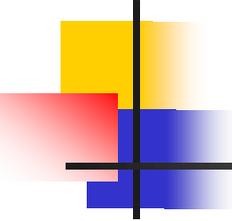


Sintaxe Básica de Java - Parte 2

Universidade Católica de Pernambuco
Ciência da Computação

Prof. Márcio Bueno
pooite@marciobueno.com

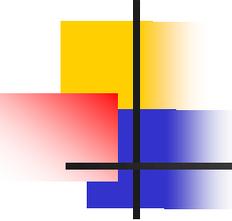
Fonte: Material da Prof^a Karina Oliveira



Instruções Condicionais

- A instrução `if`
- Sintaxe:

```
if (<expressão booleana>) {  
    <instruções do if>;  
} else if (<expressão booleana>) {  
    < instruções do else-if>;  
} else {  
    < instruções do else>;  
}
```



Instruções Condicionais

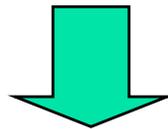
- A instrução if
 - Exemplo: Escrever um programa que receba 2 números, determine e imprima na tela o maior deles.

```
import javax.swing.JOptionPane;
public class ExemploIf {
    public static void main(String[] args) {
        int a = Integer.parseInt(JOptionPane.showInputDialog("A:"));
        int b = Integer.parseInt(JOptionPane.showInputDialog("B:"));
        if (a > b) {
            System.out.println("Maior = " + a);
        } else {
            System.out.println("Maior = " + b);
        }
    }
}
```

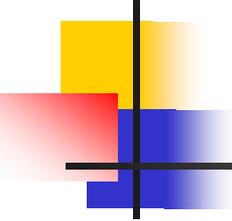
Instruções Condicionais

- O Operador Ternário ?:
 - Forma compacta de expressar uma instrução if - else
 - Sintaxe:

```
<expressão> ? <resultadoSeVerdadeiro> : <resultadoSeFalso>;
```



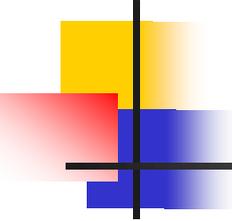
```
if (<expressão>
    <resultadoSeVerdadeiro>;
else
    <resultadoSeFalso>;
```



Instruções Condicionais

- Operador ?:
 - Exemplo: Escrever um programa que receba 2 números como parâmetros, determine e imprima na tela o maior deles.

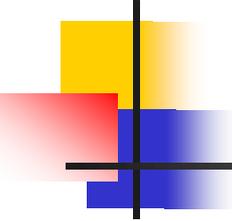
```
public class ExemploIfCompacto {  
    public static void main(String[ ] args) {  
        int a = Integer.parseInt(args[0]);  
        int b = Integer.parseInt(args[1]);  
        int maior = (a > b) ? a : b;  
        System.out.println("Maior = " + maior);  
    }  
}
```



Instruções Condicionais

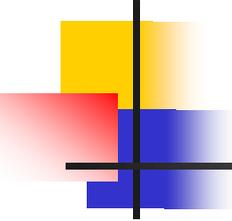
- A instrução switch
 - Forma de substituir a instrução if para se executar várias comparações em um conjunto de valores discretos.
 - Sintaxe:

```
switch (<variavelDeTeste>) {  
    case <valor de comparação 1>:  
        <instruções>;  
        break;  
    case <valor de comparação n>:  
        <instruções>;  
        break;  
    default:  
        <instruções>;  
        break;  
}
```



Instruções Condicionais

- A instrução switch
 - OBS 1: A variável de teste pode ser:
 - int (tipos inteiros)
 - char
 - A instrução break é opcional!
 - A cláusula default é opcional!

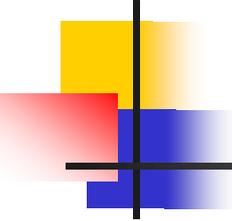


Instruções Condicionais

- Instrução switch

- Exemplo: Escrever um programa que receba 1 número e imprima na tela o dia correspondente da semana (1 = Domingo, 2 = Segunda, 3 = Terça, ...).

