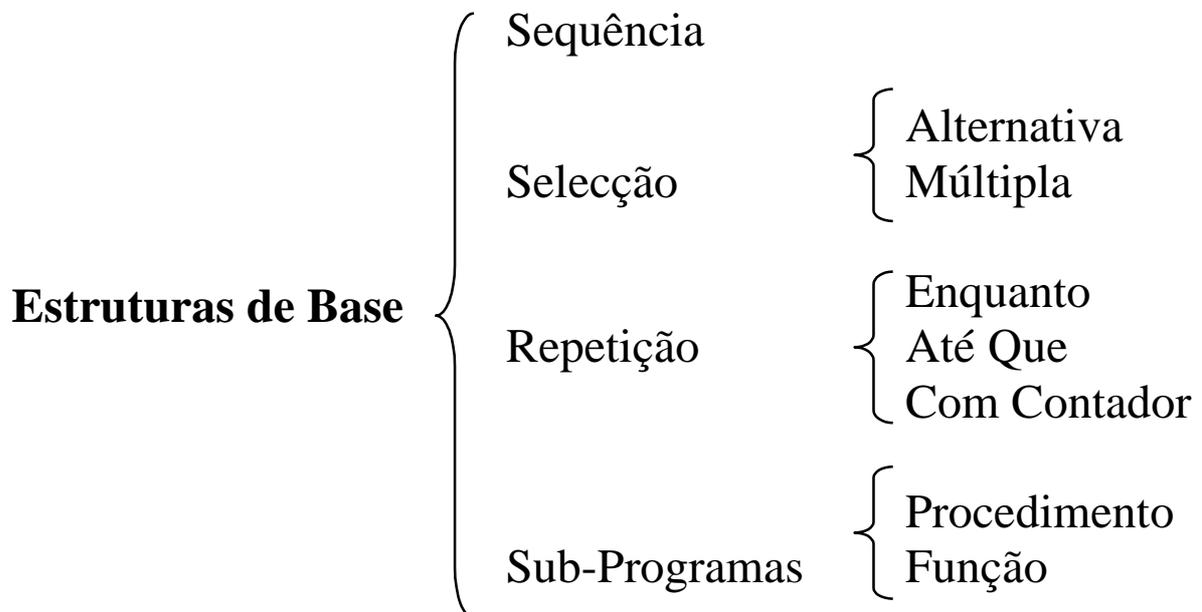


Capítulo III : A Linguagem Pascal – Estruturas de Controlo



1. A Sequência

- **As Instruções são obedecidas sequencialmente, separadas por ;**

Exemplo:

```
...  
Instrução1;  
  
Instrução2; Instrução3; Instrução4;  
Instrução5; ; Instrução7;  
...
```

↑
Instrução Nula

A Instrução Composta:



Exemplo:

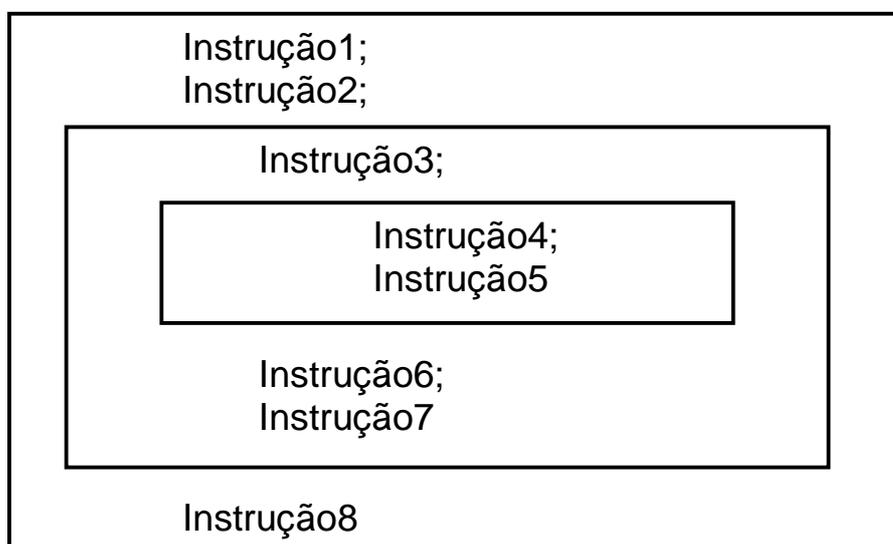
```

begin Instrução1;
      Instrução2;
      begin Instrução3;
            begin Instrução4;
                  Instrução5
            end;
            Instrução6;
            Instrução7
      end;
      Instrução8
end;

