

ESTRUTURAS DE CONTROLE

Estruturas Condicionais - IF

```
if(condição)  
    instrução;
```

```
if (5 > 3){  
    System.out.print("Maior");  
}
```



**Se 5 for maior
que 3, será
impresso na
tela "Maior".**

A condição é verificada. Se a condição for satisfeita, as instruções que se encontram dentro do bloco IF (entre chaves) serão executadas.

A condição entre parênteses deve ser uma expressão que resulte em um valor booleano (true ou false).

Estruturas Condicionais - IF Else

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

```
if (5 > 3){  
    System.out.print("Maior");  
}  
else {  
    System.out.print("Menor");  
}
```



Se 5 for maior que 3,
será impresso
na tela "Maior".

Senão, será impresso
na tela "Menor".

A condição é verificada. Se a condição for satisfeita, as instruções dentro do bloco IF (entre chaves), serão executadas, senão, as instruções do bloco ELSE (entre chaves) serão executadas.

Estruturas Condicionais - Else IF

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

```
If (nota < 4){  
    System.out.print("Reprovado");  
}  
else if (nota < 7){  
    System.out.print("Recuperação");  
}  
else{  
    System.out.print("Aprovado");  
}
```

Se nota for menor que 4, será impresso na tela "Reprovado".

Senão, se nota for menor que 7, será impresso na tela "Recuperação".

Senão, será impresso na tela "Aprovado".

A primeira condição é verificada. Se a condição for satisfeita, as instruções do bloco IF, serão executadas, senão, será testada a segunda condição (ELSE IF). Se essa condição for satisfeita, as instruções do bloco ELSE IF, serão executadas. Se a condição ELSE IF não for satisfeita, serão executadas as instruções do bloco ELSE.

Estruturas Condicionais - Aninhadas

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

```
if (nota < 4,9){  
    System.out.print("Reprovado");  
}  
else{  
    if ((nota > 5) && (nota < 6,9)){  
        System.out.print("Recuperação");  
    }  
    else{  
        System.out.print("Aprovado");  
    }  
}
```

Se nota for menor que 4,9 será impresso na tela "Reprovado".

Senão:

Se nota for maior que 5 e menor que 6,9 será impresso na tela "Recuperação".
Senão, será impresso na tela "Aprovado".

A primeira condição é verificada. Se a condição for satisfeita, então as instruções do bloco IF, serão executadas.

Se a primeira condição não for satisfeita, será executado o bloco ELSE, onde há outro IF, que terá suas condições testadas.

Estruturas Condicionais - Switch Case

Utilizada para testar uma única expressão que produz um resultado, ou, então, o valor de uma variável em que está armazenado um determinado conteúdo.

switch (variável ou valor) → Definimos a variável ou valor que desejamos comparar.

```
{
```

case valor 1: → Se o valor declarado neste case for igual ao contido no switch, código 1 será executado.

```
    // código 1
    break;
```

case valor 2:

```
    // código2
    break;
```

```
}
```

O mesmo comportamento se aplica ao segundo case. **Caso o valor contido no switch não seja atendido em uma das condições**, nenhum bloco será executado.

Os cases não aceitam operadores lógicos. Portanto, não é possível fazer uma comparação. Isso limita o case a apenas valores definidos.

Comando Break

O **switch** executa seu bloco em **cascata**. Ou seja, se a variável indicar para o primeiro caso e dentro do escolha tiver 4 casos, o escolha executará todos os outros 3 casos.

O comando **break** é utilizado para especificar a última linha de código a ser executada dentro da condição.

Se não declarado, os códigos implementados dentro dos cases subsequentes serão executados.

```
switch (opcao)
{
case valor 1:
    // código 1
    break;
case valor 2:
    // código 2
    break;
case valor 3:
    // código 3
    break;
case valor 4:
    // código 4
    break;
}
```



Comando Default

Utilizado quando precisamos definir um fluxo alternativo para situações em que o valor contido no switch não seja atendido por nenhum case.

Caso o valor do switch não seja igual a um dos valores contidos nos cases, o sistema irá executar o código implementado no bloco default;

Seu uso é opcional, mas requer o break se for utilizado.

```
switch (variável ou valor)
{
    case valor 1:
        // código 1
        break;
    case valor 2:
        // código 2
        break;
    case valor 3:
        // código 3
        break;
    case valor 4:
        // código 4
        break;
    default:
        // código 5
        break;
}
```

Estruturas Condicionais - Switch Case

```
switch (opcao)
{
    case 1:
        System.out.print ("Você Escolheu Inclusão");
        break;
    case 2:
        System.out.print ("Você Escolheu Alteração");
        break;
    case 3:
        System.out.print ("Você Escolheu Exclusão");
        break;
    default:
        System.out.print ("Opção Inválida");
        break;
}
```

Caso o valor de “opcao” seja 1, o programa deverá escrever “Você Escolheu Inclusão”.

Caso o valor de “opcao” seja 2, o programa deverá escrever “Você Escolheu Alteração”.


Caso o valor de “opcao” seja 3, o programa deverá escrever “Você Escolheu Exclusão”.

Caso nenhum case for satisfeito, será escrito “Opção Inválida”.

