UNISINOS - UNIVERSIDADE DO VALE DO RIO DOS SINOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS (C6/6) – Curso: Informática

PROGRAMAÇÃO I – AULA 03

Disciplina: Linguagem de Programação PASCAL
Professor responsável: ***Fernando Santos Osório***
Semestre: 2001/2
Horário: 63

E-mail: ***osorio@exatas.unisinos.br***
Web:
http://inf.unisinos.br/~osorio/prog1.html
Xerox : *Pasta 54* (Xerox do C6/6

1. LINGUAGEM PASCAL – Tipos de Dados

A Linguagem Pascal possui três tipos de dados principais...
NUMÉRICOS (inteiros e reais), **CARACTERES** e **LÓGICOS**

* Tipos de dados: Lógico: (Boolean / Binário)

Os dados do tipo LÓGICO podem armazenar *dois valores* apenas: verdadeiro(***True***) ou Falso(***False***). Este tipo de dados é usado em expressões lógicas, onde servem para testar ou indicar a satisfação ou não de alguma condição.

Tipo BOOLEAN: Serve para armazenar um valor lógico. Exemplos:

```
VAR
  Desempregado : Boolean;
  Falido : Boolean;
  Aprovado, Reprovado : Boolean;
BEGIN
   Desempregado := False;          Aprovado := True;
   Falido := True;                 Reprovado := False;
{ Posso trabalhar com o tipo Char de forma "equivalente" ao tipo Boolean }
{ Por exemplo:  Teve_Aprovacao := 'S';  <= Mas atenção: char é diferente de boolean !!}
```

* Expressões Condicionais:

São expressões cujo resultado é um valor do tipo Boolean (True ou False). As expressões condicionais utilizam *operadores* do tipo *relacional* ou do tipo *lógico*, podendo fazer uso de constantes e variáveis de qualquer tipo associadas a estes operadores.

° *Operadores Relacionais:*

=	Igual à	VL := A = B;
<>	Diferente de	VL := A <> B;
>	Maior que	VL := A > B;
<	Menor que	VL := A < B;
>=	Maior ou igual à	VL := A >= B;
<=	Menor ou igual à	VL := A <= B;

Obs.: VL é do tipo Boolean, A e B podem ser de qualquer tipo.

° *Operadores Lógicos*:

AND	"E" Lógico	VL := A And B;
OR	"OU" Lógico	VL := A Or B;
XOR	"OU Exclusivo" Lógico	VL := A Xor B;
NOT	"Não" (Negação Lógica)	VL := Not A;

Os operadores lógicos realizam operações da lógica booleana sobre os seus operandos, onde o resultado obtido será de acordo com as tabelas verdade de cada operando (descritas abaixo). Os operandos usados com operadores lógicos devem ser obrigatoriamente do tipo lógico (boolean).

AND	**OR**	**NOT**
True AND True => True	True OR True => True	NOT True => False
True AND False => False	True OR False => True	
False AND True => False	False OR True => True	NOT False =>True
False AND False => False	False OR False => False	

XOR
True XOR True => False
True XOR False => True
False XOR True => True
False XOR False => False

- Portanto as expressões condicionais combinam operadores lógicos, operadores relacionais e dados do tipo lógico, numérico ou caracter. Exemplos:

```
VL :=  Not ( ( A And B ) Or C );
VL :=  X  <>  Y;
VL :=  ( ( X + 3.7 ) / 2.0  )  >  6.0;            { Pode incluir expressões aritméticas... }
VL :=  ( X >= 0.0 )  And  ( X <= 10.0 );          { Pode ter operadores lógicos e relacionais... }
VL :=  Not ( ( X < 0.0 ) Or  ( X > 10.0 ) );
VL :=  CHR ( ( TRUNC ( SQRT(X) ) )  MOD 256  ) > 'A';       { Tudo misturado... }
```

2. COMANDO IF - Desvio Condicional

O comando IF permite que se realize um "desvio condicional" durante a execução de um programa. Com este comando podemos desviar a seqüência de execução para que, caso uma condição seja satisfeita, um certo número de comandos seja executado.

* SYNTAXE:

IF <*Condição*> | **IF** <*Condição*>
THEN <*Comando_Se_Condição_True*> **;** | **THEN** <*Comando_Se_Condição_True*>
| [**ELSE** <*Comando_Se_Condição_False*>] **;**

<Condição> = Expressão condicional cujo resultado é um valor lógico (True ou False).
O **Then** é *obrigatório*, porém o **Else** é *opcional* no comando If, que termina com o ';'.

Exemplos:

```
IF Aprovado
THEN WriteLn ('Parabéns');

IF  X > 6.0
THEN WriteLn ('X é maior que 6')
ELSE  WriteLn ('X é menor ou igual à 6');
```

Observações MUITO Importantes:

- ***Uso do ';':*** O comando IF/THEN/ELSE só aceita o ';' ao final de todo o comando, ou seja, quando tem o ELSE, a parte do THEN não terá o ';' logo antes deste ELSE !

