



Técnicas de Programação e Algoritmos - TPA

– Estrutura de Repetição

Teste no Início; Teste no Final e Variável de Controle

- JAVA -

Objetivos

- O objetivo das estruturas de repetição é executar uma instrução (ou conjunto de instruções) repetidas vezes, enquanto (ou até que) uma dada condição seja satisfeita. Nesta aula, você identificará as características de uma estrutura de repetição e a sua classificação em:
 - Estrutura de repetição com teste condicional no início;
 - Estrutura com teste condicional no final e
 - Estrutura com variável de controle.

Exemplos das estruturas

1

```
// Teste condicional no inicio  
while(op != 0) {  
    // Instruções  
}
```

2

```
// Teste condicional no fim  
do{  
    // Instruções  
}while(op != 0);
```

3

```
// Numero pré-definido de interações  
for(int i=0; i<=10; i++){  
    // Instruções  
}
```

Repetição com Teste no Início

While (**Teste Condicional For Verdadeiro**)

```
{  
  Comando 1;  
  Comando 2;  
  Comando 3;  
  Comando 4  
  Comandos N;  
}
```

O comando **While** fará a repetição do bloco de instruções que há dentro dele, enquanto sua condição for verdadeira.

A variável e/ou expressão do teste condicional, poderá ser do tipo inteiro, do tipo double, do tipo String ou do tipo lógico.

O comando **While**, abre com “{” e obrigatoriamente deverá ser finalizado com o “}”, e necessariamente deverá recalculer a expressão ou ler a variável responsável pelo teste condicional dentro do bloco de repetição.

Teste Condicional no início

Laço while()

```
public class LacoWhile {  
    public static void main( String[] args ) {  
        int i = 0;  
        //laço while() com bloco de código definido  
        while( i < 10 ) {  
            System.out.println( "Linha: " + i );  
            i++;  
        }  
    }  
}
```

A expressão é avaliada antes de executar o bloco de código
Ele repete enquanto a expressão for verdadeira (true)

Repetição com Teste no Final

O comando **Do While** fará a repetição do bloco de instruções que há dentro dele, **até que** sua condição for verdadeira, ou seja, enquanto a condição tiver resposta falsa ela continua a repetição

A variável e/ou expressão do teste condicional, poderá ser do tipo inteiro, do tipo double, do tipo String ou do tipo lógico.

O comando **Do While** , obrigatoriamente, deverá ser finalizado com o **Teste Condicional**, e necessariamente deverá recalculer a expressão ou ler a variável responsável pelo teste condicional dentro do bloco de repetição.

Do {

Comando 1;

Comando 2;

Comando 3;

Comando 4

Comandos N;

} While(**Teste Condicional For Verdadeiro**);

Teste Condicional no final

Laço do / while()

```
public class LacoWhile {  
    public static void main( String[] args ) {  
        int i = 0;  
        //laço do / while() com bloco de código definido  
        do {  
            System.out.println( "Linha: " + i );  
            i++;  
        } while( i < 10 );  
    }  
}
```

O bloco é executado ao menos um vez

Após a primeira repetição é que a expressão é avaliada

Repetição com Variável de Controle

for (**variável** de **valor inicial**; até **valor final** ;passo **incremento**) {

Comando 1;

Comando 2;

Comando 3;

Comando 4

Comandos N;

};

O comando **For** fará a repetição do bloco de instruções que há dentro dele, o número de vezes indicados pelo valor inicial e valor final.

A variável que irá fazer o controle do laço, deverá, obrigatoriamente ser do tipo inteiro.

Obs.: O incremento poderá ser crescente ou decrescente.

Teste Condicional com variável de controle

Laço for()

A sua estrutura é definida como a seguir:

```
for( iniciação; condição; incremento ) {  
    bloco_de_código_a_executar  
}
```

```
public class LacoFor {  
    public static void main( String[] args ) {  
        for( int i=0; i < 10; i++ ) {  
            System.out.println( "Linha: " + i );  
        }  
    }  
}
```

Comparação entre as Estruturas de Repetição

Estruturas	Comando	Resultado do Teste para repetição	Quantidade de Execução
Repetição com Teste no Início.	While	Teste verdadeiro	0 ou mais vezes.
Repetição com Teste no Final.	Do ... While	Teste falso	Pelo menos 1 vez.
Repetição com Variável de Controle.	For	*****	Determinado pelo limite estabelecido no 2º parâmetro.

Cláusula Break

Cláusula break

Aborta a execução de um laço, quando executado.

```
public class ClausulaBreak {  
    public static void main( String[] args ) {  
        char letras[] = { 'A', 'B', 'C', 'D', 'E' };  
        int i;  
        for( i=0; i<letras.length; i++ ) {  
            if( letras[i] == 'C' ) {  
                break;  
            }  
        }  
        System.out.println( "Último índice: " + i );  
    }  
}
```

Lista de exercícios

- 1 – Exibir todos os números pares existentes entre os números 1 e 20.
- 2 – Efetuar a tabuada do número informado pelo usuário.
- 3 – Exibir a “**quantidade**” de números existentes entre 100 e 125.
- 4 – Através de um valor inicial e um valor final informados pelo usuário, Apresente a soma dos números ímpares deste intervalo.
- 5 - Escreva um programa que exiba na tela em ordem decrescente, apenas os números pares existentes entre dois números digitados pelo usuário (inclusive eles).

Bom trabalho!