

Estruturas de Repetição e Manipulação de Texto

Leonardo Gresta Paulino Murta
leomurta@ic.uff.br

Aula de hoje...

- Estruturas de repetição
 - *while*
 - *repeat...until*
 - *for*
- String
 - Manipulação de textos

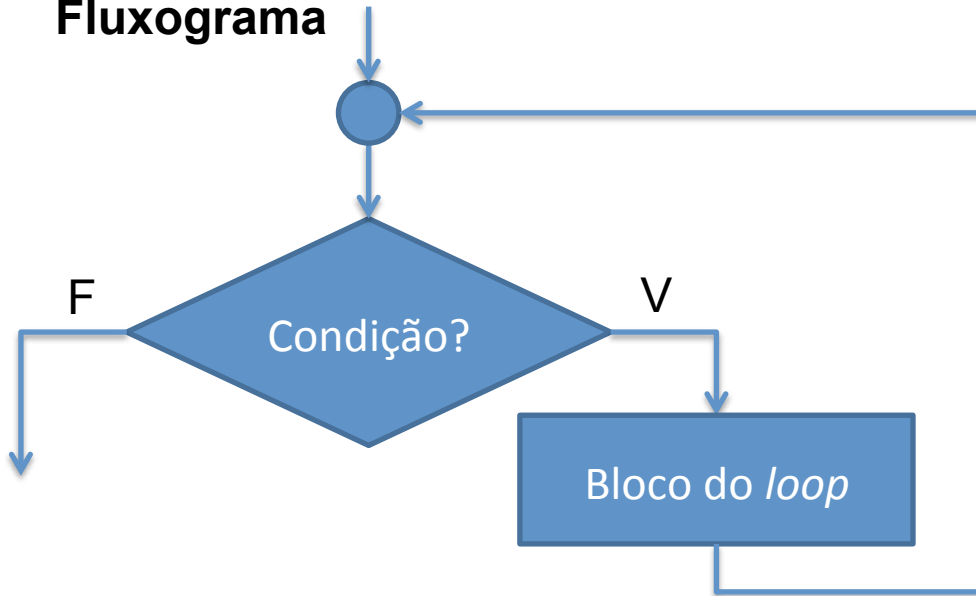
Estruturas de Repetição

- Permitem que um bloco de comandos seja executado diversas vezes
- **Repetição condicional:**
 executa um bloco de código enquanto uma condição lógica for verdadeira
 - *while*
 - *repeat...until*
- **Repetição contável:**
 executa um bloco de código um número predeterminado de vezes
 - *for*



Repetição condicional do tipo *while*

Fluxograma



Pseudocódigo

```

...
Enquanto CONDIÇÃO faça
    INSTRUÇÃO 1
    INSTRUÇÃO 2
    ...
    INSTRUÇÃO N
...
  
```

Repetição condicional do tipo *while*

Pascal

```
...  
while CONDIÇÃO do  
    INSTRUÇÃO;  
...
```

OU

```
...  
while CONDIÇÃO do  
begin  
    INSTRUÇÃO 1;  
    INSTRUÇÃO 2;  
    ...  
    INSTRUÇÃO N;  
end;  
...
```

Repetição condicional do tipo *while*

- Executa o bloco de instruções **enquanto a condição for verdadeira**
- A condição é uma expressão booleana que pode fazer uso de quaisquer operadores
- **Parece com o if**, mas repete diversas vezes enquanto a condição for verdadeira

Exemplo de *while*

- Programa para calcular fatorial de um número:

```

program fatorial;

var n, fat : integer;

begin
  write('Entre com um número inteiro positivo: ');
  readln(n);
  fat := 1;
  while n > 0 do
  begin
    fat := fat * n;
    n := n - 1;
  end;
  writeln('O fatorial desse número é ', fat);
end.