```

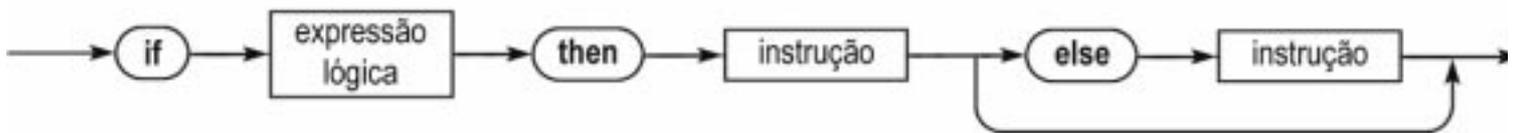
- O par begin ... end funciona como Parêntesis de Instruções.
- A “Indentação” torna o programa mais legível.

Diagrama de Estrutura:



2. A Selecção

2.1. A Alternativa

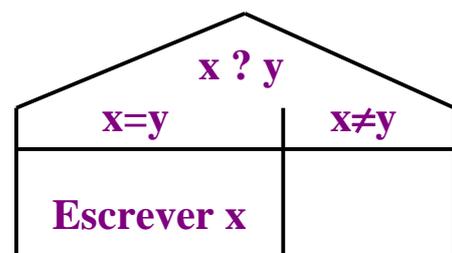


- A Expressão Lógica é calculada e, consoante o seu valor, uma das Instruções é executada.
- A Instrução if tem duas formas possíveis.

Diagramas de Estrutura

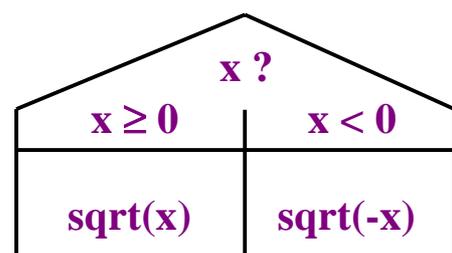
Exemplo da primeira forma:

```
if    x = y
then write(x);
```



Exemplo da segunda forma:

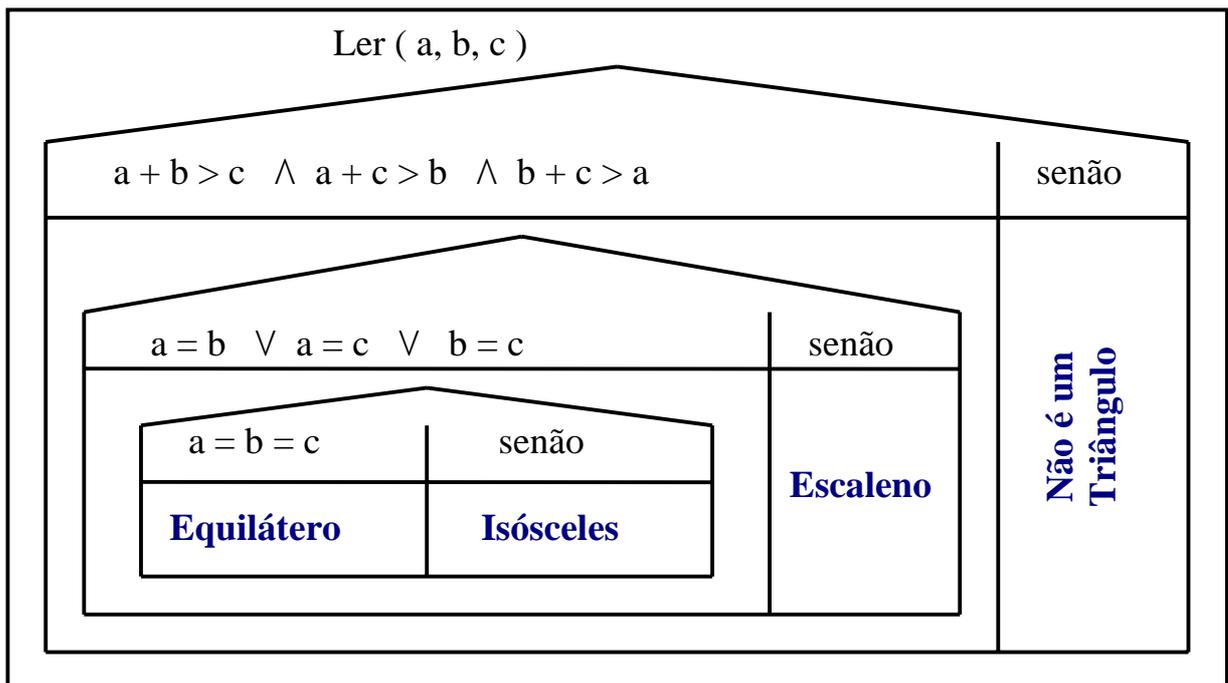
```
if    x >= 0
then raiz:= sqrt(x)
else  raiz:= sqrt(-x);
```



