



Curso de Delphi

2.Dados

Mateus Schwede - UB Social



Variáveis

- Sintaxe: **nomeVar: tipoVariavel** (Ou **nomeVar := valorVar**)
- Globais
 - Escopo e utilizáveis para todo sistema
 - Definidas no 'var' no início do código da Unit
- Locais
 - Escopo e utilizáveis somente na procedure

```
var
    //Variáveis globais aqui
    frmPrepAula2: TfrmPrepAula2;
    x: Integer;

implementation

{$R *.dfm}

procedure TfrmPrepAula2.btnSomaClick(Sender: TObject);
var
    //Variáveis locais da procedure aqui
    res: Double;
begin
    res := StrToFloat(edtVal1.Text) + StrToFloat(edtVal2.Text);
    lblRes.Caption := FloatToStr(res);
end;
```

```
const
    //Constantes aqui
    pi = 3.14;

var
    // Variáveis globais aqui
    frmPrepAula2: TfrmPrepAula2;
    x: Integer;
```



Tipos de dados

- **Integer:** 10
- **Real / Double:** 10.50
- **Boolean:** True / False
- **String:** 'ola'
- **Char:** 'a'
- Tipos definidos pelo usuário (type)

```
type
  kg = Double; //Definido pelo usuário
  TfrmPrepAula2 = class(TForm)
    edtVal1: TEdit;
    edtVal2: TEdit;
    lblRes: TLabel;
    btnSoma: TButton;
    btnSubtracao: TButton;
```

CONVERSÕES:

- ToString
- IntToStr / StrToInt
- StrToFloat / FloatToStr

Conteúdo de Caption e Text é String, precisando de conversão numérica para cálculos

Operadores aritméticos



- (+) Adição: $6+6=12$
- (-) Subtração: $6-5=1$
- (*) Multiplicação: $6*5=30$
- (/) Divisão: $6/3=2$
- (**Div**) Divisão de Integer: $13 \text{ div } 4 = 3$
- (**Mod**) Resto da divisão de Integer: $13 \text{ mod } 4 = 1$



Operadores relacionais

- (**:=**) Atribuição: $x := 5$
- (**=**) Igualdade: $x = 5$
- (**<>**) Diferente: $x <> 6$
- (**<**) Menor: $x < 10$
- (**>**) Maior: $x > 2$
- (**<=**) Menor ou igual: $x \leq 5$
- (**>=**) Maior ou igual: $x \geq 5$

Operadores lógicos

- (**Not**) Negação: $x \text{ not } 6$
- (**And**) Acréscimo: $x > 4 \text{ and } x < 12$
- (**Or**) Escolha: $x > 1 \text{ or } x < 20$

Funções matemáticas



- **ABS(X)**: Valor absoluto: $\text{ABS}(-2) = 2$
- **SQR(X)**: Quadrado: $\text{SQR}(3) = 9$
- **SQRT(X)**: Raiz quadrada: $\text{SQRT}(16) = 4$
- **EXP(X)**: Exponencial: $\text{EXP}(2) = 7,3890560\dots$
- **LN(X)**: Logaritmo natural: $\text{LN}(5) = 1,609437\dots$
- **SIN(X)**: Seno: $\text{SIN}(1) = 0,841470\dots$
- **COS(X)**: Cosseno: $\text{COS}(1) = 0,540302\dots$
- **ARCTAN(X)**: Arco tangente: $\text{ARCTAN}(2) = 1,107148$
- **ROUND(X)**: Arredondamento: $\text{ROUND}(3.8) = 4$
- **TRUN(X) / INT(X)**: Retorna inteiro: $\text{TRUNC}(3.8) = 3$
 - $\text{INT}(X)$ retorna Real, $\text{TRUNC}(X)$ retorna Integer
- **POWER(BASE,POT)**: Potenciação: $\text{POWER}(2,3) = 8$
 - Necessário incluir *Math* na lista de uses!
- **RANDOM(N)**: N° aleatório entre 0 e N-1: $\text{RANDOM}(5) = 4$

Funções String



- **DELETE(St,Pos,N)**: Remover caracteres N na ST, começando na posição Pos
- **INSERT(St1,St2,Pos)**: Inserir caracteres St1 na St2 a partir da posição Pos
- **COPY(St,Pos,N)**: Copia N caracteres de St a partir da posição Pos
- **LENGTH(St)**: Tamanho de St (Quantidade de caracteres)
- **POS(St1,St2)**: Procura em substring St1 da String St. Resultado é a posição do 1º caractere da substring. Se não encontrar substring, será 0

Anatomia do código



```
unit uPrepAula2;

interface

uses
  // Dependências da Unit
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
  System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Math;
// Inserir dependência 'Math' para utilizar função POWER

type
  kg = Double; // Tipo definido pelo usuário

TfrmPrepAula2 = class(TForm)
  edtVall: TEdit;
  edtVal2: TEdit;
  lblRes: TLabel;
  btnSoma: TButton;
  btnSubtracao: TButton;
  btnMultiplicacao: TButton;
  btnDivisao: TButton;
  procedure btnSomaClick(Sender: TObject);
private
  { Private declarations }
  // Coisas declaradas em private podem ser locais, somente acessadas na própria Unit
public
  { Public declarations }
  // Coisas declaradas em public são globais, acessadas por outras Units vinculadas
end;
```


Anatomia do código



```
const
    // Constantes aqui
    pi = 3.14;

var
    // Variáveis globais aqui
    frmPrepAula2: TfrmPrepAula2;
    x: Integer;

implementation

{$R *.dfm}
// Functions globais aqui

// Procedure de evento de componente:
procedure TfrmPrepAula2.btnSomaClick(Sender: TObject);
// Funções locais aqui
var
    // Variáveis locais da procedure aqui
    res: Double;
begin
    // res := StrToFloat(edtVal1.Text) + StrToFloat(edtVal2.Text);
    // lblRes.Caption := FloatToStr(res);
    lblRes.Caption := FloatToStr(Power(2, 3));
end;

end.
```

Condição IF then ELSE



```
var
  x: Integer;
  res: String;
begin
  x := 10;
  if x > 0 then
  begin
    res := 'X maior que zero';
  end
  else
  begin
    res := 'X menor que zero';
  end;
end;
```

```
var
  x: Integer;
begin
  x := -10;
  if x >= 0 then
  begin
    if x <> 0 then
    begin
      lblRes.Caption := 'X é positivo';
    end
    else
    begin
      lblRes.Caption := 'X é igual a 0';
    end;
  end
  else
  begin
    lblRes.Caption := 'X é negativo';
  end;
end;
```

Observações



- Código gerado pelo Delphi é problema do Delphi!
- Mostrar arquivos pertinentes que serão compartilhados com outros devs
 - Delphi Form
 - Delphi Source File
 - Delphi Project File (2 arquivos)
 - Arquivo RES (se possui img)



UB Social

ubsocial.github.io