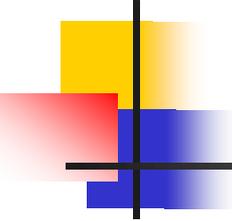
```
public class ExemploSwitch {  
    public static void main(String[] args) {  
        int dia = Integer.parseInt(JOptionPane.showInputDialog("A:"));  
        switch (dia) {  
            case 1:  
                System.out.println("Dia = Domingo");  
                break;  
            case 2:  
                System.out.println("Dia = Segunda");  
                break;  
        }  
    }  
}
```



Instruções Condicionais

- Instrução switch
 - Exemplo (Continuação):

```
    case 3:  
        System.out.println("Dia = Terça");  
        break;  
  
    ...  
    default:  
        System.out.println("Dia inválido!");  
        break;  
    }  
}  
}
```

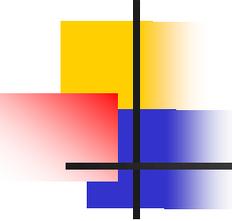


Instruções de Repetição

- A instrução for
 - Sintaxe:

```
for (<inicialização>; <expressão>; <operação>) {  
    <instruções do for>;  
}
```

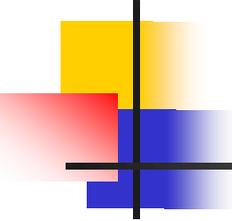
- Inicialização - expressão de atribuição executada uma única vez no início da primeira iteração do laço.
- Expressão - Expressão avaliada a cada iteração do laço. Se **verdadeira continua**, se **falsa termina**.
- Operação - define como a variável de controle será alterada. É executada ao final de cada iteração do laço.



Instruções de Repetição

- A instrução for
 - Exemplo: Como imprimir os 1000 primeiros números inteiros a partir de 1.

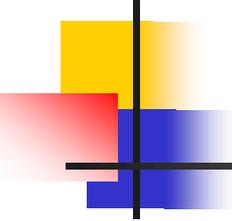
```
public class ExemploFor {  
    public static void main(String[ ] args) {  
        for (int num = 1; num <= 1000; num++) {  
            System.out.println(" " + num);  
        }  
    }  
}
```



Instruções de Repetição

- A instrução `while`
 - O laço se repete enquanto a expressão for verdadeira.
 - Sintaxe:

```
while (<expressão>) {  
    <instruções do while>;  
}
```



Instruções de Repetição

- A instrução `while`
 - Exemplo: Como imprimir os 1000 primeiros números a partir de 1.

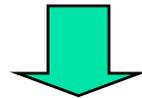
```
public class ExemploWhile {  
    public static void main(String[] args) {  
        int num = 1;  
        while (num <= 1000) {  
            System.out.println(" " + num);  
            num++;  
        }  
    }  
}
```

Instruções de Repetição

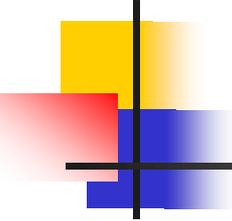
- A instrução do - while
 - Ao contrário dos comandos for e while que verificam a expressão no início do laço, do - while sempre verifica a expressão no final, garantindo a execução do bloco de instruções pelo menos uma vez.