Problema:

Dados três números inteiros e positivos, decidir se poderão ser considerados como os lados de um triângulo e, em caso afirmativo, indicar o seu tipo.

Diagrama de Estrutura:



Programa:

```

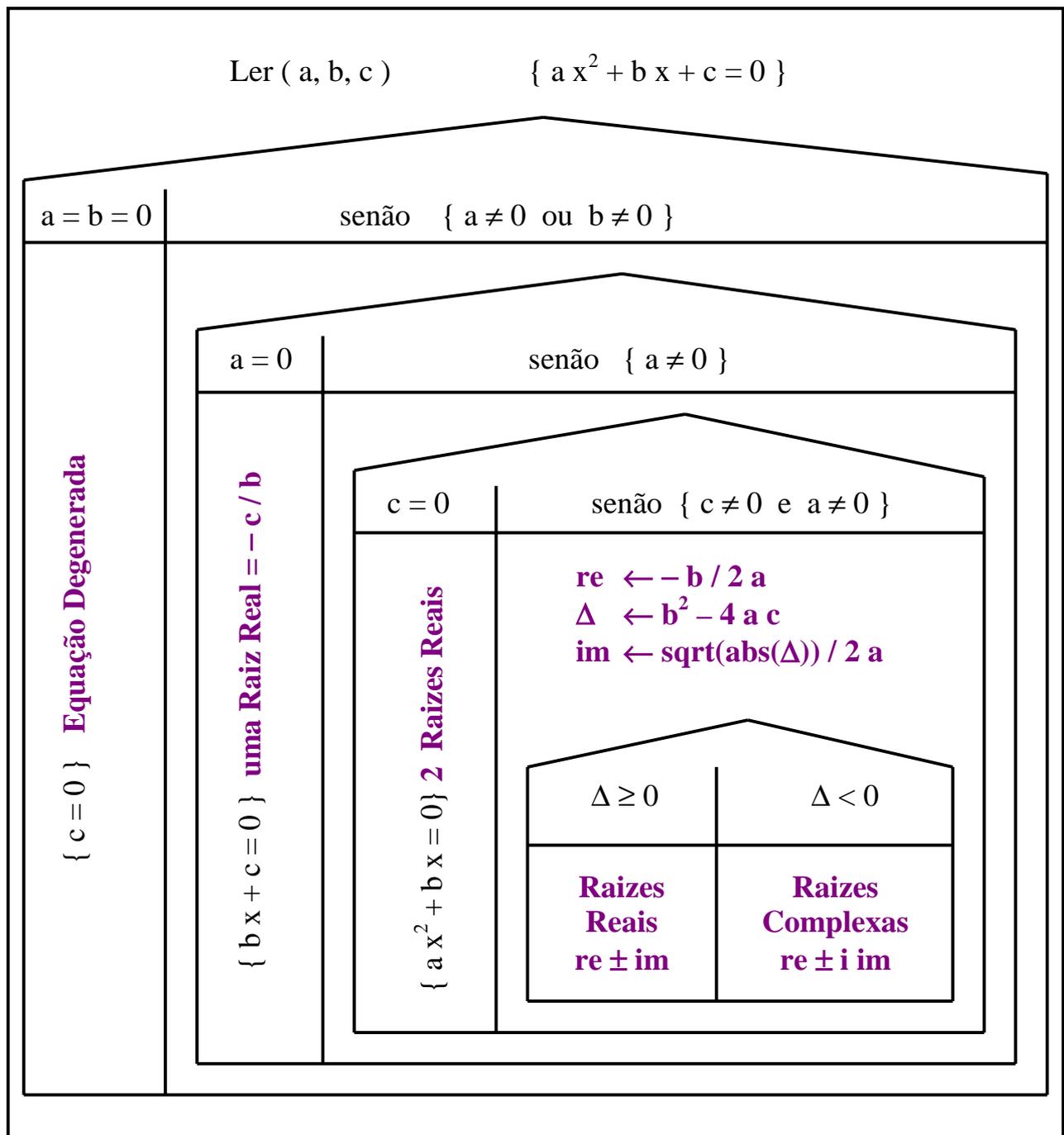
program triangulos (input, output);
var a, b, c : integer;
begin read(a, b, c);
      if (a<b+c) and (b<a+c) and (c<a+b)
      then if (a=b) or (a=c) or (b=c)
           then if (a=b) and (b=c)
                then writeln('Triangulo Equilatero')
                else writeln('Triangulo Isosceles')
           else writeln('Triangulo Escaleno')
      else writeln('Nao pode ser um Triangulo')
end.
  
