UNIP
UNIVERSIDADE PAULISTA

UNICLUSTER

Estruturas de Controle e Repetição

Carlos Arruda Baltazar
UNIP – Cidade Universitária

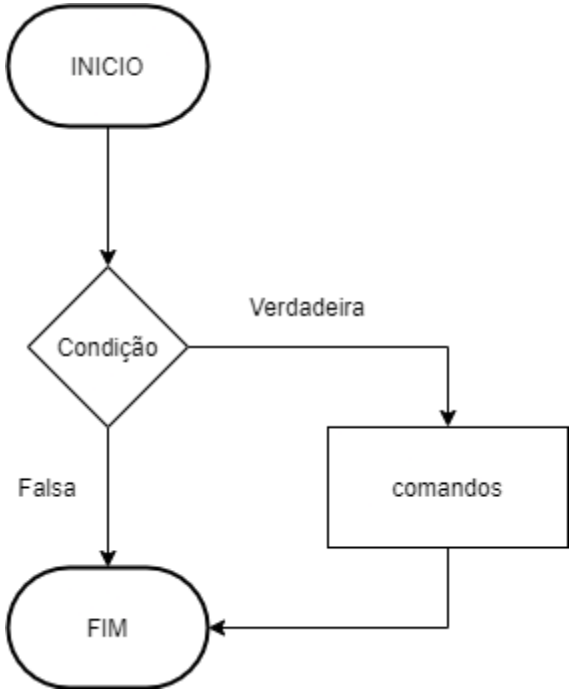
Estrutura de controle

A estrutura de controle de decisão possibilita um desvio condicional para escolher de um grupo de ações e estruturas a serem executadas quando determinados testes lógicos são ou não satisfeitas, podendo resultar dois valores lógicos: verdadeiro ou falso.

Desvio Condicional Simples

O desvio condicional simples executa um comando ou vários comandos se a condição for verdadeira. Se a condição for falsa, a estrutura é finalizada sem executar os comandos.

Desvio Condicional Simples

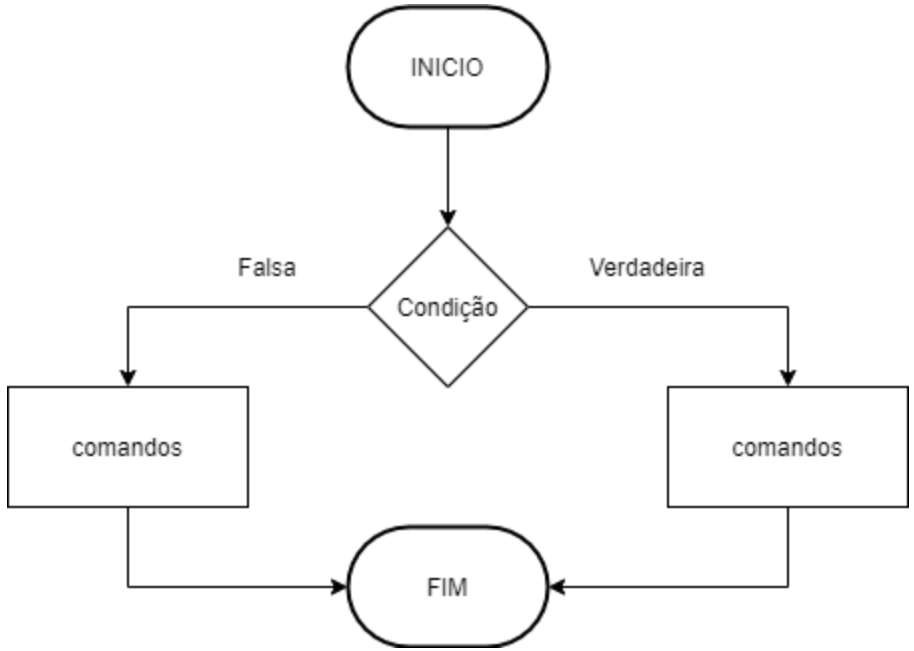



```java
package Pck_teste;

public class DesvioCondicional
{
    public DesvioCondicional()
    {
        int a = 3;
        int b = 5;

        if(a > b)
        {
            System.out.println(a);
        }

        if(a>b)
            System.out.println(a);
    }
}
```

Desvio Condicional Composto

O desvio condicional simples executa um comando ou vários comandos se a condição for verdadeira. Se a condição for falsa, a estrutura é finalizada sem executar os comandos.

