

Elementos básicos das linguagens de programação

Objetivos:

- Apresentar os últimos elementos básicos das linguagens de programação

Elementos básicos já estudados

- Entrada (read, readln)
- Saída (write, writeln)
- Atribuição (:=)
- Comando de repetição (while ... do)

- Expressões aritméticas
- Expressões booleanas

O que falta?

- Possibilitar desvios no fluxo de execução
- Eles podem ser condicionais ou incondicionais

O que falta?

- Possibilitar desvios no código
- Eles podem ser condicionais ou incondicionais
- Desvios incondicionais (goto)
- Desvios condicionais (if ... then ... else)

Desvios incondicionais

- É uma das estruturas básicas
- Permite que o fluxo de execução seja alterado para qualquer outro trecho do programa
- É implementado em Pascal através do comando `goto`

Desvios incondicionais

- Exemplo

Begin

```
10:  Read (n);  
     Cont := Cont + 1;  
     Goto 10;
```

End.

Desvios incondicionais

- Problema: este programa nunca termina...

Begin

```
10: Read (n);  
    Cont := Cont + 1;  
    Goto 10;
```

End.

Desvios condicionais

- Permite que o fluxo de execução seja alterado de maneira controlada
- É implementado em Pascal através do comando `if`

Desvios condicionais: primeira forma

- Exemplo: queremos ler um número e imprimí-lo na tela apenas se ele for positivo

```
Read (n);  
If n > 0 then  
    WriteLn (n);
```

Desvios condicionais: primeira forma

- Exemplo: queremos ler um número e imprimí-lo na tela apenas se ele for positivo

```
Read (n);  
If n > 0 then  
    WriteLn (n);
```

- Se n for negativo ou nulo, ele não será impresso

Desvios condicionais: primeira forma

- O controle (ou condição) relativo ao comando if é uma expressão booleana

```
Read (n);  
If n > 0 then  
    WriteIn (n);
```

Desvios condicionais: primeira forma

- O controle (ou condição) relativo ao comando if é uma expressão booleana

```
If < expressão booleana > then  
    Writeln (n);
```

Expressões booleanas

- Uma expressão booleana sempre resulta em um valor lógico que só pode ser **verdadeiro** ou **falso**

Desvios condicionais: segunda forma

- Exemplo: queremos ler um número.
- Se ele for positivo deve ser impresso na tela mas caso não seja queremos imprimir uma mensagem de erro

Desvios condicionais: segunda forma

```
Read (n);  
If n > 0 then  
    Writeln (n)  
Else  
    Writeln ('numero invalido');
```

Exemplos de problemas

- Determinar se um número é par

Exemplos de problemas

- Determinar se um número é ímpar

Exemplos de problemas

- Determinar se dois números são pares

Exemplos de problemas

- Determinar se, dados dois números, se pelo menos um deles é par

Exemplos de problemas

- Determinar se, dados dois números, se nenhum deles é par

Exemplos de problemas

- Dados três números reais a , b e c , caracterizando uma equação do segundo grau da forma $ax^2 + bx + c = 0$, decidir se o discriminante $(b^2 - 4ac)$ é positivo ou nulo

Exemplos de problemas

- Dados três números reais a , b e c , caracterizando uma equação do segundo grau da forma $ax^2 + bx + c = 0$, e outros dois números reais x_1 e x_2 , decidir se eles são as duas soluções reais da equação

Exemplos de problemas

- Dados três números reais a , b e c , caracterizando uma equação do segundo grau da forma $ax^2 + bx + c = 0$, e outros dois números reais x_1 e x_2 , decidir se eles são as duas soluções reais da equação. Caso não existam raízes reais, imprimir mensagem avisando ao usuário.

Desvios incondicionais

Begin

Cont:= 0;

10: Read (n);

If n = 0 then

Goto 20;

Cont := Cont + 1;

Goto 10;

20: Writeln (cont);

End.

Estes códigos fazem a mesma coisa

```
Begin
  Cont := 0;
  Read (n);
  While n <> 0 do
    Begin
      Cont := Cont + 1;
      Read (n);
    End;
  Writeln (Cont);
End.
```

```
Begin
  Cont:= 0;
10:  Read (n);
     If n = 0 then
       Goto 20;
     Cont := Cont + 1;
     Goto 10;
20:  Writeln (cont);
End.
```

Programação estruturada

- Paradigma de linguagem de programação na qual não se usa goto
- Todos os programas podem ser escritos usando-se atribuição, desvios condicionais, repetição, além de entrada e saída
- **Neste curso o uso de goto é proibido!**

Programação estruturada

- Entrada (read, readln)
- Saída (write, writeln)
- Atribuição (:=)
- Comando de repetição (while ... do)
- Desvios condicionais (if ... then ... else)

- Expressões aritméticas
- Expressões booleanas