```

Problema:

Discutir e calcular as raízes da Equação Quadrática, com coeficientes inteiros,

$$a x^2 + b x + c = 0$$

Diagrama de Estrutura:



$$x = \underbrace{-\frac{b}{2a}}_{re} \pm \underbrace{\frac{\sqrt{b^2 - 4ac}}{2a}}_{im}$$

Programa:

```

program equacao_quadratica (input, output);
var a, b, c : integer;
    delta, re, im : real;

begin write('Escreva os valores dos 3 coeficientes inteiros:');
    readln(a, b, c);
    if (a = 0) and (b = 0)
    then (* equacao: c = 0 *)
        writeln('Equacao Degenerada')
    else if (a = 0)
    then (* equacao: bx + c = 0 *)
        writeln('Uma Raiz Simples =', -c/b)
    else (* a ≠ 0*)
        if (c = 0)
        then (* equacao: ax2 + bx = 0 *)
            writeln('Raizes Reais :', -b/a, 'e 0')
        else begin (* Caso Geral *)
            re := -b/(2*a);
            delta := sqr(b) - 4*a*c;
            im := sqrt(abs(delta))/(2*a);
            if delta >= 0
            then writeln('Raizes Reais :',
                re+im, ' e ', re-im)
            else writeln('Raizes Complexas :',
                re, '+i', im, ' e ', re, '-i', im)
        end
    end.

```

Exercício: Aperfeiçoar o Formato dos Resultados.

Nota:

De modo análogo ao caso das expressões,

$$\begin{array}{l}
 a * b + c \quad \nearrow \quad (a * b) + c \quad \text{(pelas regras de Prioridade)} \\
 \quad \quad \quad \searrow \quad a * (b + c) \quad \text{(pela utilização de Parêntesis)}
 \end{array}$$

existe uma possível ambiguidade na utilização encadeada de dois **ifs**:

if cond1 then if cond2 then inst1 else inst2;

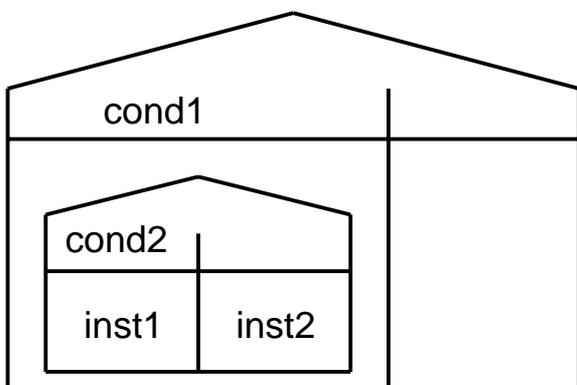
A qual dos dois **ifs** pertence o **else**?

