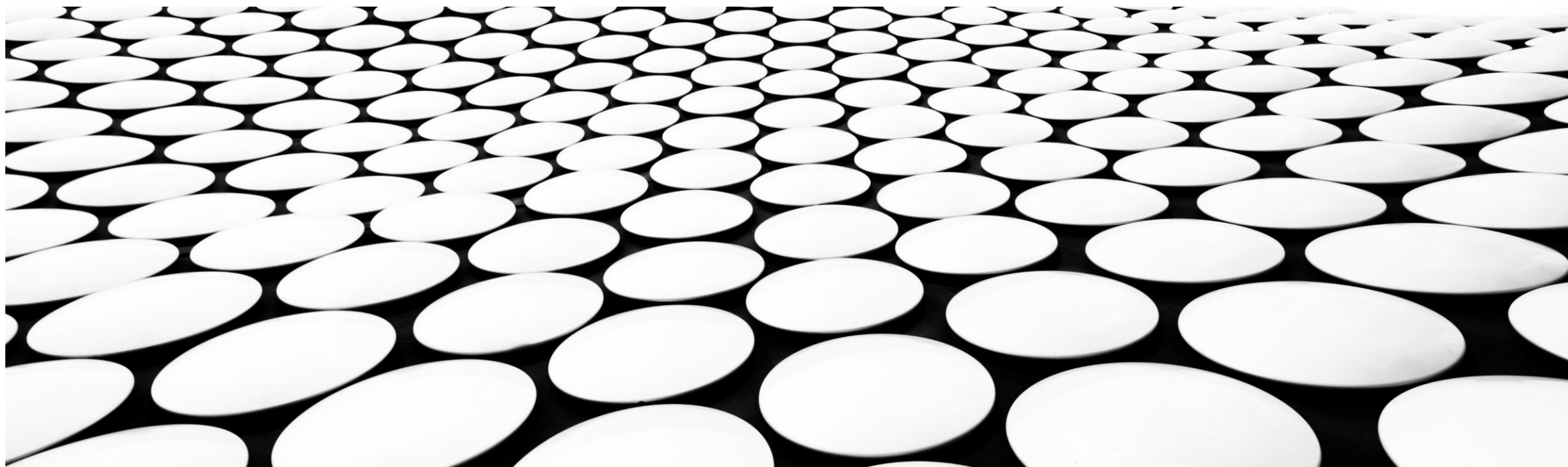

PROGRAMAÇÃO ORIENTADA A OBJETOS: INTRODUÇÃO A APLICATIVOS EM JAVA

PROF. FELIPE MARQUES (COM CONTRIBUIÇÕES DO PROF. RAFAEL BURLAMAQUI AMARAL)



LINGUAGEM JAVA

Primeiro Programa em JAVA

```
public class Principal {  
    public static void main(String[] args) {  
        System.out.println("Alo Mundo!");  
    }  
}
```

LINGUAGEM JAVA

Palavras Reservadas

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

OPERADORES

- Atribuição: `=, +=, -=, *=, /=`
- Aritméticos: `+, -, *, /, %`
- Incremento e decremento: `i++, i--, ++j, --j`
- Relacionais: `<, >, <=, >=, ==, !=`
- Concatenação: `+`
- Lógicos: `&&, ||, !`

PONTO E VÍRGULA E CHAVES

- Ponto e vírgula:
 - Todos os comandos em Java terminam obrigatoriamente com o ponto e vírgula (;).
- Chaves: { }
 - São utilizadas para definir um bloco de comandos.

COMENTÁRIOS E CARACTERES ESPECIAIS,

- Comentários

- Linha: //
- Bloco de comando: /* ... */

- Caracteres especiais

- \n: quebra de linha
- \t: tabulação horizontal

OPERADORES

Símbolo	Operação	Exemplo	Símbolo	Operação	Exemplo
			==	Igual a (atenção ao usar com referências)	a == b
=	Atribuição simples	a = b;	!=	Diferente de (atenção ao usar com referências)	a != b
+=	Atribuição composta, um atalho para operações como a = a + b	a += b;	>	Maior que (apenas para valores numéricos)	a > b
--=			>=	Maior ou igual a (apenas para valores numéricos)	a >= b
*=			<	Menor que (apenas para valores numéricos)	a < b
/=			<=	Menor ou igual a (apenas para valores numéricos)	a <= b
+	Adição	a + b			
-	Subtração	a - b			
*	Multiplicação	a * b			
/	Divisão	a / b			
%	Resto da divisão inteira	a % b			
+	Expressão numérica positiva	+a			
-	Negação de expressão numérica	-a			
++	Incremento	++a OU a++			
--	Decremento	--a OU a--			
+	Concatenação de String	"Texto" + a a + "Texto"			
&&	E lógico (and)	a && b			
	OU lógico (or)	a b			
!	Negação (not)	!a			

