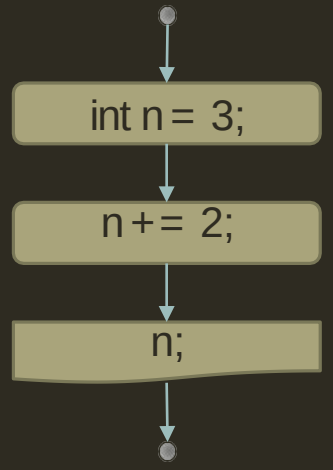
Estruturas Condicionais

FUNDAMENTOS DE PROGRAMAÇÃO

Estruturas Condicionais(Simples, Compostas, Aninhadas, Escolha Múltipla)

SEQUÊNÇIA

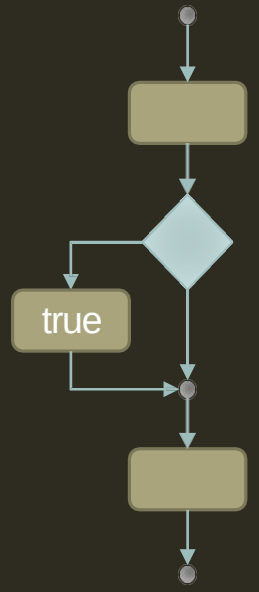
Sequência é a execução “passo a passo” de um algoritmo de forma ordenada e lógica, em uma sequência não podemos saltar passos exemplo de 1 para 5, esse é o princípio de um algoritmo sendo assim tem que ser feito de maneira sequençial para alcançar um determinado resultado. Exemplo:



CONDIÇÕES SIMPLES

Condições Simples são aquelas em que usamos apenas o bloco do **SE(if)**, esse bloco é executado apenas quando for verdadeiro tendo então um desvio do seu fluxo normal de execução do programa, caso seja falso o programa segue seu fluxo normal, então vamos para um exemplo:

Usando Fluxograma



CONDIÇÕES SIMPLES

Em portugol temos o seguinte:

```
se(condição)
        <instrução>
fimSe
```

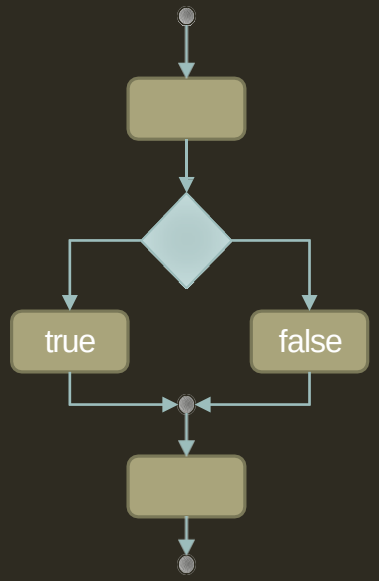
Traduzindo para o Java temos:

```
if(condição){
        <instrução>;
}
```

CONDIÇÕES COMPOSTAS

Condições Compostas são aquelas em que usamos um bloco do **SE(if)**, esse bloco é executado apenas quando for verdadeiro e um bloco **SENAO(else)** executado apenas quando for falso, terminado volta para o seu fluxo normal. Então vamos para um exemplo:

Usando Fluxograma



CONDIÇÕES COMPOSTAS

Em portugol temos o seguinte:

```
se(condição)
        <instrução>
senao
        <instrução>
fimSe
```

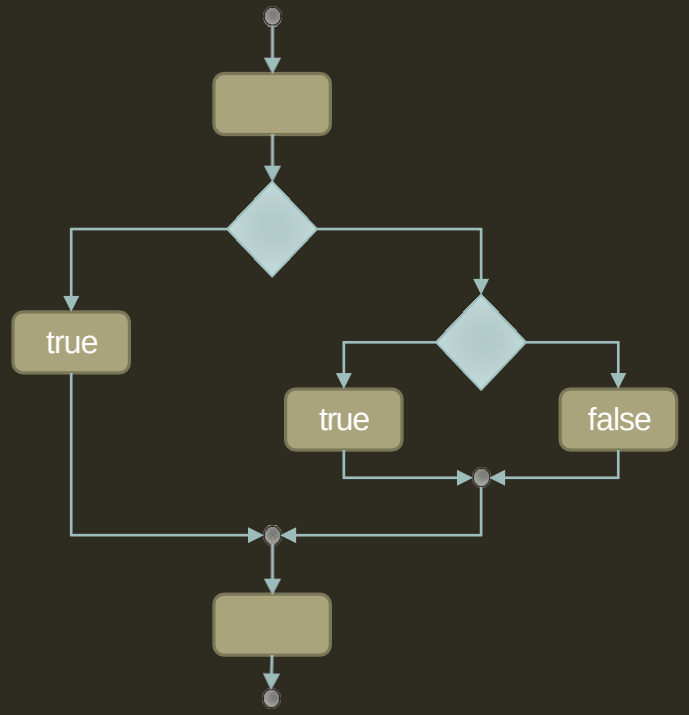
Traduzindo para o Java temos:

```
if(condição){
        <instrução>;
}else{
        <instrução>;
}
```