- **Por convenção, cada else corresponde ao último if indicado.**

**Pela Regra de Prioridade
convencionada:**

```

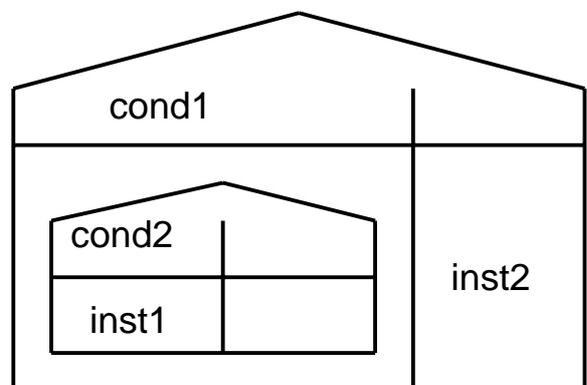
If cond1
then if cond2
      then inst1
      else inst2;
  
```



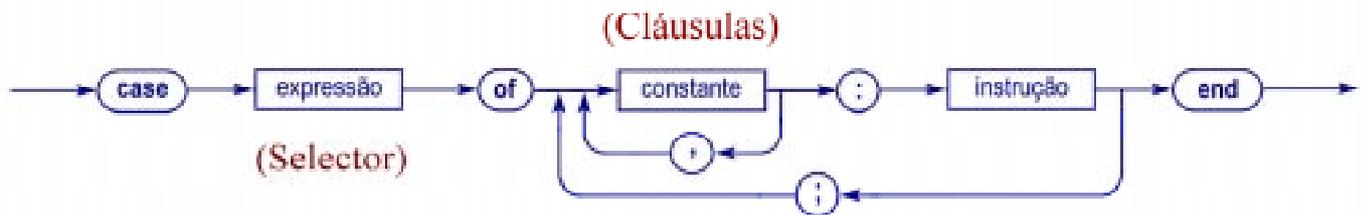
**Pela Utilização da Composição
de Instruções:**

```

If cond1
then begin if cond2
      then inst1
      end
else inst2;
  
```



2.2. A Selecção Múltipla



- A Expressão (Selector) pode ser de qualquer Tipo Escalar, mas não Real;
- O Selector e as Cláusulas são do mesmo Tipo;
- O Selector só pode tomar os valores estipulados nas Cláusulas.

Exemplo:

```

type tipodeveiculo = (bicicleta, motorizada, automovel,
                        autocomreb, autocarro, camioneta);
var   veiculo : tipodeveiculo;
...
case veiculo of
    bicicleta, motorizada : writeln('200$00');
    automovel : writeln('400$00');
    autocomreb : writeln('600$00');
    autocarro, camioneta : writeln('1000$00')
end;
    
```

veiculo?			
bicicleta ou motorizada	automovel	autocomreb	autocarro ou camioneta
200\$00	400\$00	600\$00	1000\$00

Exemplo: Se o dia 1 de Novembro de 1999 foi uma Segunda-Feira, como calcular os restantes?

dia mod 7 ?						
0	1	2	3	4	5	6
domingo	segunda	terça	quarta	quinta	sexta	sabado

Uma solução:

```

...
case dia mod 7 of
  0 : writeln('Domingo');
  1 : writeln('Segunda Feira');
  2 : writeln('Terca Feira');
  3 : writeln('Quarta Feira');
  4 : writeln('Quinta Feira');
  5 : writeln('Sexta Feira');
  6 : writeln('Sabado')
end;

```

Mas se o utilizador escrever um valor de dia \notin [1, 30]?

```

...
if (dia >= 1) and (dia <= 30)
then case dia mod 7 of
  0 : writeln('Domingo');
  1 : writeln('Segunda Feira');
  2 : writeln('Terca Feira');
  3 : writeln('Quarta Feira');
  4 : writeln('Quinta Feira');
  5 : writeln('Sexta Feira');
  6 : writeln('Sabado')
end
else writeln(dia, 'Não é valor de dia de Novembro');

```

- Quando o Selector toma um valor não estipulado nas Cláusulas ocorre um Erro de Execução.

Exemplo: (ver Cap.II, pág. 8)

Que fazer, quando o operador $\notin \{ '+', '-', '*', '/' \}$?