Desvio Condicional Composto

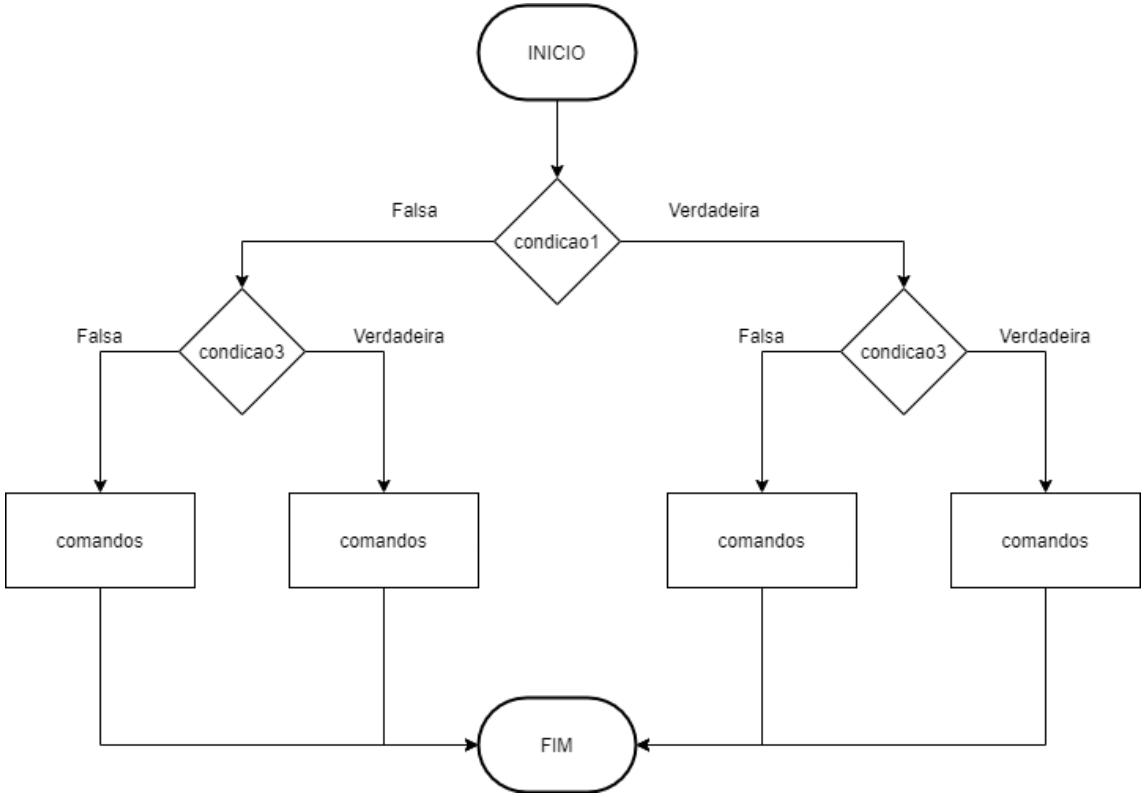



```java
package Pck_teste;

public class DesvioCondicional
{
    public DesvioCondicional()
    {
        int a = 3;
        int b = 5;

        if(a > b)
        {
            System.out.println(a);
        }
        else
        {
            System.out.println(b);
        }

        if(a>b)
            System.out.println(a);
        else
            System.out.println(b);
    }
}
```

Desvio Condicional Aninhado

Dentro de um desvio condicional aninhado é perfeitamente possível utilizarmos mais de uma linha de comando, ou até mesmo outros desvios condicionais. Existem situações em que os caminhos para a tomada de uma decisão acabam formando uma espécie de árvore com diversas ramificações, onde cada caminho é um conjunto de ações. Nesses casos podemos recorrer à utilização de vários desvios condicionais embutidos uns dentro dos outros

Desvio Condicional Simples




```java
package Pck_teste;

public class DesvioCondicional
{
    public DesvioCondicional()
    {
        int a = 3;
        int b = 5;
        int c = 7;

        if(a > b)
        {
            if(b > c)
                System.out.println(a + ";" + b + ";" + c);
            else
                System.out.println(a + ";" + c + ";" + b);
        }
        else
        {
            if(a > c)
                System.out.println(b + ";" + a + ";" + c);
            else
                System.out.println(b + ";" + c + ";" + a);
        }
    }
}
```

Operadores Relacionais

Operadores	Operação
==	Igual a
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a

Operadores Lógicos

Operadores	Operação
!=	Negação
&&	“e”
||	“ou”