- ***Begin / End:*** Podemos "desdobrar" o comando do THEN ou do ELSE em mais de um comando através do uso do BEGIN / END. Exemplos:

```
IF <expressão>
THEN  BEGIN
          <comando1>;
          <comando2>;
          <comando3>;
          ...
        END                    { Não tem o ';' pois segue o Else logo depois }
ELSE   <comando>;

IF  X >= 0
THEN BEGIN
          Write ('X é maior que zero ');
          WriteLn ('ou igual à zero');
       END
ELSE  WriteLn ('X é menor que zero');
```

  ⇨ *Portanto sempre que tiver mais de um comando associado ao THEN ou ao ELSE sou OBRIGADO a colocar um BEGIN/END.*

- ***Atenção para este ERRO CLÁSSICO de programação em Pascal:***

  ⇨ Colocar mais de um comando no Then sem o Begin/End. O primeiro comando será considerado como fazendo parte do If/Then e o segundo já estará **FORA** do If/Then.

```
IF Reprovado
THEN  WriteLn ('Sua nota foi inferior ao limite de aprovação...');
        WriteLn ('Você terá que repetir a disciplina!');
```

  ⇨ O segundo WriteLn SEMPRE será executado, independentemente do valor da variável Reprovado estar indicando que o aluno foi realmente reprovado ou não. Portanto a mensagem "você vai ter que repetir a disciplina" será exibida na tela mesmo quando o aluno não foi reprovado.

- ***Como evitar erros:***

- Usar sempre que possível o Begin/End junto ao Then ou Else;

- Alinhar os comandos no editor de textos para facilitar a visualização da lógica do programa. O alinhamento (*indentação*) deve ser aplicado tanto para o IF/THEN/ELSE quanto para o BEGIN/END;

- Quando estiver editando um programa no microcomputador, ao digitar o BEGIN escreva imediatamente após o END, preenchendo depois o seu conteúdo. Assim você não vai se esquecer de fechar o BEGIN que começou em uma parte anterior do programa;

- Para localizar um erro como o descrito acima, use a execução passo-à-passo (Step/Trace) e observe o conteúdo e resultado das variáveis e expressões (Evaluate-Modify).

- ***IF's Aninhados:***

- Um IF pode ser usado como sendo o comando de um Then ou de um Else de um outro comando IF. Quando os IF's são encadeados desta maneira, damos o nome de IFs Aninhados para este tipo de estrutura. Exemplos:

```
IF <expressão>
THEN  IF <expressão>
        THEN  IF <expressão>
                THEN  IF <expressão>
                        THEN <Comando>
                        ELSE <Comando> ;

IF  Salario < 120.00
THEN IF Salario < 0.0
        THEN WriteLn ('você paga para trabalhar!')
       ELSE  WriteLn ('Salário muito baixo!')
ELSE  IF Salario > 12000.00
        THEN WriteLn ('Salário muito alto !')
        ELSE  WriteLn ('Vamos discutir o assunto...');
```

* <u>Resumo sobre o comando IF</u>:

```
IF <expressão_condicional>

THEN  [ BEGIN <comando>; <comando>; ... ]
          <comando>
         [ [;] END ]

[ ELSE [ BEGIN <comando>; <comando>; ...]
          <comando>
          [ [;] END ]
] ;
```

EXERCÍCIOS – AULA 03

1. Supondo as seguintes declarações de variáveis e atribuições, determine o resultado as expressões que estão indicadas logo abaixo.

```
PROGRAM Exercicios_Aula03;
VAR
   A, B, C : INTEGER;
   X, Y, Media  : REAL;
   Letra : CHAR;
   Resultado : BOOLEAN;
   VarLog : BOOLEAN;
BEGIN
    A := 3;
    B := 6;
    C := 10;
    X := 6.8;
    Y := 5.3;
    Letra := 'S';
```

1.1. Resultado := (X > 6.0) And (Y > 6.0)
Resultado = _____

1.2. Resultado := (X > 5.0) Or (Y > 9.0)
Resultado = _____

1.3. Resultado := Not (X > 6.0) And Not (Y > 6.0)
Resultado = _____

1.4. Resultado := Not (X > 6.0) Or (Y > 5.0)
Resultado = _____

1.5. Resultado := Not ((X > 6.0) Or (Y > 5.0))
Resultado = _____

1.6. Resultado := ((A + B + C) DIV 3) > (SQR (X) + C / 3 + C MOD 3)
Resultado = _____

1.7. Resultado := (Letra ='S') And (Letra='s') ;
Resultado = _____

1.8. Resultado := Not (Letra = 's ') ;
Resultado = _____

1.9. Resultado := (Letra in ['A'..'Z']) And Not (Letra in ['a'..'z']) ;
Resultado = _____ { Questão desafio }

1.10. Resultado := VarLog Xor (False Xor VarLog); { O valor de VarLog não é conhecido }
Resultado = _____ { Questão desafio }

2. Analise os trechos de código fornecidos abaixo, considerando as mesmas declarações e atribuições de variáveis dadas no exercício 1, e determine o valor final (ao final da execução) das variáveis pedidas.

