

# Programação Orientada à Objetos

Aula 03

Laços

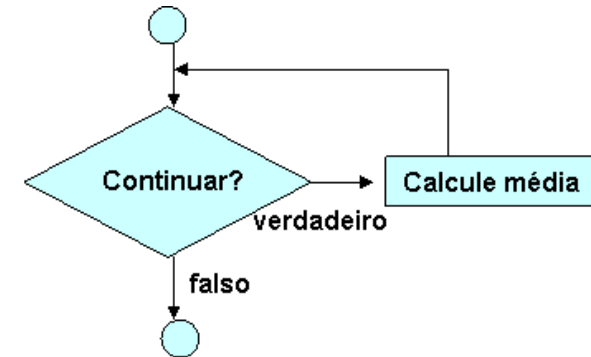
Henrique Poyatos

[henrique.poyatos@bandtec.com.br](mailto:henrique.poyatos@bandtec.com.br)

# Laços / Repetição

## Estrutura "while"

- A estrutura de repetição **while** permite especificar uma ação ou um bloco de ações que devem permanecer sendo repetidas enquanto determinada condição for verdadeira.
- *Exemplo:*
  - *Enquanto o usuário desejar continuar*
    - *calcule a média de cada aluno*
- O corpo da estrutura **while** pode ser uma instrução única ou um bloco de comandos.
- Quando a condição do comando **while** se tornar falsa, a ação (ou bloco) do comando será finalizada. O programa continuará com a ação imediatamente após o comando **while**.
- **IMPORTANTE:** você deve sempre prever o comando ou ação que tornará falsa a condição do comando **while**. Caso contrário seu programa entrará em loop.



# Laços / Repetição

## Estrutura "while"



```
// programa exemplo de utilização da estrutura de repetição while
import javax.swing.JOptionPane;
public class SwitchCase {
    public static void main(String arg[]) {
        int notaProva = 0, notaTrab = 0, notaProj = 0;
        float mediaFinal = 0;
        String esc;
        int escolha = 1;
        while ((escolha >=1) && (escolha <=4)){
            esc = JOptionPane.showInputDialog ("Digite sua Escolha : ");
            escolha = Integer.parseInt(esc);
            switch (escolha) {
                case 1: notaProva= Integer.parseInt(JOptionPane.showInputDialog("Nota da Prova: "));
                    break;
                case 2: notaTrab = Integer.parseInt(JOptionPane.showInputDialog("Nota do Trabalho: "));
                    break;
                case 3: notaProj = Integer.parseInt(JOptionPane.showInputDialog("Nota do Projeto: "));
                    break;
                default: if(escolha<4) mediaFinal = (notaProva + notaTrab + notaProj)/(3-escolha+1);
            }
            JOptionPane.showMessageDialog(null,"Media Final: "+
                mediaFinal,"Resultados",JOptionPane.INFORMATION_MESSAGE);
            mediaFinal = notaProva = notaTrab = notaProj = 0;
        } // fim do while
        System.exit( 0 );
    } // fim do main
} // fim da classe pública
```

# Laços / Repetição

## Estrutura "while"

---

```
...  
byte cont;  
  
while (cont <= 10)  
{  
    Console.Write(cont);  
    cont++;  
}
```

# Laços / Repetição

## Estrutura "while"

- A estrutura de repetição **do/while** permite repetir uma ação ou um bloco de ações até que determinada condição seja verdadeira. A diferença para a estrutura **while** está no fato de que **do/while** inicia pela execução do bloco e somente após a mesma analisa a condição.

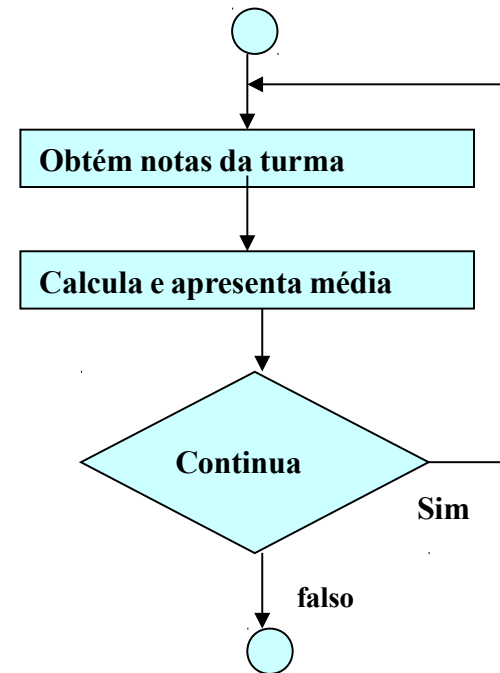
Exemplo:

