

INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
PIAUÍ  
Campus Picos

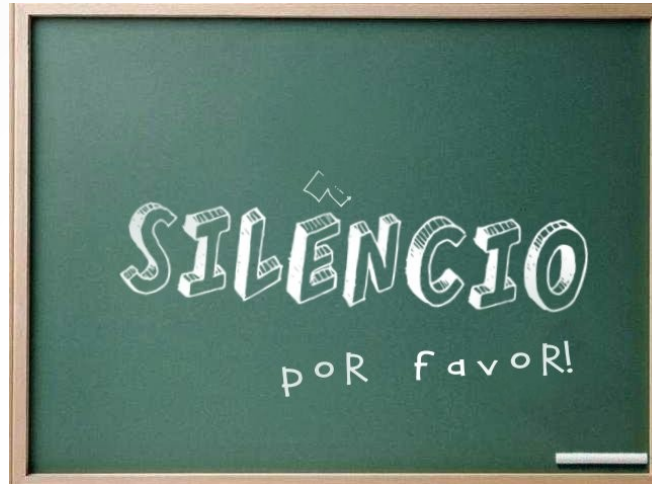
# *Apresentação*

## *Programação Orientada a Objetos* *Estruturas de Controle*

***TEC INFO M3***

Prof. João Paulo  
[joao.nascimento@ifpi.edu.br](mailto:joao.nascimento@ifpi.edu.br)

# *Por favor...*



- Manter o telefone celular sempre silencioso ou desligado em sala de aula;
- Se for receber uma ligação importante, atender fora da sala de aula;
- Fazer silêncio em sala de aula!

# Conteúdo Programático

- Comandos de decisão
  - If
  - If-else
  - switch
- Comandos de repetição
  - While
  - Do-while
  - For
- Bibliografia

# Operadores Aritméticos

Operação Java	Operador	Expressão algébrica	Expressão Java
Adição	+	$f + 7$	<code>f + 7</code>
Subtração	-	$p - c$	<code>p - c</code>
Multiplicação	*	$bm$	<code>b * m</code>
Divisão	/	$x/y$ ou $\frac{x}{y}$ ou $x \div y$	<code>x / y</code>
Resto	%	$r \bmod s$	<code>r % s</code>

Operador	Operação	Ordem de avaliação (precedência)
*	Multiplicação	Avaliado primeiro. Se houver vários operadores desse tipo, eles são avaliados da esquerda para a direita.
/	Divisão	
%	Resto	
+	Adição	Avaliado em seguida. Se houver vários operadores desse tipo, eles são avaliados da esquerda para a direita.
-	Subtração	
=	Atribuição	Avaliado por último.

# Atribuições Aritméticas

Operador de atribuição	Expressão de exemplo	Explicação	Atribuições
<i>Suponha: int c = 3, d = 5, e = 4, f = 6, g = 12;</i>			
+=	c += 7	c = c + 7	10 a c
-=	d -= 4	d = d - 4	1 a d
*=	e *= 5	e = e * 5	20 a e
/=	f /= 3	f = f / 3	2 a f
%=	g %= 9	g = g % 9	3 a g

Operador	Chamado	Expressão de exemplo	Explicação
++	pré-incremento	++a	Incrementa a por 1 e então utiliza o novo valor de a na expressão em que a reside.
++	pós-incremento	a++	Utiliza o valor atual de a na expressão em que a reside, então incrementa a por 1.
--	pré-decremento	--b	Decrementa b por 1 e então utiliza o novo valor de b na expressão em que b reside.
--	pós-decremento	b--	Utiliza o valor atual de b na expressão em que b reside, então decrementa b por 1.



# Operadores Relacionais

Padrão algébrico operador de igualdade ou relacional	Operador de igualdade ou relacional em Java ou operador relacional	Exemplo de condição em Java	Significado da condição em Java
<i>Operadores de igualdade</i>			
=	==	x == y	x é igual a y
≠	!=	x != y	x não é igual a y
<i>Operadores relacionais</i>			
>	>	x > y	x é maior que y
<	<	x < y	x é menor que y
≥	>=	x >= y	x é maior que ou igual a y
≤	<=	x <= y	x é menor que ou igual a y

# Operadores Lógicos

expressão1	expressão2	expressão1 && expressão2
false	false	false
false	true	false
true	false	false
true	true	true

expressão1	expressão2	expressão1    expressão2
false	false	false
false	true	true
true	false	true
true	true	true

expressão1	expressão2	expressão1 ^ expressão2
false	false	false
false	true	true
true	false	true
true	true	false

expressão	!expressão
false	true
true	false

# Operações Precedência

Operadores	Associatividade	Tipo
++ --	da direita para a esquerda	unário pós-fixado
++ -- + - ! (tipo)	da direita para a esquerda	unário pré-fixado
* / %	da esquerda para a direita	multiplicativo
+ -	da esquerda para a direita	aditivo
< <= > >=	da esquerda para a direita	relacional
== !=	da esquerda para a direita	igualdade
&	da esquerda para a direita	E lógico booleano
^	da esquerda para a direita	OU lógico booleano exclusivo
	da esquerda para a direita	OU lógico booleano inclusivo
&&	da esquerda para a direita	E condicional
	da esquerda para a direita	OU condicional
?:	da direita para a esquerda	ternário condicional
= += -= *= /= %=	da direita para a esquerda	atribuição