Solução em Pascal “standard”:

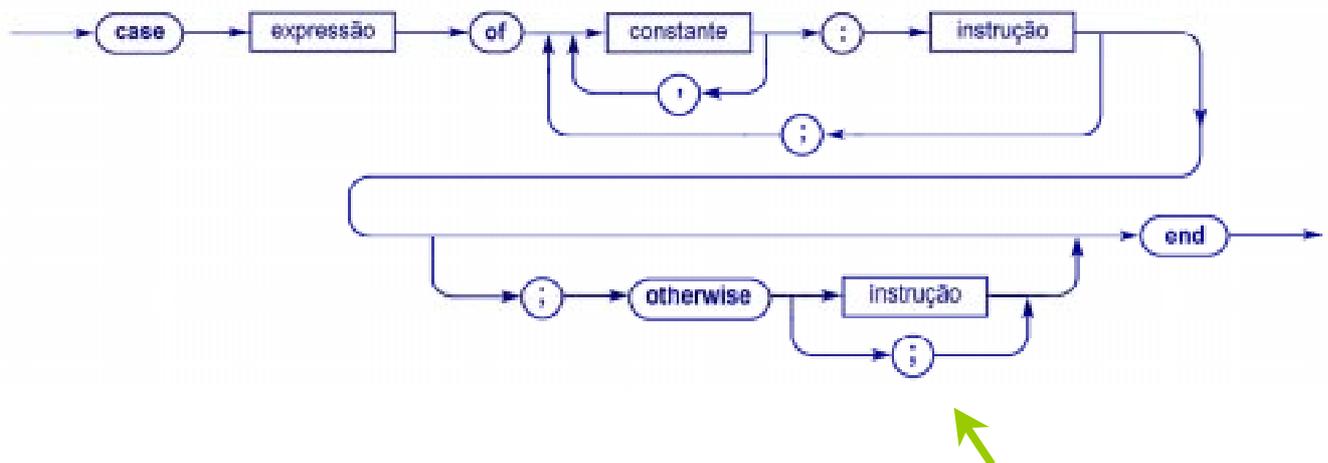
```
program calculadora2(input, output);
var  oper: char;
     x, y, resultado: real;

begin  read(x, oper, y);

      if (oper = '+') or (oper = '-') or (oper = '*') or (oper = '/')
      then begin case oper of
                '+': resultado := x+y;
                '-': resultado := x-y;
                '*': resultado := x*y;
                '/': resultado := x/y
              end;
          writeln(x, oper, y, '=', resultado)
        end
      else writeln('ERRO: Operador nao Previsto')
    end.
```

- Contudo, na maioria das implementações actuais do Compilador Pascal (p.ex. no TurboPascal), existe a opção otherwise (ou else).

Forma não “standard” da Seleção Múltipla:



Exemplo:

```

case oper of
  '+': resultado := x+y;
  '-': resultado := x-y;
  '*': resultado := x*y;
  '/': resultado := x/y;
otherwise writeln('ERRO')
end;
  
```

Ou então:

```

case oper of
  '+': resultado := x+y;
  '-': resultado := x-y;
  '*': resultado := x*y;
  '/': resultado := x/y;
otherwise ;
end;
  
```

Mas, para este exemplo, esta solução levanta outro problema. Qual? Como resolvê-lo?

Solução em Pascal “não-standard”:

