Estrutura de repetição for

A estrutura de repetição "for", ou então, laço de repetição, iterador é a estrutura mais utilizada quando precisamos executar diversas vezes um mesmo bloco de código. Isso porque, conseguimos facilmente declarar, inicializar e incrementar valores no cabeçalho da estrutura. Logo, nós temos que o nosso código acaba por ficar encapsulado, o que confere segurança na execução do laço como também, uma melhor legibilidade do código e maior produtividade do programador.

Para melhor entendermos a estrutura de repetição "for", devemos dividi-la em 3 partes:

```
for(parte1, parte2, parte3)
```

Inicialmente, nós temos que "for" é o nome da estrutura e os valores que estão contidos no parêntesis a frente, são os valores do cabeçalho da estrutura.

Vamos descrever o que cada parte da estrutura é responsável:

parte1: região reservada para declaração e inicialização de variáveis. É importante notar, que podemos declarar quantas variáveis forem necessárias. Normalmente, declaramos uma única variável, porém, somos livres para declarar e inicializar quantas forem necessárias.

parte2: na segunda região, ou então, na região central, é onde colocamos a condição para que o laço seja repetido. A condição a ser estabelecida pode ser qualquer uma, porém, é comum e recomendável colocar uma condição utilizando a variável ou variáveis declaradas da <parte1>. Então, podemos dizer que a região central é a mais importante, até porque, é a responsável por determinar quantas vezes o laço será repetido.

<parte3>: a terceira e última região é um local reservado e normalmente utilizado para incrementar valores a variável que foi declarada na <parte1> e comparada na <parte2>. Não é obrigatório a incrementação de valores a cada ciclo, porém, é recomendável que assim o façamos, até porque, esse é o propósito da estrutura de repetição "for".

Exemplo

```
public class Aula0029 {

    public static void main(String[] args) {
        //Looping FOR I
```

```
        for(int i = 10; i <= 10; ++i){
            System.out.println( i );

        /*
        * for(parte1; parte2; parte3)
        * parte1: é onde nós declaramos uma variável
        * parte2: é onde nós colocamos uma condição
         * para qeu continue ou seja terminado
         * parte3: é onde nós incrementamos a nossa
variável
        * */
        }
    }
}
```

Neste exemplo damos continuidade a aula anterior onde iniciamos o estudo dos laços de repetição "for". Ainda que a estrutura e sua definição sejam simples, a quantidade de combinação que podemos utilizar são enormes e consequentemente, a quantidade de recursos que podemos extrair também.

Agora, o objetivo é esmiuçar, ainda mais, cada parte da estrutura, bem como, as diferentes maneiras que podemos utilizar o laço de repetição "for".

É importante observar mais uma peculiaridade que não foi dita no vídeo: a declaração e a inicialização da variável que está no cabeçalho da instrução for é executado uma única vez!

Exemplo

```
for(int i = 0; i!=10; i++)
  // executa o bloco
```

A cada ciclo o curso de execução do nosso programa percorrerá o cabeçalho da estrutura for, verificara o valor e incrementa uma unidade, porém, a primeira parte do cabeçalho só sera execauutada uma única vez.

Exemplo

```
public class Aula0030 {
    public static void main(String[] args) {
//        Looping FOR II

        /*
        * for(parte1; parte2; parte3)
        * parte1: é onde nós declaramos uma variável
        * parte2: é onde nós colocamos uma condição
         * para qeu continue ou seja terminado
         * parte3: é onde nós incrementamos a nossa
variável
        * */

        for(int i = 0; //inicialização
            i <= 10;//expressão
                ++i//Atualização
                ){
            System.out.println( i );
        }
    }
}
```

Material de apoio segundo vídeo logica

Porque IF e ELSE e tão importante na Lógica de programação?

A estrutura condicional *if/else* permite ao programa avaliar uma expressão como sendo verdadeira ou falsa e, de acordo com o resultado dessa verificação, executar uma ou outra rotina.

Na linguagem Java o tipo resultante dessa expressão deve ser sempre um *boolean*, pois diferentemente das demais, o Java não converte *null* ou inteiros como 0 e 1 para os valores *true* ou *false*.

Sintaxe do if/else:

```
if (expressão booleana) {
    // bloco de código 1
} else {
    // bloco de código 2
}
```

As instruções presentes no bloco de código 1 serão executadas caso a expressão booleana seja verdadeira. Do contrário, serão executadas as instruções presentes no bloco de código 2.

O Java utiliza as chaves como delimitadores de bloco e elas têm a função de agrupar um conjunto de instruções. Apesar do uso desses delimitadores ser opcional caso haja apenas uma linha de código, ele é recomendado, pois facilita a leitura e manutenção do código, tornando-o mais legível.

Else if

Complementar ao *if/else* temos o operador *else if*. Esse recurso possibilita adicionar uma nova condição à estrutura de decisão para atender a lógica sendo implementada.

Sintaxe do *if/else* com *else if*:

```
if (expressão booleana 1) {
    // bloco de código 1
} else if (expressão booleana 2) {
    // bloco de código 2
} else {
    // bloco de código 3

}
```

Dessa forma, se a expressão booleana 1 for verdadeira, o bloco de código 1 será executado. Caso seja falsa, o bloco de código 1 será ignorado e será testada a expressão booleana 2. Se ela for verdadeira, o bloco de código 2 será executado. Caso contrário, o programa vai ignorar esse bloco de código e executar o bloco 3, declarado dentro do *else*.

Podemos utilizar quantos *else if* forem necessários. Entretanto, o *else* deve ser adicionado apenas uma vez, como alternativa ao caso de todos os testes terem falhado.

Exemplo pratico

Suponha que você está desenvolvendo um software para controle de estoque que precisa informar como está a quantidade de itens de cada produto: se suficiente, para quantidades superiores a 100; em alerta, para quantidades entre 100 e 50; e abaixo do ideal, para quantidades menores do que 50. Como programar esse código?

Exemplo de uso de *if/else*:

```
int estoque = //valor recuperado do sistema

if (estoque >= 100) {
    System.out.println(“Produto com quantidade suficiente.”);
} else if (estoque < 100 && estoque > 50) {
    System.out.println(“Alerta: Avaliar possibilidade de novo pedido.”);
} else {
    System.out.println(“ATENÇÃO! Faça um novo pedido.”);
}
```