Faça

*Obtenha as notas da turma*

*Calcula e Apresente a média*

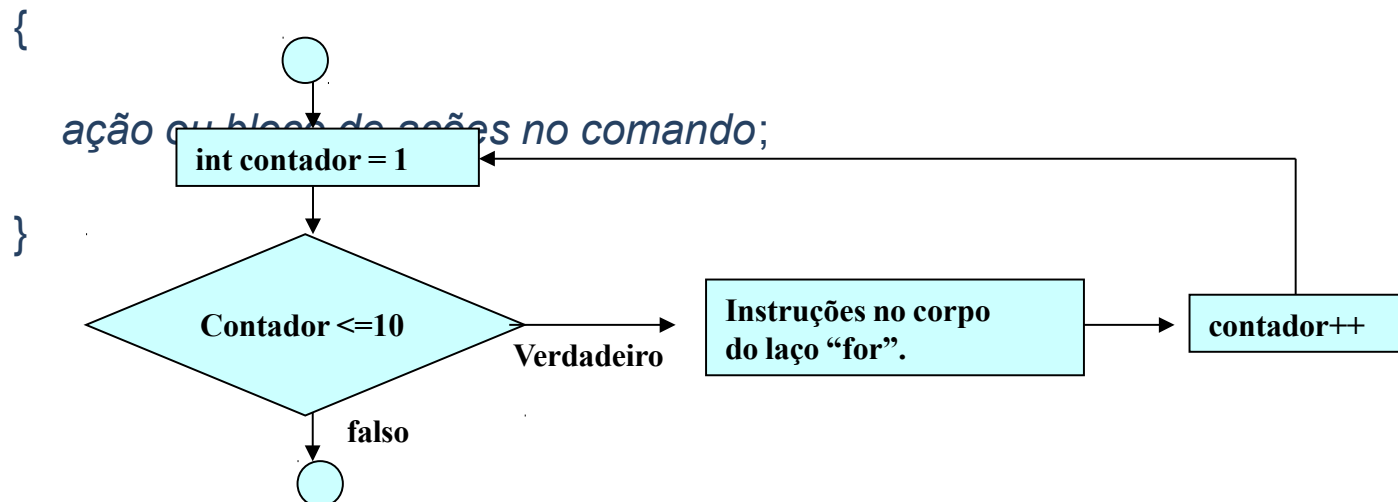
- Enquanto houver mais turmas para calcular a média
- O **do/while** sempre admite que a primeira interação ocorre antes da confirmação da condição
- O corpo da estrutura **do/while** pode ser uma instrução única ou um bloco de comandos.
- Quando a condição do comando **do/while** se tornar falsa, o programa continuará com a ação imediatamente após o comando **do/while**.
- IMPORTANTE: você deve sempre prever o comando ou ação que tornará falsa a condição do comando **do/while**. Caso contrário seu programa entrará em loop.



# Laços / Repetição

## Estrutura "for"

- A estrutura de repetição **for** permite repetir uma ação ou um bloco de ações com controle de contador ou da condição de permanência no looping.
- *Exemplo:*
  - Para o contador 'i' de 1 a 10 faça
  - Obtenha a nota do i-ésimo aluno
  - some a nota do i-ésimo aluno ao total;
  - Calcule a média da turma como sendo o total dividido por 10
- O comando **for** tem três seções de variáveis:  
**for** (inicializadores; condição de continuação; incrementos)



# Laços / Repetição

## Instruções **break** e **continue**

- As instruções **break** e **continue** modificam o comportamento das estruturas de repetição **while**, **for**, **do/while** ou **switch**. .
- A instrução **break** interrompe o laço (no caso das estruturas de repetição) e impede a execução de outros casos de um comando **switch**.
- *Exemplo:*

*Enquanto verdade permanente*

*(ex: `x == 1`, sem mudar `x`)*

*realize as operações ....*

*se condição de fim for alcançada **break**;*

*Fim do Enquanto*

- A instrução **continue** permite o salto do conjunto de operações, com retorno à expressão condicional do laço, reiniciando o mesmo (portanto, ao contrário do **break**, não interrompe o laço).
- Normalmente **break** e **continue** interrompem laços em que estão inseridos. Para interromper um conjunto aninhado de estruturas, deve-se utilizar **break** e **continue** rotulados (com denominação de blocos).

# Laços / Repetição

## Instruções break e continue

```
// for e continue
```

```
// Realizando um laço que somará apenas os números pares.
```

```
for (int contador=0; contador < numero; contador++) {  
    if ((contador % 2) != 0) //resto da divisão por dois não é zero  
        continue;  
    pares++;  
    saidaFinal += contador + " ";  
}
```

```
// do/while e continue
```

```
do {  
    if ((contador % 2) != 0) // resto da divisão por dois não é zero  
        continue;  
    pares++;  
    saidaFinal += contador + " ";  
} while (++contador < numero);
```



Copyright © 2013 Prof. Henrique Poyatos

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do professor.