COMANDOS DE REPETIÇÃO E DESVIO CONDICIONAIS

Prof. André Vignatti – DINF - UFPR

IMPRIMINDO SEQUÊNCIA DE NÚMEROS

Problema: imprimir todos os números entre 1 e 5.

```
program contar;

begin
        write (1);
        write (2);
        write (3);
        write (4);
        write (5);
end.
```

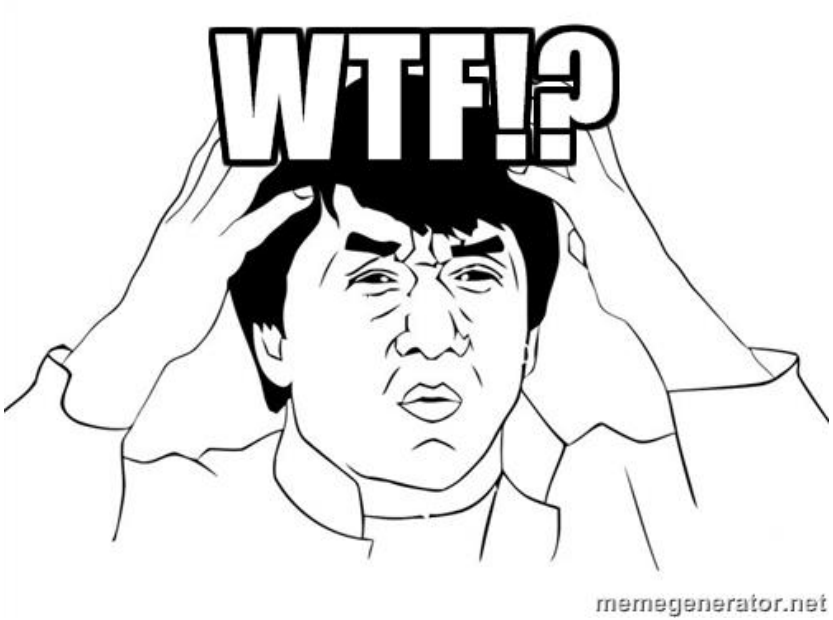
IMPRIMINDO SEQUÊNCIA DE NÚMEROS

Problema: imprimir todos os números entre 1 e 20.

```
program contar;

begin
        write (1);
        write (2);
        write (3);
        write (4);
        write (5);
        write (6);
        write (7);
        write (8);
        write (9);
        write (10);
        write (11);
        write (12);
        write (13);
        write (14);
        write (15);
        write (16);
        write (17);
        write (18);
        write (19);
        write (20);
end.
```

IMPRIMINDO SEQUÊNCIA DE NÚMEROS

Problema: imprimir todos os números entre 1 e 1000000.



Calma: veremos hoje! ☺

IMPRIMINDO SEQUÊNCIA DE NÚMEROS

Problema: imprimir todos os números entre 1 e 5.

```
program contar;
var i: integer;

begin
        i:= 1;

        write (i);
        i:= i + 1;

        write (i);
        i:= i + 1;

        write (i);
        i:= i + 1;

        write (i);
        i:= i + 1;

        write (i);
        i:= i + 1;
end.
```

SOLUÇÃO COM COMANDO DE REPETIÇÃO "WHILE"

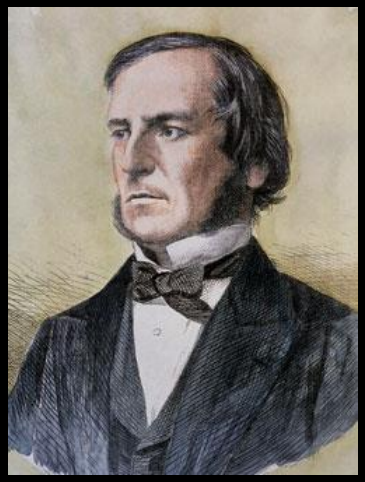
```
program contar;
var i: integer;

begin
        i:= 1;
        while i <= 30000 do
        begin
                write (i);
                i:= i + 1;
        end;
end.
```

Comando de repetição "while"

Formato:
while <*expressão booleana*> **do**

Expressão booleana:
ou **verdadeiro**, ou **falso**

LÓGICA BOOLEANA (≈1847)



George Boole

Lógica booleana = Fazer "contas" com lógica

Objetivo: descobrir grandes verdades (ou mentiras) a partir de pequenas verdades/mentiras

Álgebra Booleana:

- operadores: e, ou, não
- operandos: V, F

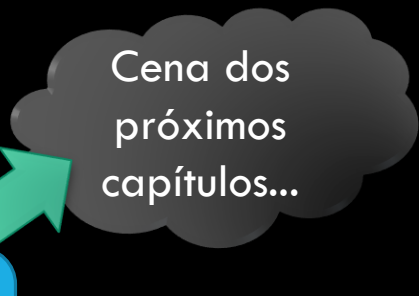
Exemplos:

- "*2 é par*" é VERDADEIRO
- "*3 é par*" é FALSO
- "*2 é par* e *3 é par*" é FALSO
- "2 é par e 4 é par e 6 é par e 8 é par e 10 é par e 12 é par e 14 é par e 16 é par e 18 é par e 20 é par e 22 é par e 24 é par e 26 é par . . ." é VERDADEIRO

CURSO RÁPIDO DE EXPRESSÕES BOOLEANAS EM PASCAL

Exemplos (em Pascal):

- 1=2
- 5<3
- 3<=3
- 2+3 <> 5
- not(9>10)
- (media >= 7) and (percentualFaltas <= 0.75)
- (idade >= 65) or (tempo servico >= 25)
- not (velocidade >= 60)



Pascal tem também **variáveis do tipo booleano:** armazenam V/F

QUADRADOS

Problema: Imprimir uma tabela com os valores de x e x^2, $\forall x$ tal que $1 \leq x \leq 30$

```
program quadrados;
var i: integer;

begin
        i:= 1;
        while i <= 30 do
        begin
              write (i,' ',i*i);
              i:= i + 1;
        end;
end.
```

LEMBRANDO: SOMA DE UM PAR DE NÚMEROS

Problema: ler dois números do teclado, e imprimir a soma deles na tela

```
program soma2;
var a,b: integer;

begin
        read (a);
        read (b);
        write (a+b);
end.
```

SOMA DE PARES DE NÚMEROS

Problema: Ler vários pares de números e imprimir a soma de cada par.

- O que são "**vários pares**"???

Problema mal definido!

Vamos considerar **duas interpretações**

1ª INTERPRETAÇÃO: SOMA DE 30 PARES

1ª Interpretação: ler 30 pares de números do teclado e imprimir, para cada par, a soma deles

```
program soma2variasvezes_v1;
var a,b,cont: integer;
(* cont conta os numeros lidos *)
begin
      cont:= 1;
      while cont <= 30 do
      begin
            read (a);
            read (b);
            writeln (a+b);
            cont:= cont + 1;
      end;
end.
```

Isso é um **comentário:**

- trecho ignorado pelo compilador
- ajuda na legibilidade do código

2ª INTERPRETAÇÃO: SOMA ATÉ PAR (0,0)

2ª Interpretação: ler pares de números e imprimir, para cada par, a soma deles. O algoritmo deve parar a execução quando os dois números lidos forem iguais a zero

```
program soma2variasvezes_v2;
var a,b: integer;

begin
      read (a);
      read (b);
      while (a <> 0) or (b <> 0) do
      begin
            writeln (a+b);
            read (a);
            read (b);
      end;
end.
```

Sinal de "diferente"

DESVIOS CONDICIONAIS

Problema: Ler um único número do teclado e imprimi-lo apenas se ele for positivo.

```
program imprime_se_positivo;
var a,b: integer;

begin
      read (a);
      if a > 0 then
            writeln (a); (* so executa se a for positivo *)
end.
```

Comando de desvio condicional

Formato:
if <*expressão booleana*> **then**

DESVIOS CONDICIONAIS: SENÃO

Problema: Ler um único número e imprimi-lo apenas se for positivo. Senão, imprimir "número inválido"

```
program imprime_se_positivo_v2;
var a,b: integer;

begin
      read (a);
      if a > 0 then
            writeln (a)                    (* so executa se a for positivo *)
      else
            writeln ('numero invalido');  (* executa se a for nulo ou negativo *)
end.
```

"**else**" executa se a expressão booleana do "**if**" for falsa

RESUMO DO CAPÍTULO 5 (DUAS ÚLTIMAS AULAS)

Até agora, vimos:

Comandos

- Entrada e saída (*read* e *write*, respectivamente);
- Atribuição (:=);
- Repetição (*while/do*);
- Desvio condicional (*if/then*, ou *if/then/else*);

Expressões

- Aritméticas;
- Booleanas.

FATO: Qualquer algoritmo pode ser programado no computador com os comandos/expressões do quadro acima!!

SIM: QUALQUER!!!