



ETE PORTO DIGITAL LÓGICA DE PROGRAMAÇÃO

Profª Msc. Aline Chagas

Aula teórica


O que vamos aprender?

- Estruturas de repetição: Enquanto, Faca Enquanto e Para.



Aula teórica

Estruturas de repetição

- Caso os programas necessitem ser executados mais de uma vez, como podemos resolver esse problema?
 - Todos os programas apresentados até agora não permitem que sejam executados mais de uma vez!
 - Se quisermos repetir algum programa precisamos sair do programa e executá-lo novamente e isso é inviável!
 - Precisamos repetir alguns programas!
- 

Aula teórica

Estruturas de repetição

- é uma estrutura com controle de fluxo de execução, que permite repetir diversas vezes um mesmo trecho do programa.
- Do mesmo jeito que na estrutura de decisão, a estrutura de repetição depende do teste de uma condição.
- São três os tipos de estruturas de repetição: Enquanto, Faça enquanto e Para.

Estruturas de repetição: Enquanto

- A função do comando **enquanto** é: executar uma lista de comandos enquanto uma determinada condição for verdadeira.
- A sintaxe é respectivamente a palavra reservada **enquanto**, a **condição** a ser testada entre parênteses, e entre chaves a **lista de instruções** que se deseja executar.



Estruturas de repetição: Enquanto

Aula teórica

```
logico condicao = verdadeiro  
enquanto (condicao)  
{  
    //Executa a as instruções dentro do laço enquanto a  
    condicao for verdadeira  
}
```



Estruturas de repetição: Enquanto

Aula teórica



Aula prática

Estruturas de repetição: Enquanto

```
programa
{

    funcao inicio()
    {
        caracter parar
        parar = 'N'

        enquanto (parar != 'S'){
            escreva("Deseja parar o laço? S (sim) ou N (não)")
            leia(parar)
        }
    }
}
```


Estruturas de repetição: Enquanto

- Recomendada quando não sabemos o número de repetições.

Sintaxe / regra

```
enquanto condição  
    comando 1  
    comando 2
```

Regra:

V: executa e volta

F: pula fora

Aula prática

Estruturas de repetição: Enquanto

```
Algoritmo "teste_enquanto"
```

```
Var
```

```
    x, soma : inteiro
```

```
Inicio
```

```
    soma <- 0
```

```
    escreva("Digite o primeiro numero: ")
```

```
    leia(x)
```

```
    enquanto x <> 0 faca
```

```
        soma <- soma + x
```

```
        escreva("Digite outro numero: ")
```

```
        leia(x)
```

```
    fimenquanto
```

```
    escreval("SOMA = ", soma)
```

```
Fimalgoritmo
```

Estruturas de repetição: Enquanto

Problema "crescente" (*adaptado de URI 1113*)

Leia uma quantidade indeterminada de duplas de valores inteiros X e Y. Escreva para cada X e Y uma mensagem que indique se estes valores foram digitados em ordem crescente ou decrescente. O programa deve finalizar quando forem digitados dois valores iguais.

Exemplo:

```
Digite dois numeros:  
5  
4  
DECRESCENTE!  
Digite outros dois numeros:  
3  
8  
CRESCENTE!  
Digite outros dois numeros:  
2  
2
```

Estruturas de repetição: Enquanto

Problema "senha_fixa" (adaptado de URI 1114)

Escreva um programa que repita a leitura de uma senha até que ela seja válida. Para cada leitura de senha incorreta informada, escrever a mensagem "Senha Invalida! Tente novamente:". Quando a senha for informada corretamente deve ser impressa a mensagem "Acesso Permitido" e o algoritmo encerrado. Considere que a senha correta é o valor 2002.

Exemplo:

```
Digite a senha: 2312
Senha Invalida! Tente novamente: 2010
Senha Invalida! Tente novamente: 1022
Senha Invalida! Tente novamente: 2002
Acesso permitido!
```