# Decisão simples

- IF

```
if ( studentGrade >= 60 )  
    System.out.println( "Aprovado" );
```

# *Decisão composta*

- IF-ELSE

```
if ( grade >= 60 )  
    System.out.println( "Aprovado" );  
else  
    System.out.println( "Reprovado" );
```

# *Endeamento IF-ELSE*

```
if ( studentGrade >= 90 )  
    System.out.println( 'A' );  
else if ( studentGrade >= 80 )  
    System.out.println( 'B' );  
else if ( studentGrade >= 70 )  
    System.out.println( 'C' );  
else if ( studentGrade >= 60 )  
    System.out.println( 'D' );  
else  
    System.out.println( 'F' );
```

# Switch

```
switch(expression) {  
    case x:  
        // code block  
        break;  
    case y:  
        // code block  
        break;  
    default:  
        // code block  
}
```

```
int day = 4;  
switch (day) {  
    case 6:  
        System.out.println("Today is Saturday");  
        break;  
    case 7:  
        System.out.println("Today is Sunday");  
        break;  
    default:  
        System.out.println("Looking forward to the Weekend");  
}
```



# Condicional Ternário

```
if (a > 0)
    b = 1;
else
    b = 2;
```



```
b = (a > 0) ? 1 : 2;
```

```
System.out.println(1 == 1 ? "Verdadeiro" : "False");
```

Com if else:

```
if(1 == 1)
    System.out.println("Verdadeiro")
else
    System.out.println("Falso")
```

# While

```
while (condição) {
```

```
    ...  
}
```

```
public class Aula {  
  
    Run | Debug  
    public static void main(String args[]) {  
  
        //Imprimir 1 a 10...  
        int i = 1;  
        while( i <= 10 ) {  
  
            System.out.println(i);  
            i++;  
        }  
    }  
}
```

# Do-while

do {

■ ■ ■

} while(condição);

```
public class Aula {  
  
    Run | Debug  
    public static void main(String args[]) {  
  
        //Imprimir 1 a 10...  
        int i = 1;  
        do {  
            System.out.println(i);  
            i++;  
        }while( i<=10 );  
    }  
  
}
```

# For

```
for( iterador; condição; incremento ) {
```

```
    ...  
}
```

```
public class Aula {  
  
    Run | Debug  
    public static void main(String args[]) {  
  
        //Imprimir 1 a 10...  
        for( int i = 1; i <= 10; i++ ) {  
  
            System.out.println(i);  
  
        }  
  
    }  
  
}
```



# For-each

```
for( iterador: array/coleção ) {
```

```
    ...
```

```
}
```

```
public class Aula {  
  
    Run | Debug  
    public static void main(String args[]) {  
  
        //Imprimir 1 a 10...  
        int numeros[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
  
        for( int i: numeros ) {  
  
            System.out.println( i );  
  
        }  
  
    }  
  
}
```

# Break e Continue

break;

continue;

```
import java.util.Scanner;

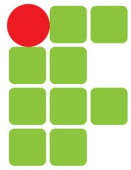
public class Aula {

    Run | Debug
    public static void main(String args[]) {

        //Deve-se digitar um valor positivo...
        while(true) {

            System.out.println("Favor digitar um valor positivo (x > 0): ");
            Scanner sc = new Scanner(System.in);
            int x = sc.nextInt();
            if( x < 0 )
                continue; //retorna pro início do laço while

            System.out.println("Valor positivo! Bye!");
            break; //sai do laço
        }
    }
}
```



# *Bibliografia*

- DEITEL, Harvey; DEITEL, Paul. Java Como Programar. 8a Ed. Bookman, 2010.
- SANTOS, Rafael. Introdução à programação orientada a objetos usando Java. Rio de Janeiro: Campus, 2003. 319, [4] p. (Editora Campus ; Sociedade Brasileira de Computação). ISBN 85-352-1206-X (broch.).
- SINTES, Anthony. Aprenda Programação Orientada a Objetos. São Paulo: Pearson, 2002. 693p. (Editora Pearson ; Sociedade brasileira de computação). ISBN 85-34-1461-0.
- SILVA FILHO, Antonio Mendes da. Introdução à programação orientada a objetos com C++. Rio de Janeiro: Elsevier, 2010. 283 p. ISBN 978-85-352-3702-3
- VILARIM, Gilvan de Oliveira. Programação orientada a objetos: um curso básico. Curitiba: Editora do Livro Técnico, 2015. 168p. ISBN 9788563687920.
- CARDOSO, Caíque. Orientação a objetos na prática: aprendendo orientação a objetos com Java . Rio de Janeiro: Ciência Moderna, 2006. 175 p. ISBN 85-7393-538-3.