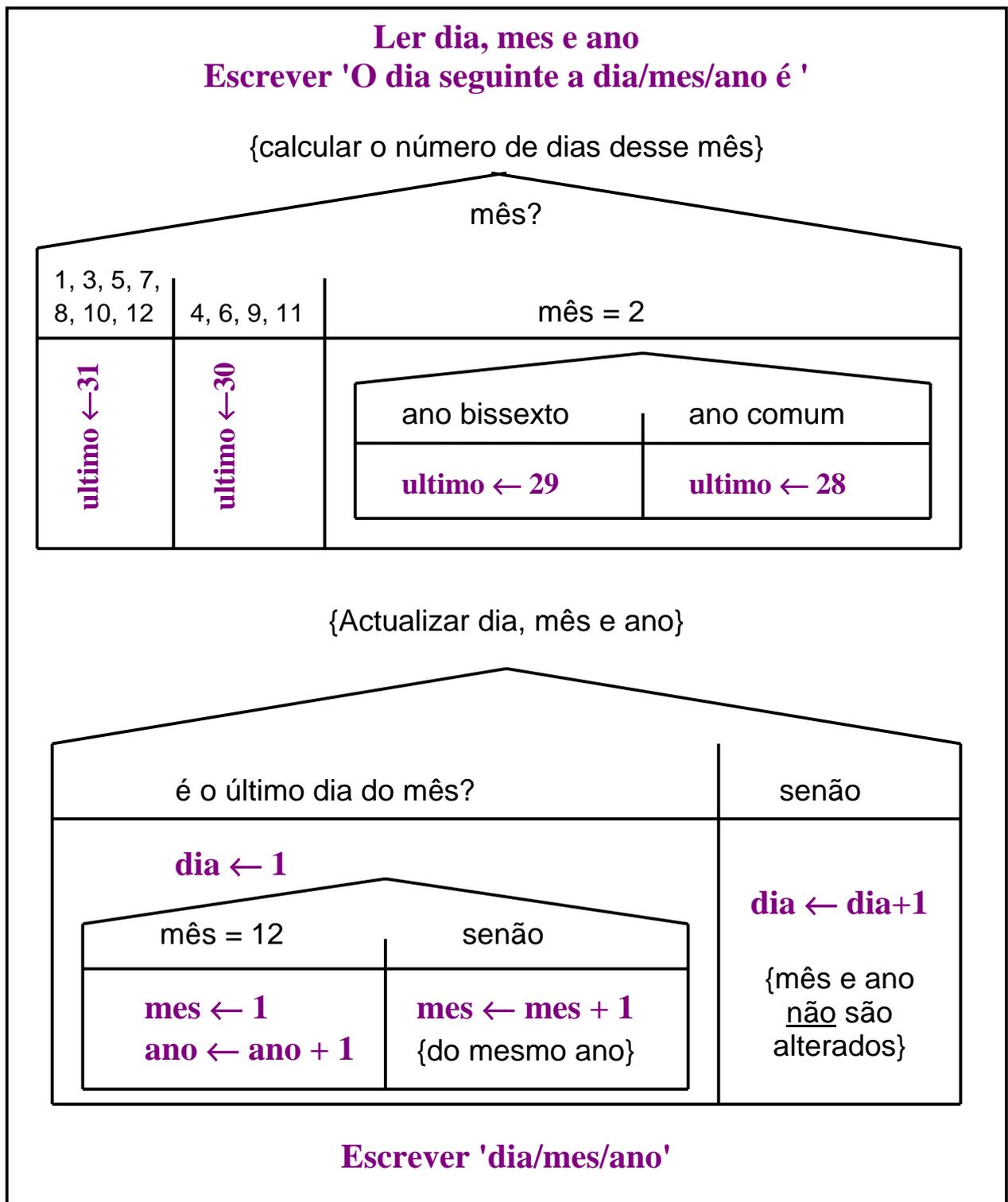
```
program calculadora3(input, output);  
var oper: char;  
    x, y, resultado: real;  
    certo: boolean;  
  
begin writeln('Qual a Operacao?');  
    read(x, oper, y);  
  
    certo:= true;  
    case oper of  
        '+': resultado := x+y;  
        '-': resultado := x-y;  
        '*': resultado := x*y;  
        '/': resultado := x/y;  
    otherwise certo:= false  
    end;  
  
    if certo  
    then writeln(x, oper, y, '=', resultado)  
    else writeln('ERRO: Operador nao Previsto')  
end.
```

Exercício: Incluir um teste capaz de evitar o caso do erro provocado por uma tentativa de divisão por zero.

Problema: Calcular a data do dia seguinte a um dado dia.

Exemplo:

O dia seguinte a 31/12/1999 é 1/1/2000.



Programa:

```
program dia_seguinte(input, output);
var dia: 1..31;
    mes: 1..12;
    ano: 1582..maxint;
    ultimo: 28..31;
    bissexto: boolean;

begin  writeln('Indique o dia, mes e ano da presente data');
        read(dia, mes, ano);
        write('Dia seguinte a ', dia:3, '/', mes:2, '/', ano:4, ' =');

        bissexto:= (ano mod 4 = 0) and (ano mod 100 <> 0)
                    or (ano mod 400 = 0);
        (*Calculo do numero de dias do mes*)
        case mes of
            1, 3, 5, 7, 8, 10, 12 : ultimo := 31;
            4, 6, 9, 11 : ultimo := 30;
            2 : if bissexto
                then ultimo := 29
                else ultimo := 28
        end;

        (*Actualizacao de dia, mes e ano*)
        if dia = ultimo
        then begin dia := 1;
                if mes = 12
                then begin mes := 1;
                            ano := ano+1
                end
                else mes := mes+1
        end
        else dia := dia+1;

        writeln(dia:3, '/', mes:2, '/', ano:4)
end.
```