Estrutura de repetição

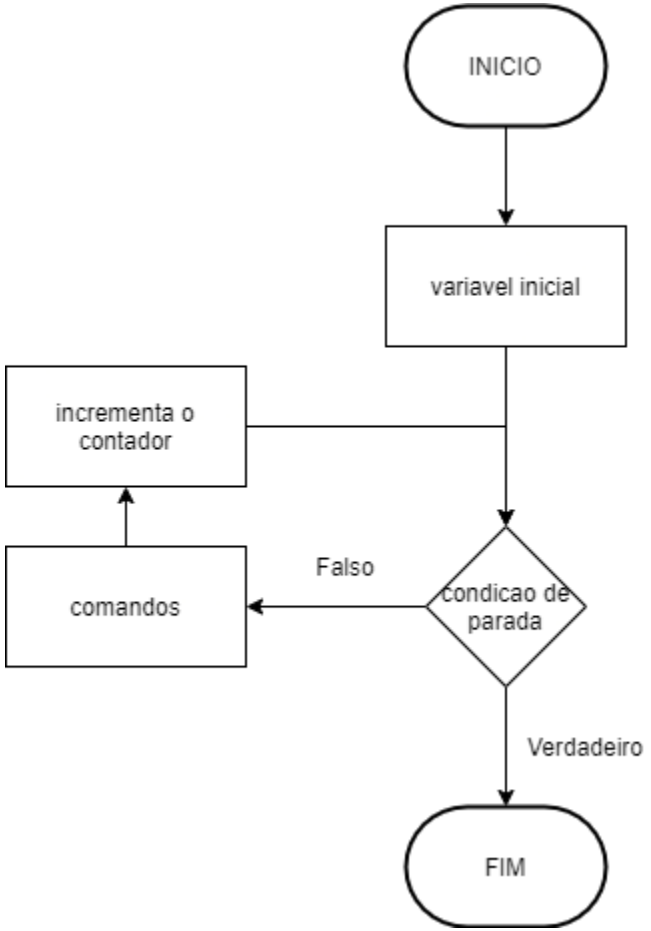
Dentro da lógica de programação é uma estrutura que permite executar mais de uma vez o mesmo comando ou conjunto de comandos, de acordo com uma condição ou com um contador.

São utilizadas, por exemplo, para repetir ações semelhantes que são executadas para todos os elementos de uma lista de dados, ou simplesmente para repetir um mesmo processamento até que a condição seja satisfeita.

For

A estrutura de repetição “For” é utilizada para executar um conjunto de comandos repetidamente por um número x de vezes. Uma situação inicial, uma condição de parada e uma ação a ser executada a cada repetição são especificadas na instrução. Para esta estrutura, uma variável é inicializada com um valor, essa variável é utilizada para controlar a quantidade de vezes em que o conjunto de comandos será executado. E ao final do conjunto de comandos, a variável sempre sofrerá uma alteração, aumentando ou diminuindo de acordo com a lógica utilizada.

For

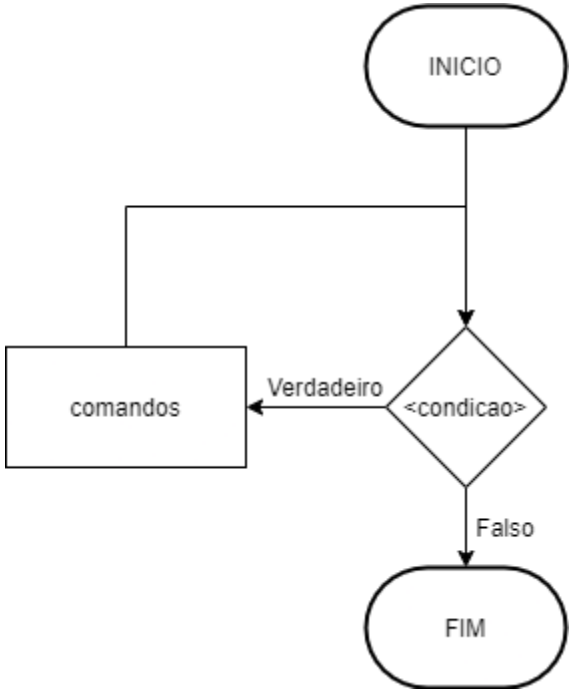



```java
package Pck_teste;

public class Repeticao
{
    public Repeticao()
    {
        int x = 0;

        for (int i = 0; i < 10; i++)
        {
            x += 1;// ou x = x + 1; ou x++;
        }
    }
}
```

While

Na construção de algoritmos, hora ou outra torna-se necessário executar blocos de comandos mais de uma vez. Ou mesmo executar repetidamente um bloco de comandos até que alguma condição seja atendida. A partir dessa necessidade surgem as estruturas de repetição, também conhecidas como laços. Então, vamos tratar de forma especial a estrutura de repetição WHILE. Seu funcionamento é tão simples quanto um desvio condicional, a diferença é que os passos dentro deste bloco, são repetidos enquanto uma condição for satisfeita.

While




```
package Pck_teste;

public class Repeticao
{
    public Repeticao()
    {
        int x = 0;

        while (x < 10)
        {
            x += 1;// ou x = x + 1; ou x++;
        }
    }
}
```

OBRIGADO