CONDIÇÕES ANINHADAS

Condições Aninhadas são condições que contém outras condições dentro dela, fazendo isso dizemos que estamos aninhando uma dentro da outra, geralmente esse aninhamento aparece no bloco do SENAO(else). Vamos para um exemplo:

Usando Fluxograma



CONDIÇÕES ANINHADAS

Em portugol temos o seguinte:

```
se(condição)
        <instrução>
senao
        se(condição)
                <instrução>
        senao
                <instrução>
        fimSe
fimSe
```

CONDIÇÕES ANINHADAS

Traduzindo para o Java temos:

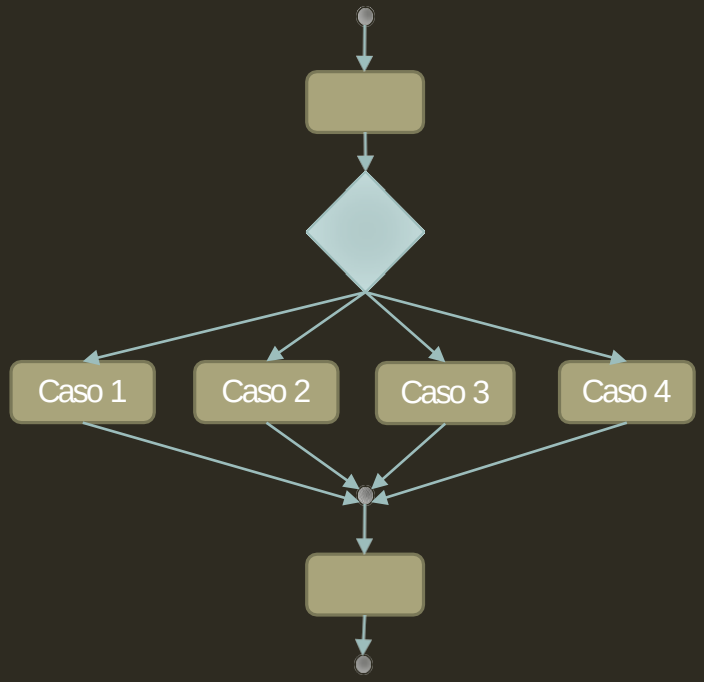
```
if(condição){
        <instrução>;
}else{
        if(condição){
                <instrução>;
        }else{
                <instrução>;
        }
}
```

NB: Podemos aumentar o nível de aninhamento de acordo com o problema a resolver!

ESCOLHA MÚLTIPLA

Escolha Múltipla é uma estrutura de condição usado para valores fixos(específicos) sejam eles númericos ou caracteres(string ou char), ou seja não aceita valores intervalares feitos pelos operadores relacionais e/ou lógicos. Vamos a um exemplo:

Usando Fluxograma



ESCOLHA MÚLTIPLA

Em portugol temos o seguinte:

```
Escolha (expressão)
        caso valor 1:
                <instrução>
        caso valor 2:
                <instrução>
        caso valor 3:
                <instrução>
        caso valor 4:
                <instrução>
        outroCaso:
                <instrução>
fimEscolha
```

ESCOLHA MÚLTIPLA

Traduzindo para o Java temos:

```
switch (expressão){
        case valor 1:
                <instrução>;
                break;
        case valor 2:
                <instrução>;
                break;
        case valor 3:
                <instrução>;
                break;
        case valor 4:
                <instrução>;
                break;
        default:
                <instrução>;
}
```

ESCOLHA MÚLTIPLA

Break serve para terminar/sair da estrutura condicional switch(escolha), sendo obrigatório o uso dele nesse tipo de estrutura logo após a um bloco de instruções em cada case.

Default é um bloco da estrutura switch(escolha) é o outro caso, ele serve no momento que nenhuma das expressões inseridas faz parte de um case sendo o fluxo direcionado para o bloco default.

FIM

Feito por:

- Anselmo Nhamage