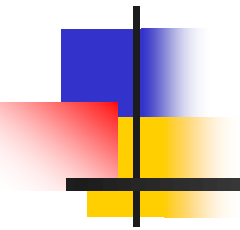


A Linguagem Java

Alberto Costa Neto

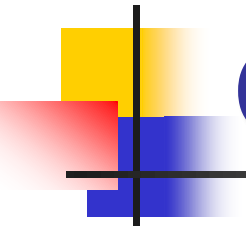
DComp - UFS

Roteiro

- Comentários
- Variáveis
- Tipos Primitivos de Dados
- *Casting*
- Comandos de Entrada e Saída
- Operadores
- Constantes

Comentários

```
/** Classe para impressão de texto */

public class PrimeiroPrograma {

    /* Método principal
       inicia a execução da aplicação Java */

    public static void main (String[] args){

        System.out.println("Seja bem vindo a programação"
                                    + " em Java");
    } //finaliza o método

} //finaliza a classe
```



Comentários

/* Estilo C padrão */

// Estilo C++

/** Especial para geração de Javadoc */

Variáveis

```
import java.util.Scanner;
public class Quadrado {
        public static void main (String[] args){
            System.out.println("Qual o lado? ");
            Scanner dado = new Scanner(System.in);
            float lado = dado.nextFloat();
            System.out.println("Area do quadrado: "
                                + lado * lado);

        }
}
```

Variáveis

- Declaração de variáveis
  - <tipo> <nome-da-variável> [=<valor-inicial>]
  - O tipo sempre deve ser especificado
  - O tipo não pode ser alterado após a declaração.
  - Os identificadores são *case-sensitives*
  - Exemplos:
    - double valor;
    - long x, y = 10, z = 100;
    - boolean ok = false

Variáveis

- Identificadores de variáveis em Java devem seguir as regras:
  - A primeira posição deverá ser com uma letra, "_" ou "$"
  - Não ser uma palavra reservada
  - Não ser igual as literais : true, false ou null
  - Não ser repetido dentro do seu escopo

Variáveis

- Palavras Reservadas

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

Tipos Primitivos de Dados

- **Classificação**
  - Lógicos
  - Numéricos Inteiros
  - Numéricos Reais
  - Caracteres

Tipos de Dados Primitivos

- Lógicos
  - boolean
  - valores: true ou false (padrão)
  - Não são inteiros
  - Exemplo:
    - boolean cpfOk = false;

Tipos de Dados Primitivos

- Numéricos Inteiros

Tipo	Tamanho	Valor Mínimo	Valor Máximo
byte	8 bits	-128	127
short	16 bits	-32768	32767
int	32 bits	-2147483648	2147483647
long	64 bits	-9223372036854775808	9223372036854775807

Exemplos:

```
byte  b = 127;
short s = -32768;
int   i = 2147483647;
long  l = 9223372036854775807L; (valores long terminam com 'L' ou 'l' )
```

Tipos de Dados Primitivos

- Numéricos Reais

Tipo	Tam.	Valor Mínimo	Valor Máximo
float	32 bits	± 1.40239846E-45	± 3.40282347E+48
double	64 bits	± 4.94065645841246544E-324	± 1.79769313486231570E+308

Exemplos:

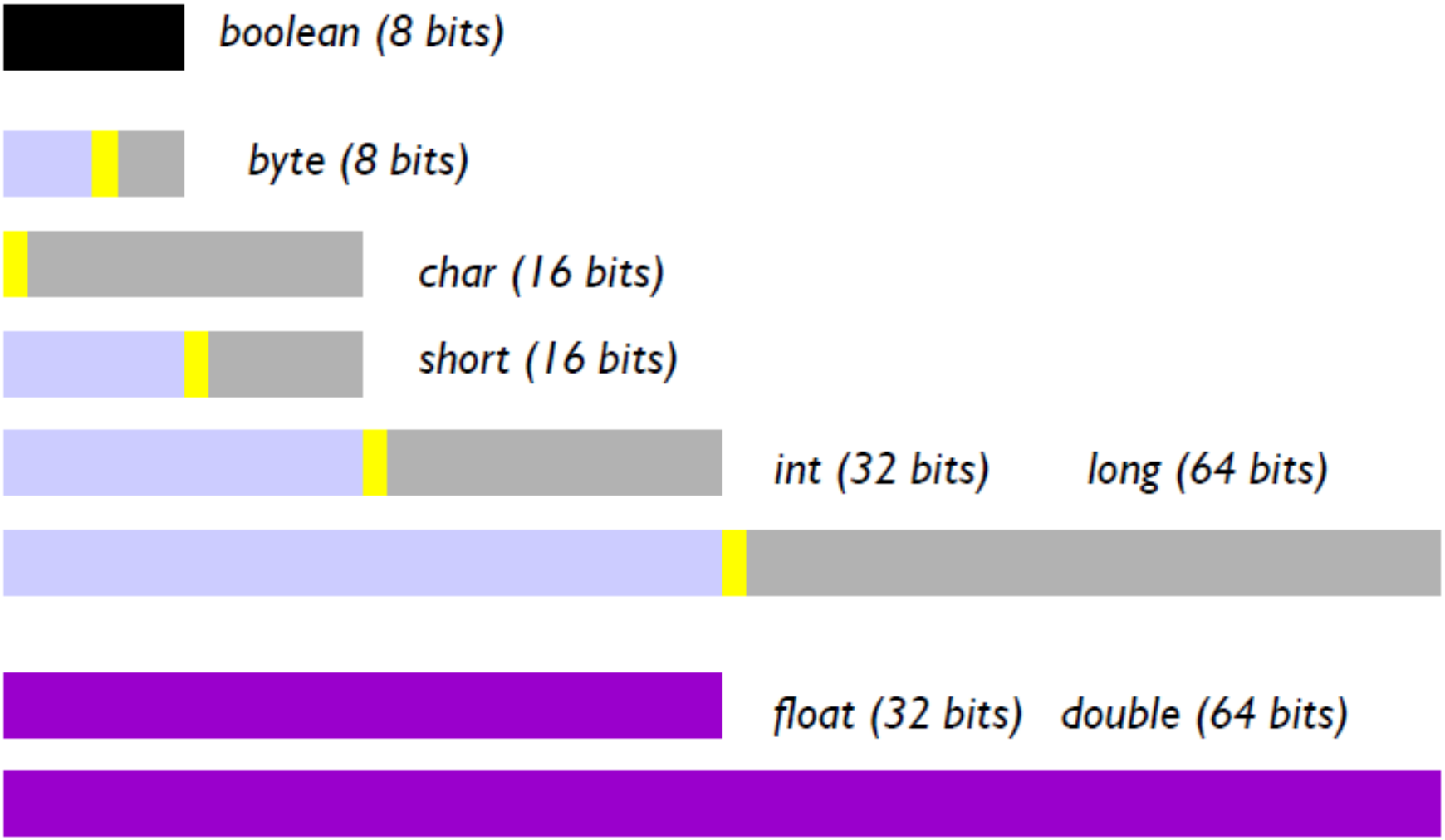
```
float  f1 = 0.0f; (valores float terminam com 'f ' ou 'F')
float  f2 = 3.00e+8F;
double d1 = 0.0;
double d2 = 3.00E+8;
```

Tipos de Dados Primitivos

- Caracteres
  - char
  - Delimitado por apóstrofo
  - Caracteres Unicode de \u0000 a \uFFFF
  - Valor padrão \u0000
  - Exemplo
    - 'A' ou '\u0041'

Tipos de Dados Primitivos

- Tamanho dos Tipos



Casting

- Utilizado para atribuição de valores entre variáveis de tipos diferentes
- Conseguem acomodar o valor ou parte deste valor
- Podem ser implícitos ou explícitos

```
long x = 10000;
int i = (int) x;
```

```
double d3 = 3.14;
int i = (int) d3;
```

Casting

- Castings possíveis
  - Impl => Implícito e automático.

PARA: / DE:	byte	short	char	int	long	float	double
byte	----	*Impl.*	(char)	*Impl.*	*Impl.*	*Impl.*	*Impl.*
short	(byte)	----	(char)	*Impl.*	*Impl.*	*Impl.*	*Impl.*
char	(byte)	(short)	----	*Impl.*	*Impl.*	*Impl.*	*Impl.*
int	(byte)	(short)	(char)	----	*Impl.*	*Impl.*	*Impl.*
long	(byte)	(short)	(char)	(int)	----	*Impl.*	*Impl.*
float	(byte)	(short)	(char)	(int)	(long)	----	*Impl.*
double	(byte)	(short)	(char)	(int)	(long)	(float)	----

Comandos de Entrada e Saída

- Entrada de Dados pelo console

```
Scanner dado = new Scanner(System.in);
int numero = dado.nextInt();
```

```
Scanner dado = new Scanner(System.in);
float numero1 = dado.nextFloat();
float numero2 = dado.nextFloat();
```

```
Scanner dado = new Scanner(System.in);
String nome = dado.next();
```

Comandos de Entrada e Saída

- Saída
  - Imprime o resultado no console e pula uma linha:

```
System.out.println("Sorteio da Mega Sena!");
System.out.println("O resultado é", result);
```

