

INTRODUÇÃO A LÓGICA

Curso Superior Tecnológico em

Sistemas para Internet

Semana – Laço de Repetição For, While e Do-While

Instituto Federal do Tocantins

Prof. Helder – helder@ifto.edu.br

Laço de repetição

Fazer uso de laço de repetição reduz de forma significativa o tamanho do código.

Força o processador a executar um conjunto de instruções até que uma condição seja satisfeita.

Laço FOR

```
for(int i=0; i<=5; i++)
    System.out.println(i);
```

```
for(int i=0;;i++){
    System.out.println(i);
    if(i==5)
        break;
}
```

```
int i=0;
for(i=0;;i++){
    System.out.println(i);
    if(i==5)
        break;
}
```

```
int i=0;
for(;;){
    System.out.println(i);
    if(i==5)
        break;
    i++;
}
```

Imprime de 0 a 5 na tela

Imprime de 0 a 5 na tela.
 Condição de saída do laço não é uma instrução do for como no exemplo 1, mas sim uma instrução dentro do bloco do for

Imprime de 0 a 5 na tela.
 Igual ao exemplo anterior, exceto que a variável “i” foi criada fora do bloco do for

Imprime de 0 a 5 na tela.
 O diferencial aqui é que o for não recebe parâmetro algum

Laço FOR - Instruções

Sequencia e quantidade de vezes da execução da instrução contidas como parâmetros para o for:

```
for(instrução1; instrução 2; instrução 3)
```

Primeira vez executando o laço:
instrução1 -> instrução 2;

A partir da segunda vez, executando o laço:
Instrução 3 -> Instrução 2

OBS: PARA TODAS AS INSTRUÇÕES SEU USO É OPCIONAL

Laço for – Papel das instruções

```
for(instrução1; instrução 2; instrução 3)
```

Instrução1 – criar e/ou inicializar variáveis, pode fazer isso com mais de uma variável, basta separá-las com vírgula. Essas variáveis podem ser internas ao bloco do for ou não.

Instrução 2 – condição de saída – define quando sair do laço.

Instrução 3 – operações para alterar o conteúdo da variável, podendo essa variável ser interna ao bloco do for ou não.

Laço for – Papel das instruções - Exemplos

<pre>for(int i=5,j=0;i<100;i++){ }</pre>	<pre>for(int i=5,j=0;i<100;i++, j+=2){ }</pre>	<pre>int i,j; for(i=5,j=0;i<100;i++, j+=2){ }</pre>
Criando duas variáveis no laço	Criando duas variaveis no laço e as incrementando	Inicializando duas variáveis externas no laço e as incrementando

Laço for – Uso do break e continue

```
int i,j;
for(i=0, j=0; i <=3; i++){
    j++;
}
System.out.println(j);
```

Impressão do valor de j == 4

```
int i,j;
for(i=0,j=0; i <=3; i++){
    if(i==2)
        break;
    j++;
}
System.out.println(j);
```

Impressão do valor de j ==2

```
int i,j;
for(i=0,j=0; i <=3; i++){
    continue;
    j++;
}
System.out.println(j);
```

Impressão do valor de j == 0

Laço for – abreviado

Utilizado apenas para ler dados de uma variável que aponta para um conjunto de dados.

```
int x[] = {10,20,30,40,50};  
for(int j:x)  
    System.out.println(j);
```

↑ Tem algum elemento em x?

Se sim

faça uma cópia do elemento e armazene em j

Aponte para o próximo elemento contido em x

Imprima j

Se não

Sai do laço

Laço for – abreviado

Utilizado apenas para ler dados de uma variável que aponta para um conjunto de dados.

Imprime todos os elementos do vetor na tela	Imprime todos os elementos do vetor na tela
<pre>int x[] = {10,20,30,40,50}; for(int i =0; i < x.length; i++) System.out.printl(x[i]);</pre>	<pre>int x[] = {10,20,30,40,50}; for(int j:x) System.out.printl(j);</pre>
Adicionando 1 a cada elemento do vetor	Adicionando 1 a cada elemento do vetor
<pre>int x[] = {10,20,30,40,50}; for(int i =0; i < x.length; i++) x[0]++;</pre>	Não tem como fazer com for each não armazena os índices de onde pegou os dados

Laço While

```
while(condição){\n    ....\n}
```

condição – representa um valor Boolean ou qualquer expressão que resulte em um valor Boolean;

Para executar as instruções contidas no bloco do laço while a **condição** deverá ser verdadeira. A sequencia de instruções no bloco do while serão executadas repetidamente até que a **condição** seja falsa;

Laço Do-While

```
do{  
    ....  
} while(condição);
```

condição – representa um valor Boolean ou qualquer expressão que resulte em um valor Boolean;

Para reexecutar as instruções contidas no bloco do laço do-while a **condição** deverá ser verdadeira. A sequencia de instruções no bloco do-while serão executadas repetidamente até que a **condição** seja falsa;

O bloco de instruções será executado ao menos uma vez.

Exemplo – Impressão de 0 a 2 na tela

Cenário 1	Cenário 2	Cenário 3
<pre>for(int i=0; i<=2; i++){ System.out.print(i); }</pre>	<pre>int i=0; while(i<=2){ System.out.print(i); i++; }</pre>	<pre>int i=0; do{ System.out.print(i); i++; } while(i<=2);</pre>
Resultado da execução	Resultado da execução	Resultado da execução
<ul style="list-style-type: none"> - 0 - 1 - 2 	<ul style="list-style-type: none"> - 0 - 1 - 2 	<ul style="list-style-type: none"> - 0 - 1 - 2

Exemplo – Advinhe o número que estou pensando

Cenário 1	Cenário 2	Cenário 3
<pre>Scanner dd = new Scanner(System.in); int num; int nMagico =5; for(;){ System.out.print("Numero: "); num = dd.nextInt(); if(num == nMagico){ System.out.println("Acertou\n"); break; }else System.out.println("Errou. Tente novamente"); }</pre>	<pre>Scanner dd = new Scanner(System.in); int num; int nMagico =5; while(true){ System.out.print("Numero: "); num = dd.nextInt(); if(num == nMagico){ System.out.println("Acertou\n"); break; }else System.out.println("Errou. Tente novamente"); }</pre>	<pre>Scanner dd = new Scanner(System.in); int num; int nMagico =5; do { System.out.print("Numero: "); num = dd.nextInt(); if(num == nMagico){ System.out.println("Acertou\n"); break; }else System.out.println("Errou. Tente novamente"); }while(true);</pre>

Resumo

- Não existe um laço mais rápido que outro, existem contextos onde um é melhor que outro.
- Se sabe quantas vezes o laço vai ocorrer - use for
- Se não sabe quantas vezes o laço vai ocorrer – use while
- Se precisar que as instruções do bloco do laço seja executada ao menos uma vez – use do-while