TIPOS DE DADOS BÁSICOS DE JAVA

Tipo	Faixa de Valores
boolean	true ou false
char	0 a 65535
byte	-128 a 127
short	-32.768 a 32.767
int	$\pm 2.147.483.647$
long	$\pm 9.223.372.036.854.775.807$
float	$\pm 3.40282347E+38$
double	$\pm 1.7976931348623E+308$
String	Tamanho limitado a memória disponível

CONTROLE DE FLUXO – DESVIO CONDICIONAL

```
if ( expr ){  
    // É possível também aninhar comandos  
}  
else{  
    // bloco de comandos  
}
```

Exemplo:

```
class Teste{  
    public static void main (String[] args){  
        int a = 5, b = 3, m;  
        if ((a > 0) && (b > 0))  
            m = (a + b) / 2;  
        else{  
            System.out.println("numero negativo");  
            m = 0;  
        }  
    }  
}
```

CONTROLE DE FLUXO – DESVIO CONDICIONAL

```
switch (expr) {  
    case const1:  
        //comando1;  
  
        break;  
  
    case const2:  
        //comando2;  
  
        break;  
  
    ...  
  
    default:  
        //comando3;  
  
}
```

Exemplo:

```
class Teste{  
    public static void main(String[] args){  
        int a = 5;  
        switch (a){  
            case 5:  
                System.out.println(5);  
                break;  
            case 0:  
                System.out.println(0);  
                break;  
            default:  
                System.out.println("erro");  
        }  
    }  
}
```

CONTROLE DE FLUXO - LAÇOS

```
while (expr) {  
    // bloco de comandos  
}
```

Exemplo:

```
class Teste{  
    public static void main(String[] args) {  
        int i = 0;  
        while (i<10){  
            System.out.println(i);  
            i++;  
        }  
    }  
}
```

CONTROLE DE FLUXO - LAÇOS

```
do{  
    // bloco de comandos  
}while (expr);
```

Exemplo

```
class Teste{  
    public static void main(String[] args){  
        int i = 0;  
        do{  
            System.out.println(i);  
            i++;  
        }while (i<10);  
    }  
}
```

CONTROLE DE FLUXO - LAÇOS

```
for ( <inicialização> ; < expressão > ; <operação> ) {  
    //bloco de comandos  
}
```

Exemplo:

```
class Teste{  
    public static void main(String[] args){  
        int i = 0;  
        for(i = 0;i < 10; i++){  
            System.out.println(i);  
        }  
    }  
}
```

EXERCÍCIOS

1. Faça um programa que imprima as tabuadas do 1 até 10 (as 10 primeiras linhas de cada tabuada).
2. Calcule o IMC.

Condição	IMC em Mulheres
Magro	peso < 20
Normal	20 <= peso < 25
Acima do peso	25 <= peso < 29
Obeso	29 <= peso < 30
Muito obeso	peso >= 30

3. Faça um programa que atribua valores para os coeficientes (a, b, c) de uma equação do segundo grau, calcule e exiba o valor de seu discriminante, $\Delta = b^2 - 4ac$. Obs: o usuário não irá digitar nada.

EXERCÍCIOS

Nesse primeiro momento não solicite nada ao usuário.
Crie variáveis e atribua valores.

3. Faça um programa para calcular a média aritmética de duas notas. Também calcule e apresente a nota ponderada, considerando peso 2 e 3, respectivamente.
4. Escreva um programa que dado três números inteiros, apresente o menor deles e todos que são positivos.
5. Faça um programa que verifique se um número é primo.
6. Faça um programa que calcule o fatorial.

BIBLIOGRAFIA

- Apostila Caelum
- Santos, Rafael. Introdução à programação orientada a objetos usando Java. Editora Campus, 2003.
- H. M. Deitel e P. J. Deitel, Java: como programar, Pearson Prentice-Hall.