

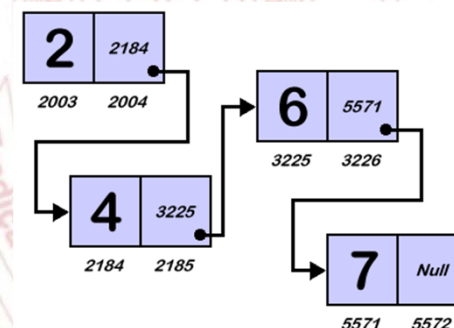
Universidade Presbiteriana Mackenzie



Faculdade de
Computação e Informática



Estrutura de Dados I



Aula 2: Revisão Java Básico



Prof. Ms. Daniel Arndt Alves

Faculdade de Computação e Informática



Tópicos da Aula

- Organização de um programa Java
- Estrutura básica de uma classe
- Exemplo de uma classe
- Palavras reservadas
- Tipos de dados da linguagem Java
 - Tipos primitivos



Tópicos da Aula

- Operadores
 - Operadores Básicos
 - Operadores Aritméticos
 - Operadores Relacionais
 - Operadores Lógicos e Binários
 - Precedência de operadores
- Comandos de controle
- Padrões de comentário



Organização de um programa Java

```
class Classe1 { ... } // Classes "locais"
class Classe2 { ... } // Classes "locais"
...
// O nome do arquivo será ClassePrincipal.java
public class ClassePrincipal {
    ...
    // Programa principal
    public static void main(String args[]) {
        ...
    }
}
```


Estrutura básica de uma classe

```
class <nome da classe> {  
    // Variáveis de classe  
    static <tipo0> atributo0;  
    ...  
    // Variáveis de instância: LP-I  
    <tipo1> atributo1;  
    <tipo2> atributo2, atributo3;  
    ...  
    // Métodos construtores: LP-I  
    // Métodos de classe  
    static void método1 (<parâmetros>) {...}  
    static <tipo> método2 (<parâmetros>) {...}  
    // Métodos de instância: LP-I  
    void método3 (<parâmetros>) {...}  
    <tipo> método4 (<parâmetros>) {...}  
}
```

Exemplo de uma classe

```
class numero_racional {  
  
    int numerador,    // Numerador do número  
    int denominador; // Denominador do número  
  
    numero_racional(int num, int denom) {  
        // Construtor  
        numerador = num;  
        denominador = denom;  
    }  
  
    float racional_para_float() {  
        // Converte números racionais para ponto flutuante  
        float real;  
        real = (float) numerador / (float) denominador;  
        return (real);  
    }  
}
```



Exemplo de um Programa – Aula00

```
class Data { // Classe interna ao programa
    static boolean ehBissexto(int ano) {
        // Multiplo de 400 -> nao eh bissexto
        if (ano % 400 == 0) {
            return true;
        }
        // Sabemos que nao eh multiplo de 400
        // Mas se for multiplo de 100 -> nao eh bissexto
        if (ano % 100 == 0) {
            return false;
        }
        // Nao eh multiplo de 400 nem multiplo de 100
        // Basta agora verificar se eh multiplo de 4
        if (ano % 4 == 0) {
            return true;
        }
        return false;
    }
}
```



Exemplo de um Programa – Aula00

```
public class Aula00 {  
    public static void main(String[] args) {  
        if (Data.ehBissexto(2011))  
            System.out.println("\n2011 é bissexto);  
        else  
            System.out.println("\n2011 não é bissexto);  
    }  
}  
// Todo o código fonte ficará no arquivo Aula00.java
```


Palavras reservadas

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

*Palavras reservadas, mas não implementadas



Tipos de dados da linguagem Java

- Java possui duas categorias de tipos de dados:
 - Tipo Primitivo
 - Tipo de Instância ou Referência
- Tipo Primitivo
 - São tipos de dados predefinidos pela linguagem e correspondem a dados mais simples ou escalares
- Tipo de Instância ou Referência
 - Uma instância é um objeto do tipo definido pela classe

Tipos primitivos

Tipo	Tamanho	Intervalo
boolean	1	true, false
char	16 (s/ sinal)	0 ~ 65.535 (inclui UNICODE)
byte	8	-128 ~ 127
short	16	-32.768 ~ 32.767
int	32	-2.147.483.648 ~ 2.147.483.647
long	64	-922.337.203.685.475.808 ~ 922.337.203.685.475.807
float	32	-3.4E-38 ~ 3.4E38
double	64	-1.7E-308 ~ 1.7E308