```

Exemplo de *while*

- Qual a saída do programa abaixo?

```

program fatorial;

var i : integer;

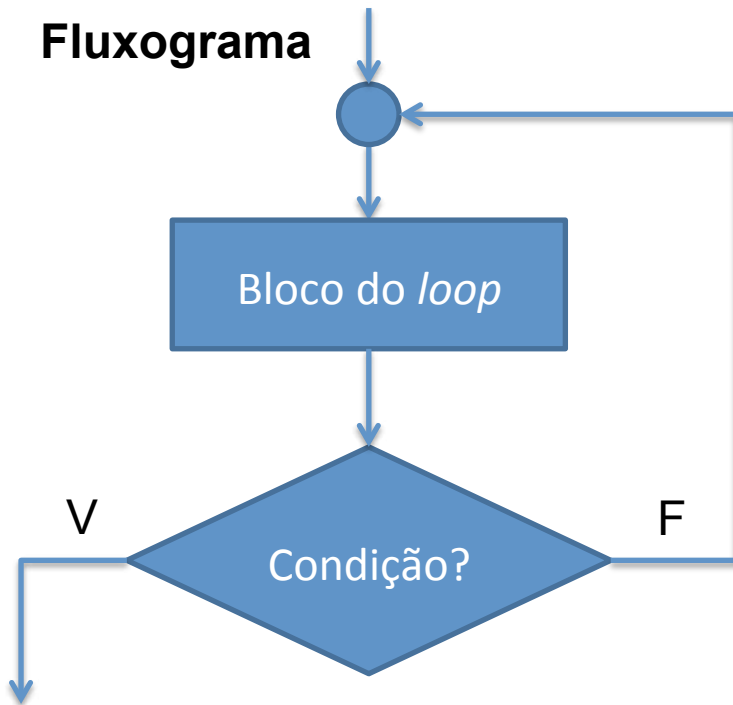
Begin
  i := 0;
  while true do
  begin
    writeln(i);
    i := i + 1;
  end;
end.

```

- Evitem loops infinitos!

Repetição condicional do tipo *repeat...until*

Fluxograma



Pseudocódigo

```

...
Repetir
  INSTRUÇÃO 1
  INSTRUÇÃO 2
  ...
  INSTRUÇÃO N
até CONDIÇÃO
...
  
```

Repetição condicional do tipo *repeat...until*

Pascal

```
...  
repeat  
    INSTRUÇÃO 1;  
    INSTRUÇÃO 2;  
    ...  
    INSTRUÇÃO N;  
until CONDIÇÃO;  
...
```

Repetição condicional do tipo *repeat...until*

- Executa o bloco de instruções **até que a condição seja verdadeira**
 - CUIDADO: a semântica de parada é oposta à do *while* !!!
- **Garante que ocorrerá ao menos uma execução**
 - A verificação da condição é feita depois do bloco de instruções

Exemplo de *repeat...until*

- Programa para calcular fatorial de um número:

```

program fatorial;

var n, fat : integer;

begin
  write('Entre com um número inteiro positivo: ');
  readln(n);
  fat := 1;
  repeat
    fat := fat * n;
    n := n - 1;
  until n < 1;
  writeln('O fatorial desse número é ', fat);
end.

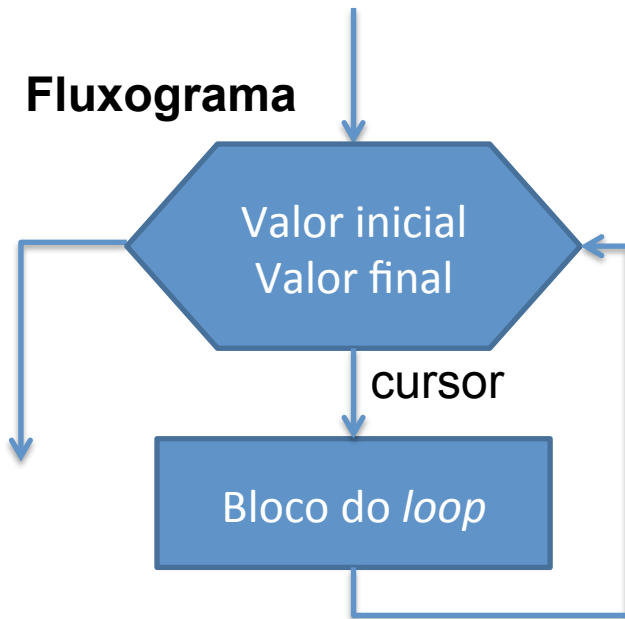
```

Mas então... dá no mesmo?

