nome2[, nome3[,..... nomeN]]];
```

- ▶ A especificação do tipo e ao menos um nome são obrigatórios;
- ▶ A declaração é terminada por ; (ponto e vírgula).

Declaração de variáveis

- ▶ Variáveis de um mesmo tipo podem ser declaradas separadas ou em conjunto;

//declarações separadas

char opcao1;

char opcao2;

//declaração em conjunto

char opcao3, opcao4;

Declaração de variáveis

- ▶ Java é case sensitive

- ▶ Ex.: `char sexo;`
`char Sexo;`

Declaração e inicialização de variáveis



Declaração e inicialização de variáveis

- ▶ É possível definir um valor inicial para uma variável (o que é chamado inicialização) diretamente em sua declaração;

```
int i, quantidade = 0;  
float angulo = 1.57f, temp;  
double valor = 13.546;  
boolean ok = false;  
char letra = 'c';
```

Declaração e inicialização de variáveis

- ▶ Caso a variável não seja inicializada, a linguagem Java atribuirá valores padrão para cada uma delas;
- ▶ Variáveis dos tipos char, byte, short, int, long, float e double o valor 0 por default;
- ▶ Variáveis do tipo boolean, por default, atribui-se false.

Escopo de variáveis



Declaração e inicialização de variáveis

- ▶ O conjunto dos locais no qual uma declaração de variável tem validade é denominado escopo;
- ▶ Um bloco de comandos é um conjunto de comandos da linguagem delimitados por uma chave inicial { e outra chave final };
- ▶ A partir do ponto em que ocorreu a declaração de uma variável, é dito que a variável está em seu escopo, com validade limitada ao bloco de comandos onde ocorreu essa declaração, mas incluindo escopos mais internos.

Declaração e inicialização de variáveis

```
int i = 33;  
{  
    int j = i * 2;  
}  
int k = j;
```

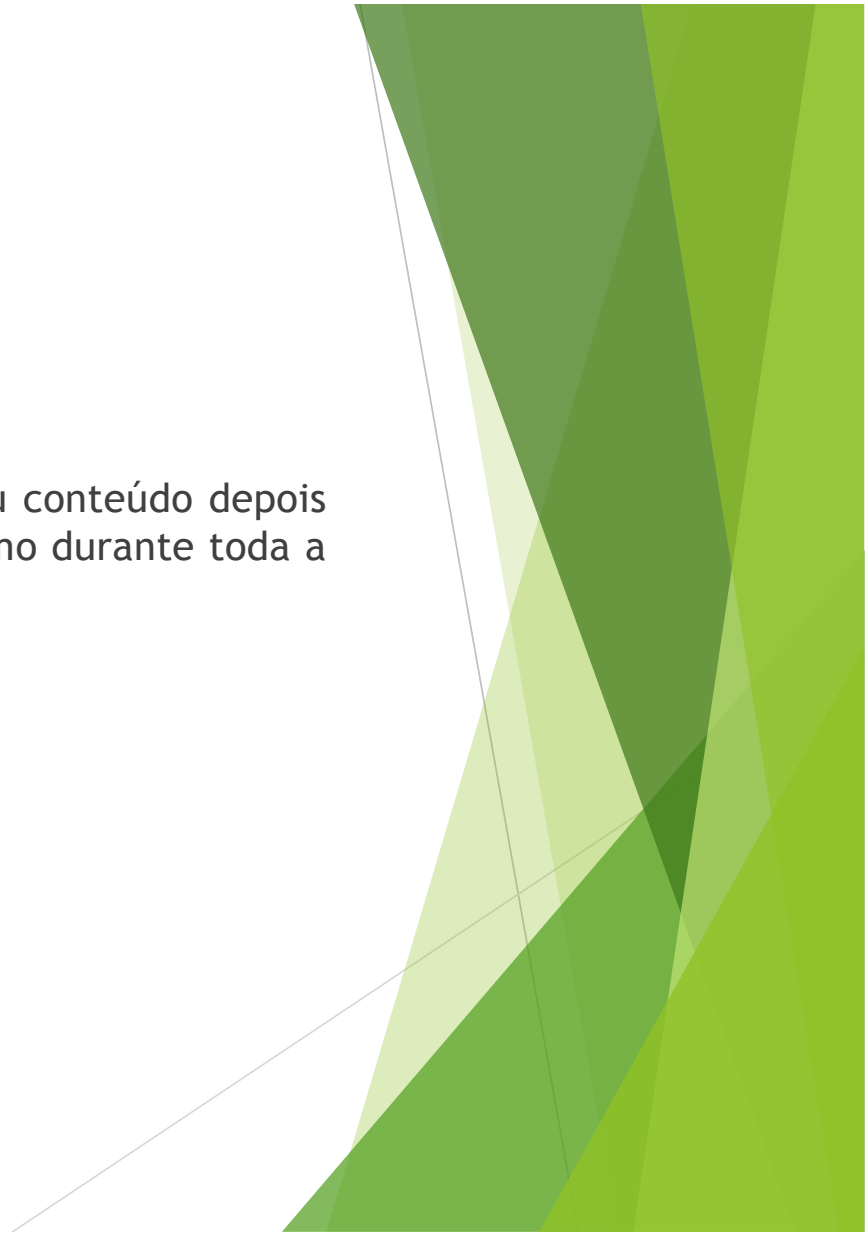
Qual o valor da variável k?

Declaração de constantes



Declaração de constantes

Uma constante é um tipo de variável que não pode alterar seu conteúdo depois de ter sido inicializado, ou seja, o conteúdo permanece o mesmo durante toda a execução do programa.



Declaração de constantes

- ▶ `final double PI = 3.14;`
- ▶ `final int MILISEGUNDOS_POR_SEGUNDO = 1000;`

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.