Operadores

- Em Java, trabalha-se com quatro conjuntos de operadores:
 - Básicos
 - Aritméticos
 - Relacionais
 - Lógicos e Binários

Operadores Básicos

Operador	Função
.	Referência a um método, função ou atributo de um objeto
,	Separador de identificadores
;	Separador de declarações e comandos
[]	Declarador de vetores, matrizes e delimitador de índices
{ }	Separador de blocos e escopos locais
()	Precedência de operadores, listas de parâmetros

Operadores Aritméticos

Operador	Função	Exemplo
+	adição	$x = x + 3$
-	subtração	$x = x - 3$
/	divisão	$x = y / 4$
%	resto da divisão	$x = y \% 5$
++	incremento de 1	$x++;$ $++x;$
--	decremento de 1	$x--;$ $--x;$
+=	atribuição aditiva	$x += 3$
-=	atribuição subtrativa	$x -= 3$
*=	atribuição multiplicativa	$x *= 3$
/=	atribuição divisiva	$x /= 3$
%=	atribuição de resto	$x \% = 3$

Operadores Relacionais

Operador	Função	Exemplo
==	igual	x == y
!=	diferente	x != y
>	maior	x > y
<	menor	x < y
>=	maior ou igual	x >= y
<=	menor ou igual	x <= y

