

# Estrutura de Repetição

Aula II

# Relembrando

- O que é uma estrutura de repetição?
- O que é um contador?
- Comandos de Estrutura de Repetição em Pascal
  - while (visto na aula passada)
  - for
  - repeat

# for

- O comando **for** permite que o programador informe a quantidade de vezes que o laço será repetido, definindo o **valor inicial** do contador e o **valor final**, que será a condição de parada. A incrementação do contador fica invisível ao programador.

## Forma Geral

```
for cont:=valorInicial to valorFinal do  
    begin  
        //instruções  
    end;
```

# Exemplo

```
Program exemplo_for;
```

```
var
```

```
    cont:integer;
```

```
Begin
```

```
    for cont:=1 to 5 do
```

```
        begin
```

```
            write(cont);
```

```
        end;
```

```
End.
```

# Exemplo

Program exemplo2\_for;

var

cont, soma:integer;

Begin

soma := 0;

for cont:=1 to 5 do

begin

soma := soma + cont;

end;

write(soma);

End.

Que valor será impresso na linha 11?

```
1. Program exemplo3_for;  
2.   var  
3.     cont, soma:integer;  
4.   Begin  
5.     soma := 0;  
6.     for cont:=1 to 10 do  
7.       begin  
8.         soma := soma + cont;  
9.         cont := cont + 1;  
10.      end;  
11.    write(soma);  
12.  End.
```

O que acontece ao executar este algoritmo?

```
Program exemplo4_for;
```

```
var
```

```
    cont:integer;
```

```
Begin
```

```
    for cont:=1 to 10 do
```

```
        begin
```

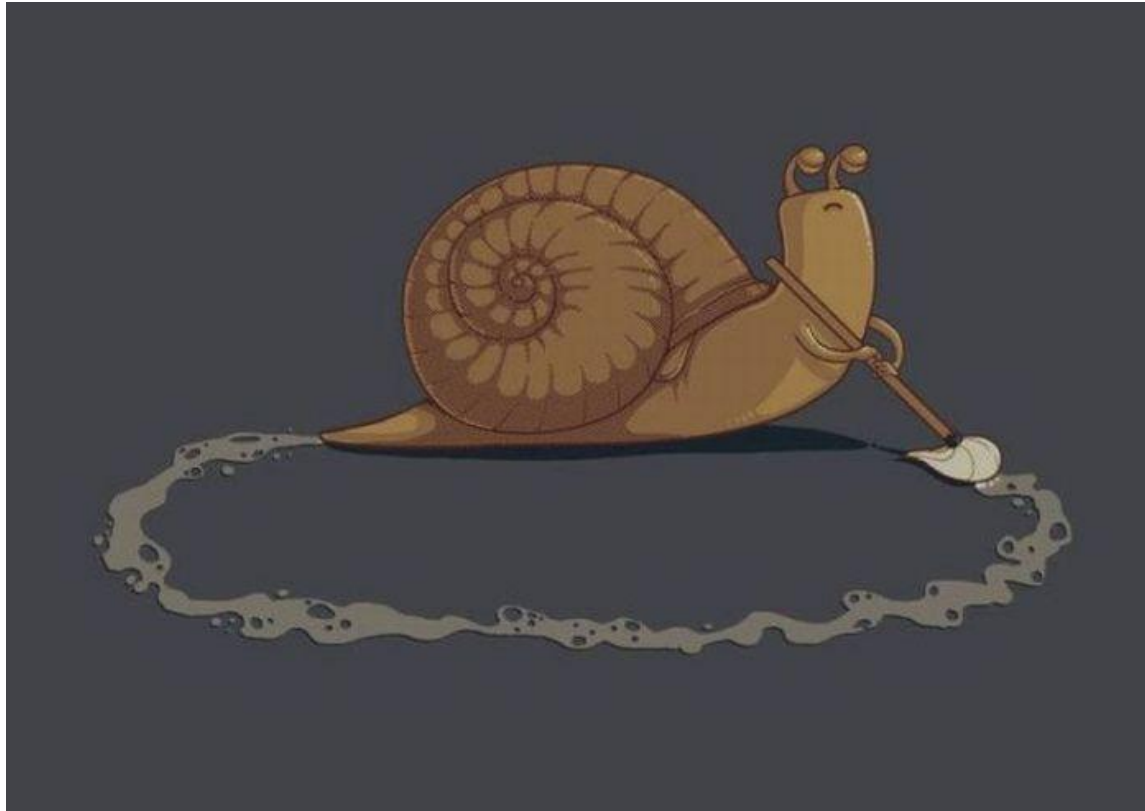
```
            write(cont);
```

```
            cont := cont - 1;
```

```
        end;
```

```
End.
```

# Loop eterno



Pois a condição de parada nunca  
será verdadeira, ou seja, o  
contador nunca será 10.



# Problema

- Faça um algoritmo que leia 10 números e some todos eles. Ao final, informe o valor da soma dos números.

# Resposta

```
Program soma_numeros;  
  var  
    cont, num, soma:integer;  
Begin  
  soma := 0;  
  for cont:=1 to 10 do  
    begin  
      write('Digite o ',cont,' numero:');  
      read(num);  
      soma := soma + num;  
    end;  
    write(soma);  
End.
```

# Problema

- Faça um algoritmo que leia 10 números e some todos os números pares. Ao final, informe o valor da soma.