- Naaaaaaaaaaaaaaaaão!!!
- Reparem que pedimos para o usuário **"Entre com um número inteiro positivo: "**
 - Para esse cenário, ambas as estruturas funcionaram
- O que acontece se pedirmos para o usuário **"Entre com um número inteiro não negativo: "**
 - Qual das duas estruturas resolve o problema corretamente se o usuário entrar com zero?
 - Qual o resultado provido pela outra?
 - Lembrem: fatorial de zero é 1!

Repetição contável do tipo *for*

Fluxograma



Pseudocódigo

```

...
Para CURSOR variando de VALOR INICIAL
a VALOR FINAL
  INSTRUÇÃO 1
  INSTRUÇÃO 2
  ...
  INSTRUÇÃO N
...
  
```

Repetição contável do tipo *for*

Pascal

```

...
for CURSOR := VALOR INICIAL to/downto VALOR FINAL do
    INSTRUÇÃO 1;
...

```

OU

```

...
for CURSOR := VALOR INICIAL to/downto VALOR FINAL do
begin
    INSTRUÇÃO 1;
    INSTRUÇÃO 2;
    ...
    INSTRUÇÃO N;
end;
...

```

Repetição contável do tipo *for*

- Executa o bloco de instruções por um número predeterminado de vezes
- **Cursor**
 - Pode ser do tipo integer ou char
 - Tem seu valor acessado durante a execução
- **Valor inicial e final**
 - Pode ser uma variável
 - Pode ser uma expressão
 - Pode ter valores inteiros ou char, em função do tipo do cursor
 - Pode ser crescente (*to*) ou decrescente (*downto*)

Exemplo de *for*

- Programa para calcular fatorial de um número:

```

program fatorial;

var i, n, fat : integer;

begin
  write('Entre com um número inteiro positivo: ');
  readln(n);
  fat := 1;
  for i := 1 to n do
    fat := fat * i;
  writeln('O fatorial desse número é ', fat);
end.

```

Exemplo de *for*

- Qual a diferença de

```
for i := 1 to n do  
  fat := fat * i;
```

- Para

```
for i := n downto 1 do  
  fat := fat * i;
```

- ?

Manipulação de texto

- Existe uma variedade de funções para manipulação de texto
- Veremos aqui funções padrões, mas há outras funções no módulo módulo **strutils**
 - Necessita “uses strutils;” logo após o “program ...;”
 - <http://www.freepascal.org/docs-html/rtl/strutils/index-5.html>

Manipulação de texto

- `texto1 = texto2`
 - Informa se duas strings são iguais
 - Ex.: `'Flamengo' = 'flamengo' → false`
 - Ex.: `'Flamengo' = 'Flamengo' → true`
- `length(texto : string)`
 - Retorna o tamanho da string
 - Ex.: `length('Flamengo') → 8`
- `texto1 + texto2`
 - Concatena duas strings
 - Ex.: `'Fla' + 'mengo' → 'Flamengo'`
- `texto[posicao : integer]`
 - Retorna o caractere na posição informada
 - Ex.: `s := 'Flamengo';`
 - `s[2] → 'l'`

Manipulação de texto

- `texto1 < texto2` (valem os demais operadores relacionais)
 - Retorna true se texto1 for lexicamente menor que texto2
 - `'Fla' < 'Flu' → true`
- `pos(subtexto : string, texto : string)`
 - Busca pela primeira ocorrência de subtexto em texto
 - Retorna zero caso não encontre nenhuma ocorrência
 - Ex.: `pos('Flu', 'Fla x Flu') → 7`
 - Ex.: `pos('Bota', 'Fla x Flu') → 0`