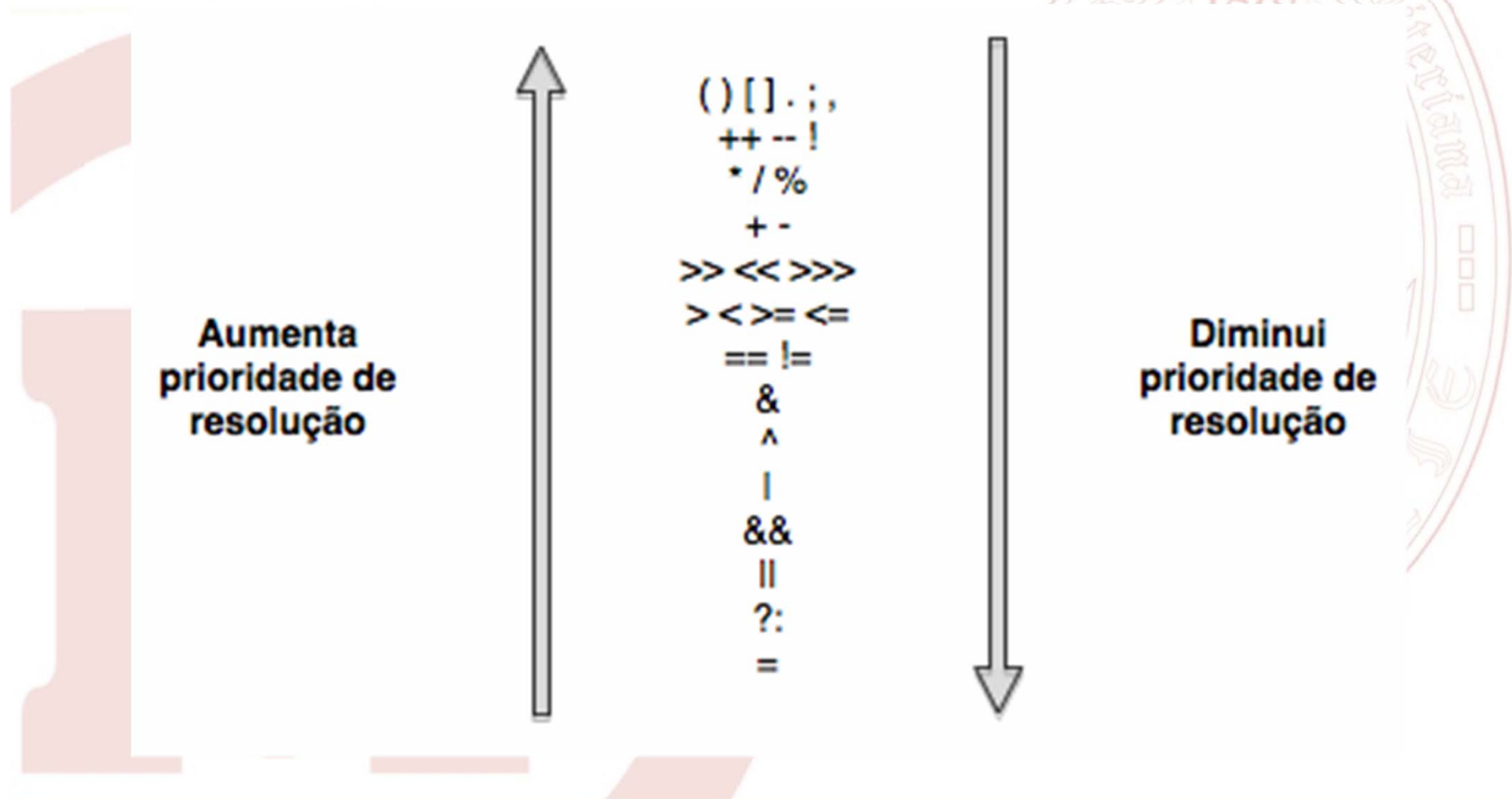
Operadores Lógicos e Binários

Operador	Função	Exemplo
~	negação bit a bit	~ x
>>	shift right	x >> 3
>>>	shift right sem sinal	x >>> 3
<<	shift left	x << 3
>>=	atribuição c/ shift right	x >>= 3
<<=	atribuição c/ shift left	x <<= 3
>>>=	atribuição c/ shift right s/ sinal	x >>>= 3
&	AND lógico bit a bit	x & y
	OR lógico bit a bit	x y
^	XOR lógico bit a bit	x ^ y
&=	atribuição com AND	x &= y
=	atribuição com OR	x = y
^=	atribuição com XOR	x ^= y

Operadores Lógicos e Binários

Operador	Função	Exemplo
? :	if-then-else lógico	<code>x = ((y > 3) ? 5 : 6)</code>
	OR dinâmico	<code>(x > y) (z >= 6)</code>
&&	AND dinâmico	<code>(x > y) && (z < 4)</code>
!	NOT dinâmico	<code>!(x > y)</code>

Precedência de operadores



Comandos de controle

- Condicional

```
if ( expressão booleana )  
    bloco 1;  
else  
    bloco 2;
```

Comandos de controle

- Exemplo de condicional

```
if ((x > 5) && (y != 70)) {  
    x = x + 5;  
    y = z + 48;  
}  
else if (x <= 5)  
    x = 48;  
else  
    x = 20;
```

Comandos de controle

- Seleção

```
switch(expressão) {  
    case valor1: bloco1;  
        break;  
    ...  
    case valorN: blocoN;  
        break;  
    default: bloco N+1;  
}
```



Comandos de controle

- Exemplo de Seleção

```
valorDaCompra = 1000.;
cliente = k;
switch(multiplicador){
    case clienteOuro:
        desconto = 0.30;
        break;
    case clientePrata:
        desconto = 0.10;
        break;
    default: desconto = 0.;
}
System.out.println(valorDaCompra * (1. - desconto));
```



Comandos de controle

- Repetição com `while`
 - A ser usado quando o número de repetições for indeterminado

```
while(expressão booleana){  
    bloco;  
}
```


Comandos de controle

- Exemplo de repetição com `while`

```
// Ler um inteiro n
k = 3;
q = 0;
while(n > k - 1){
    n = n / k;
    q++;
}
System.out.println(q);
// O que q representa?
```



Comandos de controle

- Repetição com do-while
 - A ser usado quando o número de repetições for indeterminado

```
do {  
    Bloco;  
} while(expressão booleana);
```

Comandos de controle

- Exemplo de repetição com do-while

```
int i =0;  
do {  
    System.out.println("Valor de i: " + i);  
    i++;  
} while(i < 5);
```

Comandos de controle

- Repetição com `for`
 - A ser usado quando o número de repetições for determinado

```
for (inicialização; limite; iteração){  
    Bloco;  
}
```

Comandos de controle

- Exemplo de repetição com `for`

```
// Primeiros 20 ímpares
for (int i = 1; i <= 20; i++){
    System.out.println("\n" + i);
    i += 2;
}
```


Padrões de comentário

```
/* Um comentario estilo C*/
```

```
/* Outro comentario estilo C  
   prosseguindo nesta linha
```

```
*/
```

```
// Um comentario estilo C++
```

```
// Mais outro comentario estilo C++
```

- Usar marcadores especiais e aplicar o programa javadoc para gerar a documentação no formato HTML;

A classe Scanner

```
import java.util.Scanner;
...
public class Aula00 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in); // Ler do teclado
        System.out.println("\nDigite um inteiro:");
        int k = sc.nextInt();
        System.out.println("\nDigite um double:");
        double x = sc.nextDouble();
        System.out.println(
            "\nForam lidos: " + k + " e " + x
        )
    }
}
```

Obrigado



Daniel Arndt Alves

daniel.alves@mackenzie.br