- Sintaxe:

```
do {  
    <instruções do do-while>;  
} while (<expressão>;
```



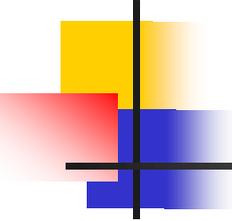
```
<instruções do while>;  
while (<expressão>) {  
    <instruções do while>;  
}
```



Instruções de Repetição

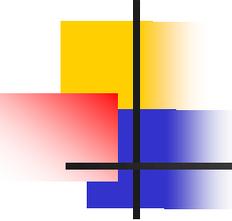
- A instrução do - while
 - Exemplo: Como imprimir os 1000 primeiros números a partir de 1.

```
public class ExemploDoWhile {  
    public static void main(String[ ] args) {  
        int num = 1;  
        do {  
            System.out.println(" " + num);  
            num++;  
        } while (num <= 1000);  
    }  
}
```



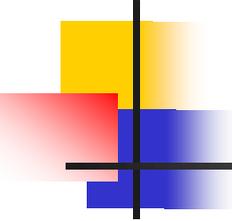
Exercícios

- Instrução if
 - Exercício 1: Escrever um programa que receba 2 números como parâmetros, realize e imprima a divisão do primeiro pelo segundo. OBS: Deve-se evitar divisão por zero.
 - Exercício 2: Escreva um programa que leia três números inteiros do teclado, e os imprima na tela em ordem ascendente.



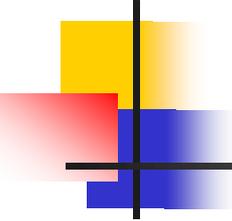
Exercícios

- Operador ?:
 - Exercício 3: Escreva um programa que leia dois números inteiros do teclado, e os imprima na tela em ordem ascendente.
 - Exercício 4: Escreva um programa que determina o valor absoluto de um número inteiro lido do teclado.
- Instrução switch
 - Exercício 5: Escrever um programa que receba 2 números e um caractere operador (+, -, *, /), realize a operação determinada e imprima na tela o resultado



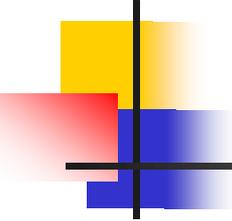
Exercícios

- Instrução for
 - Exercício 6: Escreva um programa que imprima os números pares entre 1 e 100.
- Instrução while
 - Exercício 7: Escreva um programa que coloque os números de 1 a 100 na tela na ordem inversa (começando em 100 e terminando em 1).
- Instrução do - while
 - Exercício 8: Escreva um programa que lê números do teclado enquanto esses números são menores que 10.
 - Exercício 9: Escreva um programa que testa a capacidade de adivinhar uma letra. Supondo que a letra 'secreta' seja a letra 'A', o programa deve ao final informar a quantidade de tentativas do usuário até acertar a letra



Outras Instruções

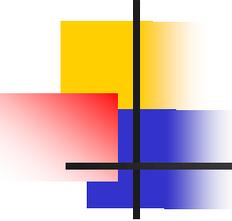
- Alterando o fluxo normal de execução de um laço
 - Instrução break - faz com que o laço seja interrompido imediatamente.
 - Instrução continue - faz com que o ciclo atual do laço seja terminado imediatamente. A execução continua com o próximo ciclo do laço.



Outras Instruções

- Exemplo com *break*:

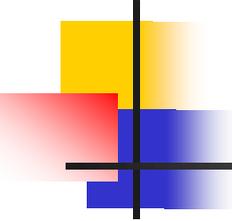
```
public class ExemploBreak {  
    public static void main(String[ ] args) {  
        int num = 1;  
        while (num <= 10) {  
            if (num == 6)  
                break;  
            System.out.println(" " + num);  
            num++;  
        }  
    }  
}
```



Outras Instruções

- Exemplo com *continue*:

```
public class ExemploContinue {  
    public static void main(String[ ] args) {  
        int num = 0;  
        while (num <= 10) {  
            num++;  
            if (num >= 6)  
                continue;  
            System.out.println(" " + num);  
        }  
    }  
}
```



Laços Infinitos

- Podem ser usados nas seguintes formas:

```
while (true) {  
    <instruções do laço>;  
}
```

```
for ( ; ; ) {  
    <instruções do laço>;  
}
```

- OBS: Este tipo de laço pode ser terminado com a instrução *break*.