# Resposta

```
Program soma_numeros;  
  var  
    cont, num, soma:integer;  
Begin  
  soma := 0;  
  for cont:=1 to 10 do  
    begin  
      write('Digite o ', cont, ' numero:');  
      read(num);  
      if (num mod 2 = 0) then  
        soma := soma + num;  
    end;  
    write(soma);  
End.
```

# repeat

- O comando **repeat** irá executar uma sequência de instruções **ATÉ** uma condição ser verdadeira.
- Este comando não verifica uma condição antes de executar a primeira vez, a condição só é verificada ao final de cada interação (loop).

## Forma Geral

```
repeat  
//instruções  
until (condicao);
```

# Exemplo

## Comando WHILE

```
Program loop ;  
  var  
    cont:integer;  
  Begin  
    cont := 1;  
    while (cont < 5) do  
      begin  
        write(cont);  
        cont := cont + 1;  
      end;  
  End.
```

## Comando REPEAT

```
Program loop ;  
  var  
    cont:integer;  
  Begin  
    cont := 1;  
    repeat  
      write(cont);  
      cont := cont + 1;  
    until (cont = 5);  
  End.
```

# Exemplo

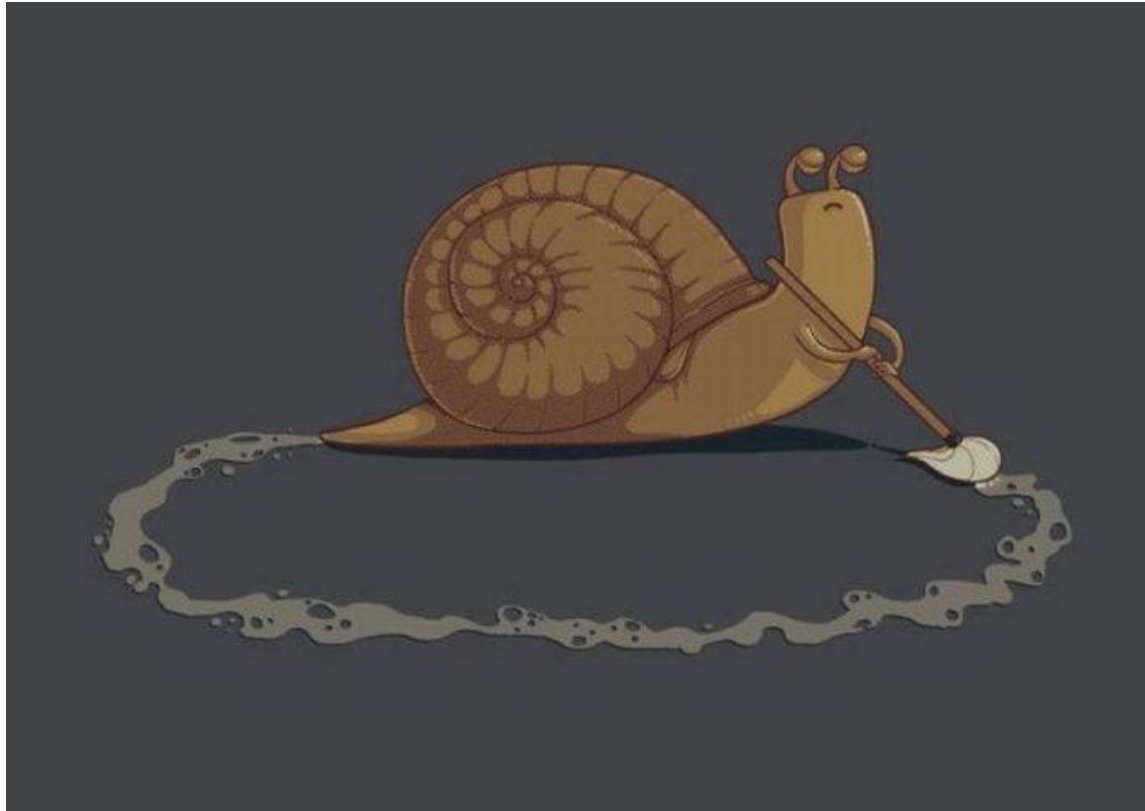
```
Program numero_dez;  
  var  
    num:integer;  
Begin  
  repeat  
    write('Numero: ');  
    read(num);  
  until (num = 10);  
  write(num);  
End.
```

O que acontece ao executar este algoritmo? O que será impresso?

```
Program loop_eterno;  
  var  
    contador:integer;  
Begin  
  contador := 0;  
  repeat  
    write(contador);  
    contador := contador + 2;  
  until (contador = 5);  
  write(idade);  
End.
```



# Loop eterno



Pois a condição de parada nunca  
será verdadeira!

# Problema

- Declare uma variável inteira **NUM** e preencha-a com um número do teclado. Obrigue o valor de **NUM** ser um valor **positivo**.  
Ao final, imprima **NUM**.

# Resposta

```
Program numero_positivo;  
var  
    num:integer;  
Begin  
    repeat  
        write('Numero: ');  
        read(num);  
    until (num > 0);  
    write(num);  
End.
```

# Problema

- Modifique o algoritmo anterior, obrigando o valor de **NUM** ser **par**.
- Ao final, imprima **NUM**.

# Resposta

```
Program numero_par;  
var  
    num:integer;  
Begin  
    repeat  
        write('Numero: ');  
        read(num);  
    until (num mod 2 = 0);  
    write(num);  
End.
```

# Problema

- Ler a idade de uma pessoa e obrigar ser válida.

Vamos aceitar idades válidas entre 1 e 120 (inclusive)

# Resposta

```
Program idade_valida ;  
  var  
    idade:integer;  
Begin  
  repeat  
    write('Idade: ');  
    read(idade);  
  until ((idade > 0) and (idade < 121));  
  write(idade);  
End.
```

- Exercício da Lista IV