Switch Case com Vários Valores

Para executar o mesmo código para valores diferentes, podemos utilizar a estrutura da seguinte forma

```
switch (variavel ou valor)
{
    case valor 1:
    case valor 2:
    case valor 3:
        // código 1
        break;
    case valor 4:
    case valor 5:
    case valor 6:
        // código 2
        break;
}
```



```
switch (mes) {
```

```
    case "Janeiro":
```

```
    case "Março":
```

```
    case "Maio":
```

```
    case "Julho":
```

```
    case "Agosto":
```

```
    case "Outubro":
```

```
    case "Dezembro":
```

```
        System.out.print ("Este mês tem 31 dias");
```

```
        break;
```

```
    case "Fevereiro":
```

```
        System.out.print ("Este mês tem 28 ou 29 dias");
```

```
        break;
```

```
    default:
```

```
        System.out.print ("Este mês tem 30 dias");
```

```
        break;
```

```
}
```

Caso o valor de “mes” seja Janeiro, Março, Maio, Julho, Agosto, Outubro, Dezembro, o programa deverá escrever “Este Mês tem 31 dias”.

Caso o valor de mes seja Fevereiro, o programa deverá escrever “Esse Mês tem 28 ou 29 dias”.

Se nenhum case for satisfeito, será escrito “Este mês tem 30 dias”.



Estruturas de Repetição

Forma de executar blocos de comandos somente sob determinadas condições, mas com a opção de repetir o mesmo bloco quantas vezes for necessário. As estruturas de repetição são úteis, por exemplo para:

- Repetir uma série de operações semelhantes que são executadas para todos os elementos de uma lista
- Repetir um mesmo processamento até que uma certa condição seja satisfeita.

São estruturas de repetição: **While, Do... While e For**



Estrutura While

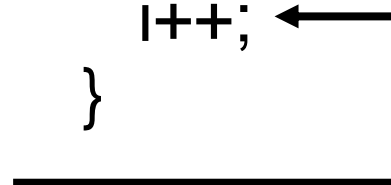
Repete a execução de um bloco de sentenças enquanto uma condição permanecer verdadeira.

```
while (condição) {  
    código  
}
```

Não se esqueça de modificar a variável usada na condição, caso contrário, o loop nunca terminará!

```
int i = 0;
```

```
while (i < 5) {  
    System.out.println(i);  
    i++;  
}
```





Estrutura While

```
Scanner entrada = new Scanner(System.in);
```

```
System.out.print ("Digite seu salario: ");
```

```
float salario = entrada.nextFloat();
```

```
System.out.print ("Digite o ano atual: ");
```

```
int ano = entrada.nextInt();
```

```
int limite = ano + 10;
```

```
while (ano < limite) {
```

```
    ano++;
```

```
    salario = salario + ((salario * 10) / 100);
```

```
    System.out.printf("Meu salário em " + ano + " será de R$ %.2f\n", salario);
```

```
}
```

Digite seu salário atual, o ano atual e veja quanto você vai receber pelos próximos dez anos, caso seu salario aumente 10% a cada ano.

Enquanto ano for menor que limite (que é o ano atual mais 10 anos), será calculado o salario a cada ano e exibido)



Estrutura Do...While

O do...while é uma variante do while, que irá executar o bloco de código uma vez, antes de verificar se a condição é verdadeira, e vai repetir o loop enquanto a condição while for verdadeira.

```
do {  
    código  
} while (condição)
```

```
int i = 0;
```

```
do {  
    System.out.println(i)  
    i++;  
}  
while (i < 5);
```

Não se esqueça de modificar a variável usada na condição, caso contrário, o loop nunca terminará!





Estrutura Do...While

```
Scanner entrada = new Scanner(System.in);
boolean continuar = true;
int opcao;

do{
    System.out.print ("*** MENU *** \n\n1. Cadastrar \n2. Ler
    \n3. Atualizar \n4. Excluir \n0. Sair \n\nInsira sua opção: ");
    opcao = entrada.nextInt();

    if (opcao == 0) {
        continuar = false;
        System.out.println("Programa Finalizado");
    }
    else if (opcao >= 5){
        System.out.println("Valor Inválido\n");
    }

} while(continuar);
```

Ex: Exibir um menu com 5 opções:

- 1. Cadastrar**
- 2. Ler**
- 3. Atualizar**
- 4. Excluir**
- 0. Sair**

O menu sera exibido ao menos uma vez. E enquanto o usuário não digitar 0, o menu será repetido.



Estrutura FOR

```
for (inicialização; teste; atualização) {  
    sentença  
}
```

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

Inicialização: Executada antes da execução do bloco de código, atribui o valor inicial à variável contadora.

Teste: Avalia a expressão, que determina se o valor da variável contadora está dentro do limite desejado. Caso positivo, o bloco de sentenças é executado

Atualização: Altera o valor da variável contadora, após o código ter sido executado.



Estrutura FOR

Exemplo: Exibir valores pares de 0 a 10

```
for (int i = 0; i <= 10; i = i + 2) {  
    System.out.print(i + " ");  
}
```

Para i igual a zero, enquanto i for menor ou igual a zero, o valor de i recebe ele mesmo mais 2 a cada passagem do laço.

Exemplo: Exibir valores de 10 a 0

```
for (int cont = 10; cont >= 0; cont--) {  
    System.out.println(cont);  
}
```

Para cont igual a dez, enquanto cont for maior ou igual a zero, sera descontado 1 de cont a cada passagem do laço

Material desenvolvido pelo
Prof. Rafael da Silva Polato
rafael.polato@etec.sp.gov.br