  - Imprime o resultado na console e não pula linha

```
System.out.print("Sorteio da ");
System.out.print("Mega Sena!");
```

Operadores

- Aritméticos
- Atribuições
- Incrementos/Decrementos
- Lógicos
- Relacionais
- Concatenação

Operadores

- Aritméticos

Operador	Sintaxe	Descrição	Resultado
+	X + Y	Soma X e Y	7
-	X - Y	Subtrai Y de X	3
*	X * Y	Multiplica X por Y	10
/	X / Y	Divide X por Y (retorna um inteiro se X e Y forem inteiros)	2
%	X % Y	Resto da divisão de X por Y	1

* Considere o valores de X e Y iguais a 5 e 2, respectivamente

Operadores

- Atribuição

Operador	Sintaxe	Descrição	Resultado
=	X = Y	Atribui a X o valor de Y	X = 2
+=	X += Y	Atribui a X o valor de X + Y	X = 7
-=	X -= Y	Atribui a X o valor de X – Y	X = 3
*=	X *= Y	Atribui a X o valor de X * Y	X = 10
/=	X /= Y	Atribui a X o valor de X / Y	X = 2
%=	X %= Y	Atribui a X o valor de X % Y	X = 1

* Considere o valores de X e Y iguais a 5 e 2, respectivamente

Operadores

- Incremento e Decremento

Operador	Sintaxe	Descrição	Exemplo	Resultado
++ Pré-inc.	++X;	Incrementa a variável antes de retornar o valor	Y = ++X	X = 3 Y = 3
++ Pós-inc.	X++;	Incrementa a variável depois de retornar o valor	Y = X++	X = 3 Y = 2
-- Pré-dec.	--X	Decrementa a variável antes de retornar o valor	Y = --X	X = 1 Y = 1
-- Pós-dec.	X--	Incrementa a variável depois de retornar o valor	Y = X--	X = 1 Y = 2

* Considere o valor inicial de X a 2

Operadores

- Lógicos

Operador	Sintaxe	Descrição
&&	X && Y	AND com curto-circuito (não avalia Y caso X seja false)
&	X & Y	AND sem curto-circuito (sempre avalia X e Y)
||	X || Y	OR com curto-circuito (não avalia Y caso X seja true)
|	X | Y	OR sem curto-circuito (sempre avalia X e Y)
!	! X	NOT

Operadores

- Relacionais

Operador	Sintaxe	Descrição	Exemplo	Resultado
==	X == Y	Igual	2 == 5 3 == 3	false true
!=	X != Y	Diferente	5 != 2 3 != 3	true false
<	X < Y	Menor que	2 < 5 2 < 2	true false
>	X > Y	Maior que	5 > 2 2 > 2	true false
<=	X <= Y	Menor ou igual	2 <= 5 2 <= 2	true true
>=	X >= Y	Maior ou igual	5 >= 2 2 >= 2	true true

Operadores

- Concatenação (Strings)

Operador	Sintaxe	Descrição	Exemplo	Resultado
+	X + Y	Concatena as Strings X e Y, gerando uma nova String	X = "ABC"; Y = "DE"; Z = X + Y;	"ABCDE"

Constantes

```
import java.util.Scanner;
public class Circunferencia {
        public static void main (String[] args){

            final float PI = 3.14f;
            System.out.println("Qual o raio? ");
            Scanner dado = new Scanner(System.in);
            float raio = dado.nextFloat();
            System.out.println("Perimetro da Circunferencia: "
                                    + 2 * PI * raio);
              }
}
```

Constantes

- Definidas a partir:
  - Modificador final
  - Atribuição de valor
- Costumam ser escritas com letras maiúsculas.
- Exemplo:
  - final float PI = 3,14f;

Dever de Sala

- 1) Escreva um programa em Java que leia 3 números inteiros, calcule e imprima a média deles.

- 2) Escreva um programa em Java que leia o valor do raio e imprima a área do círculo

Dever de Sala

- 3) Escreva um programa em Java que leia o valor do salário de um funcionário, calcule e mostre:

  a) o valor do salário,

  b) o valor do aumento

  c) o novo salário

  (Considere que o aumento foi de 25%)

Dever de Sala

- 4) Sabe-se que o Kw de energia custa um quinto do salário mínimo. Faça um programa em Java que receba o valor do salário mínimo e a quantidade de Kw consumidos, calcule e mostre:
  a) o valor de cada Kw
  b) o valor a ser pago por essa residência
  c) o valor a ser pago com desconto de 15%

Referências

- Slides Estrutura Sequencial. Prof. Marcos Dósea. UFS. 2010.
- Slides "Linguagem Java", Profª. Débora. UFS. 2010
- Java How to Program
  - Capítulo – 2