Manipulação de texto

- `copy(texto : string, inicio: integer, tamanho : integer)`
 - Retorna a substring do texto com início e tamanho indicados
 - Ex.: `copy('Flamengo', 4, 3) → 'men'`
- `delete(texto : string, inicio: integer, tamanho : integer)`
 - Remove a substring do texto com início e tamanho indicados
 - Ex.: `s := 'Flamengo';`
 - `delete(s, 1, 3) → s = 'mengo'`
- `insert(subtexto : string, texto : string, posicao: integer)`
 - Insere a substring no texto em uma determinada posição
 - Ex.: `s := 'Flago';`
 - `insert('men', s, 4) → s = 'Flamengo'`

Manipulação de texto

- lowercase(texto : string)
 - Retorna a string em minúsculas
 - Ex.: lowercase('Flamengo') → 'flamengo'
- upcase(texto : string)
 - Retorna a string em maiúsculas
 - Ex.: upcase('Flamengo') → 'FLAMENGO'
- str(numero : integer, texto : string)
 - Converte um número em texto
 - str(123, s) → s = '123'
- val(texto : string, numero : integer/real, erro : integer)
 - Converte um texto em número
 - Caso algum caractere impeça a conversão, a posição do caractere é indicada como código do erro
 - val('123', n, c) → n = 123, c = 0
 - val('12x3', n, c) → n = 0, c = 3

Exemplo

- Programa para gerar a citação a partir de um nome
 - Ex.: Leonardo Gresta Paulino Murta → MURTA, L. G. P.

```
program teste;
```

```
var
```

```
  nome, iniciais : string;
```

```
  posicao : integer;
```

```
begin
```

```
  iniciais := '';
```

```
  write('Entre com um nome completo: ');
```

```
  readln(nome);
```





Exemplo

```

posicao := pos(' ', nome);
while posicao <> 0 do
begin
    iniciais := iniciais + copy(nome, 1, 1) + '. ';
    delete(nome, 1, posicao);
    posicao := pos(' ', nome);
end;
writeln(uppercase(nome) + ', ' + uppercase(iniciais));
end.

```

Exercício

- Faça um programa para montar a tabela de multiplicação de números de 1 a 10 (ex.: $1 \times 1 = 1$, $1 \times 2 = 2$, etc.)
- Faça um programa para determinar o número de dígitos de um número informado

Exercício

- Faça um programa para calcular a série de Fibonacci para um número informado pelo usuário, sendo $F(0) = 0$, $F(1) = 1$ e $F(n) = F(n-1) + F(n-2)$
 - Por exemplo, caso o usuário informe o número 9, o resultado seria: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Exercício

- Faça um programa para listar todos os divisores de um número ou dizer que o número é primo caso não existam divisores
 - Ao final, verifique se o usuário deseja analisar outro número

Exercício

- Faça um programa que calcule o retorno de um investimento financeiro fazendo as contas mês a mês, sem usar a fórmula de juros compostos
 - O usuário deve informar quanto será investido por mês e qual será a taxa de juros mensal
 - O programa deve informar o saldo do investimento após um ano (soma das aplicações mês a mês considerando os juros compostos), e perguntar ao usuário se ele deseja que seja calculado o ano seguinte, sucessivamente
 - Por exemplo, caso o usuário deseje investir R\$ 100,00 por mês, e tenha uma taxa de juros de 1% ao mês, o programa forneceria a seguinte saída:

```
Saldo do investimento após 1 ano: 1280.9328043328942
```

```
Deseja processar mais um ano? (S/N)
```

Exercício

- Faça um programa para justificar um texto com um número de colunas informado pelo usuário
- Por exemplo, para o texto “Este é um exemplo de texto que vamos justificar usando o nosso programa.” quando justificado em 18 colunas, teríamos:

Este é um exemplo
de texto que vamos
justificar usando
o nosso programa.

- Dica: veja as funções `posex`, `rpos` e `rposex` do módulo `strutils`

Estruturas de Repetição e Manipulação de Texto

Leonardo Gresta Paulino Murta
leomurta@ic.uff.br