2.1.
```
IF  ( Letra = 's' )
    THEN  Media := ( X + Y ) / 2
    ELSE   Media := ( A + B ) / 2;
```
Media: _____

2.2.
```
IF  ( Letra <> 'N' ) And ( Letra <> 'n' )
    THEN  Resultado := True
    ELSE   Resultado := False;
```
Resultado: _____

2.3.
```
Media := ( X + Y ) / 2.0 ;
IF  Not  ( Media  <= 6.0  )
THEN  Resultado := True
ELSE   Resultado :=  ( ( X + Y * 2 ) / 3 )  > 6.0;
```
Resultado: _____
Media: _____

2.4.
```
VarLog := True;
VarLog := VarLog  And  ( ( X >= 6.0 ) Or ( Y >= 6.0 ) );
IF Not (VarLog)
THEN Resultado := True
ELSE  Resultado := False;
```
Resultado: _____

2.5.
```
A := A+5;
B := B + C;
IF  ( A > B )
THEN  A := ROUND (X)
ELSE   A :=  TRUNC (Y);
 Media := ( A + B ) / 2;
 IF ( A < 6.0 )
 THEN  A :=  ROUND ( X + 0.5 );
          B :=   6.0;
```
Media: _____
A : _____
B : _____

{ Questão tipo "Teste de Mesa" }

3. Escreva os programas completos (declaração de variáveis, entrada, processamento e saída) que sirvam para a execução das seguintes tarefas especificadas logo abaixo:

3.1. Ler um número inteiro e exibir na tela a mensagem 'Par' se ele for um número par, ou 'Ímpar' se ele for um número ímpar.

3.2. Ler um número qualquer e exibir na tela uma mensagem indicando se ele é positivo, negativo ou nulo (zero). Se ele for positivo, exibir também a raiz quadrada deste número. Se ele for negativo você deve escrever uma mensagem dizendo 'Não é possível calcular a raiz deste número'. Exemplo:

```
Entre com um número: 9
9 é positivo.
A raiz de 9 é igual a 3.
```

3.3. Ler três números inteiros e exibir na tela valores dispostos em ordem crescente e decrescente. Exemplo:

```
Entre com o 1o. número: 5
Entre com o 2o. número: -7
Entre com o 3o. número: 1
Ordem crescente: -7 1 5
Ordem decrescente:  5 1 -7
```

3.4. Faça um programa para calcular a média final de um aluno da UNISINOS, numa disciplina. São lidos: número de matrícula do aluno, a nota do Grau A e a nota do Grau B. Caso o aluno não tenha alcançado a média 6.0, pedir para ele informar a nota do Grau C e informar qual o grau que ele deseja substituir (A ou B). Após a leitura dos dados do aluno, exibir o número de matrícula, seguido da média final deste aluno, conforme indicado no exemplo abaixo:

```
Entre com o número de matrícula : 1234567-8
Entre com a nota do Grau A: 8.2
Entre com a nota do Grau B: 4.1
Média abaixo do limite de aprovação: 5.46
Entre com a nota do Grau C: 6.4
Qual grau que o Grau C substitui: B

Matrícula: 1234567-8
Média Final:  7.00
```

3.5. Efetuar a leitura de três valores (variáveis A, B e C) e efetuar o cálculo das raízes de uma equação de segundo grau. Testar para ver se a equação possui duas raízes, uma única raiz ou se ela não possui raízes reais. Exemplo de tela de saída:

```
Entre com o coeficiente A: 3
Entre com o coeficiente B: 6
Entre com o coeficiente C: 0
As raízes da equação são: -2 e 0.
```

Lembrete:

$$X = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

$$\Delta = B^2 - 4AC \Rightarrow Onde\ \ podemos\ ter: \Delta > 0, \Delta < 0\ ou\ \Delta = 0$$

3.6. Ler uma data de nascimento de uma pessoa fornecida através de uma *string* escrita sob a forma DD/MM/AAAA (DD = Dia, MM = Mês, AAAA = Ano). Testar a validade desta data para saber se esta é uma data válida. Testar os dias válidos: dia > 0, dia <= 28 em fevereiro (29 se o ano for bissexto), dia <= 30 em abril, junho, setembro e novembro, dia <= 31 nos outros meses. Testar a validade do mês: mês > 0 e mês < 13. Testar a validade do ano: ano <= ano atual (constante igual a 1999). Imprimir: "data válida" ou "data inválida" no final da execução do programa.

3.7. Ler os seguintes dados de uma pessoa: nome, sexo (M ou F), idade (0 à 150) e nacionalidade (brasileira ou estrangeira). Testar a validade dos dados fornecidos, indicando se o sexo, a idade e a nacionalidade são válidos ou inválidos. Se um dos dados fornecidos for inválido indicar ao usuário. Se todos os dados forem válidos, exibir uma mensagem como segue, onde aparecem os dados fornecidos:

"*José Silva*, brasileir**o** do sexo **masculino** e maior de idade, está habilitad**o** a dirigir", ou
"*Maria Silva*, brasileir**a** do sexo **feminino** e maior de idade, está habilitad**a** a dirigir", ou
"*Junior Silva*, brasileir**o** do sexo **masculino** e menor de idade, não está habilitad**o** a dirigir".