

Vídeo sobre fluxo condicional e de repetição

Olá! Vamos conhecer alguns fluxos condicional e de repetição?

Iniciando pelo fluxo condicional, temos o **if**, seu uso indica para o programa que desejamos iniciar um teste. Assim, ele avaliará determinada condição entre parênteses (bloco). Caso essa condição retorne um valor verdadeiro (*true*), o programa executará as instruções definidas no código.

Neste exemplo, a variável idade é definida com o tipo *int* (número inteiro) e valor 8.

A condição é: se a idade informada for menor que 10, imprimirá em tela (exibirá) a mensagem “Ainda é uma criança”.

```
*FLUXO CONDICIONAL
*Utilização do IF
*@author Professor Carlos Moreno
*/
public class ControleFLuxo {
    /*Método principal da classe*/
    public static void main (string [] args) {

        int idade = 8; // A variável idade é definida como 8 (8anos)
        if (idade < 10) { // Se a idade for menor que 10 (anos)
            System.out.println ("Ainda é uma criança"); // Imprima em
            tela a mensagem

        } //fim do método
    } //fim da classe
```

O **else**, por sua vez, significa que, caso o bloco seja falso, executará o comando *else*. Isto é, se o valor da variável, assim como o resto da divisão, for igual a zero, então o número (definido na variável) será par. Do contrário, será exibida a mensagem “este número é ímpar”.

Observe, neste exemplo, que a variável “número” é definida com o tipo *int* e valor 8.

A condição é: se o valor da variável “número” (definida como 8), dividindo por 2 (sinal de divisão é %), com o resultado da divisão (resto da divisão) igual a zero, então o número informado é par.

```
*FLUXO CONDICIONAL
*Utilização do IF, ELSE
*@author Professor Carlos Moreno
*/

public class ControleFLuxo2 {
    /*Método principal da classe */
    public static void main (String[] args) {

        int numero = 8; //define o valor da variável em 8
        if ((numero % 2) == 0) { //verifica se o resto da divisão é
            igual a 0
            System.out.println ("Este é um número par"); //Imprima em
            tela a mensagem
        } else {
            System.out.println ("Esse é um número impar") ; //Imprima em
            tela a mensagem

        }

        //fim do método
        //fim do método
        //fim da classe
    }
}
```

Mas, caso seja necessário avaliar outras condições, podemos usar o *else if*, até que alguma das condições esteja satisfeita no código.

A variável “idade” é definida com o tipo *int* e recebe o valor 7.

A condição é: o valor da variável “idade” (definida em 7) é menor ou igual a 7?

Se a condição for verdadeira, o código exibirá em tela a mensagem “Ainda é uma criança”.

Por outro lado, se o valor da variável for maior que 7 e menor ou igual a 18, então o código exibirá em tela a mensagem “É um adolescente”.

Já se o valor da variável for maior que 18 e menor ou igual a 60, o código exibirá em

tela a mensagem 'É um adulto'.

Se alguma das condições anteriores não for satisfeita, será exibida a mensagem "Está na melhor idade", ou seja, o valor da variável será maior que 60.

```
*FLUXO CONDICIONAL
*Utilização do IF, ELSE IF, ELSE
*@author Professor Carlos Moreno
*/

public class ControleFLuxo3 {
    /*Método principal da classes */
    public static void main (String[] args) {

        int idade = 7;
        if (idade <= 7){ //a idade é menor ou igual a 7?
            System.out.println("Ainda é uma criança"); //se verdadeiro,
            imprima

        } else if (idade > 7 && idade <= 18) //a idade é maior que 7
        e menor ou igual a 18?
            System.out.println("É um adolescente"); //se verdadeiro,
            imprima

        } else if (idade > 18 && idade <= 60) //a idade é maior que
        18 e menor ou igual a 60?
            System.out.println("É um adulto"); //se verdadeiro, imprima

        } else {
            System.out.println("Está na melhor idade"); //senao,
            imprima

        }

    } //fim do método
} //fim do método
} //fim da classe
```

Agora falando sobre os laços ou loops, ou estruturas de repetição, estas são utilizadas na execução, de modo repetido, de determinado bloco ou certa instrução, enquanto uma condição estiver sendo satisfeita.

Dentro disso, a estrutura **while** é utilizada quando desejamos executar esses laços em determinado bloco, enquanto a condição for verdadeira.

O código traz uma variável "i" do tipo *int* (inteiro), recebendo o valor zero.

A estrutura de repetição *while* informa que, enquanto o valor da variável "i" for menor que 2, o código exibirá o resultado em tela, mostrando os valores que foram incrementados em "i++", ou seja, serão exibidos em tela os valores 0, 1 e 2.

```
*FLUXO DE REPETIÇÃO
*Utilização do WHILE (ENQUANTO)
*@author Professor Carlos Moreno
*/
import java.util.arraylist;
//Importamos a classe ArrayList para poder fazer uso do
vetor e armazenar elementos.

public class While {
    /*Método principal da classes*/
    public static void main (String [] args) {
        //Fluxo de repetição usando While
        int i = 0; //a variável i recebe o valor zero (0)
        while (i < 2) { // enquanto i for menor que 2
            System.out.println(i); //imprima valor de i
            i++; //incremento da variável i.
        } //fim do método
    } //fim da classe
```

A estrutura **do...while**, por sua vez, inicia a execução da instrução, imprimindo e incrementando o valor da variável "i", enquanto este for verdadeiro.

O código traz uma variável "i", com o valor 3.

A estrutura de repetição *do...while* informa que, enquanto o valor da variável "i" for menor que 5, o código será executado enquanto a condição não ultrapassar do valor 4.

Isto é, a estrutura de repetição executará o programa enquanto o valor inserido for menor que 5. Ou seja, na saída estarão os números 3 e 4.

```
*FLUXO DE REPETIÇÃO
*Utilização DO...WHILE (FAÇA...ENQUANTO)
*@author Professor Carlos Moreno
*/
import java.util.arraylist;
//Importamos a classe ArrayList para poder fazer uso do
vetor e armazenar elementos.

public class DoWhile {
    /*Método principal da classes*/
    public static void main (String[] args) {
        //Fluxo de repetição usando While
        int i = 5; //a variável i recebe o valor cinco (5)
        do {
            System.out.println(i); //imprima valor de i i++; //
            incremento da variável i.
        }
        while (i < 3) {           // enquanto i for menor que 2
        }//fim dométodo
    }//fim da classe
```

Para finalizar, a estrutura *for* é semelhante à *do...while*.

O código traz uma variável "i", iniciando o valor em 0 e informando que o valor da variável é menor que 5, incrementando-a. Isto é, será exibido em tela o resultado da variável "i" de zero a cinco, tal como 0, 1, 2, 3 e 4.

O número 5 não será exibido, pois a variável está definida como menor que 5 (< 5). Caso estivesse como <=5, ele seria exibido.

```
/**
 *FLUXO DE REPETIÇÃO
 *Utilização FOR
 *@author Professor Carlos Moreno
 */

class FOR {
public static void main(String args []) {
for (int i = 0; i < 5; i++ ) { //ordem crescente, de 0 à 5
System.out.println(i);
}
}
}
```

Agora é com você! Pratique estes fluxos modificando as condições iniciais mostradas aqui para se familiarizar cada vez mais com a linguagem.