

FACULDADE REGES RIBEIRÃO PRETO

REVISÃO ALGORITMO E LÓGICA DE PROGRAMAÇÃO

PROF. JOSÉ CARLOS DOS REIS JUNIOR

Variáveis

Variáveis são espaços na memória do computador destinados a armazenar valores. Elas possuem um nome único que serve como uma referência para acessar e manipular esses valores. As variáveis podem armazenar diferentes tipos de dados, como números inteiros, números de ponto flutuante, strings, valores booleanos, entre outros.

Exemplos:

1. Variáveis do tipo inteiro:

```
``delphi
// Exemplo em Delphi:
var
    idade: Integer;
begin
    idade := 25;
end;
``
```

2. Variáveis do tipo real (ou ponto flutuante):

```
```delphi
```

```
// Exemplo em Delphi:
```

```
var
```

```
 preco: Double;
```

```
begin
```

```
 preco := 10.50;
```

```
end;
```

```
```
```

3. Variáveis do tipo texto:

```
```delphi
```

```
// Exemplo em Delphi:
```

```
var
```

```
 nome: String;
```

```
begin
```

```
 nome := 'João';
```

```
end;
```

```
```
```

4. Variáveis lógicas (verdadeiro ou falso):

```
```delphi
```

```
// Exemplo em Delphi:
```

```
var
```

```
 aprovado: Boolean;
```

```
begin
```

```
 aprovado := True;
```

```
end;
```

```
```
```

Variáveis são fundamentais na lógica de programação, pois permitem armazenar e manipular dados, tornando possível criar algoritmos e programas que realizam tarefas específicas.

Variáveis Inteiro:

As variáveis do tipo inteiro (Integer) no Delphi são usadas para armazenar números inteiros, ou seja, números sem casas decimais.

Elas podem representar valores positivos, negativos ou zero.

Variáveis Real:

As variáveis do tipo real (Double) no Delphi são usadas para armazenar números de ponto flutuante, ou seja, números com casas decimais.

Elas podem representar valores positivos, negativos ou zero e têm uma precisão maior que as variáveis do tipo Single.

Variáveis String:

As variáveis do tipo string (String) no Delphi são usadas para armazenar texto ou sequências de caracteres.

Elas podem representar nomes, mensagens, endereços, entre outros.

Conversões de tipo variáveis:

1. Conversão de Inteiro para String:

Definição: A conversão de inteiro para string é o processo de transformar um valor numérico inteiro em uma representação textual.

Exemplo em Delphi:

[MOSTRAR RESULTADO EM LABEL, EXEMPLO](#)

```
``delphi
var
  idade: Integer;
  idadeTexto: String;
begin
  idade := 25;
  idadeTexto := IntToStr(idade);
end;
``
```

Neste exemplo, a variável "idade" é do tipo Integer e possui o valor 25. A função `IntToStr()` é utilizada para converter esse valor inteiro em uma string, e o resultado é atribuído à variável "idadeTexto".

2. Conversão de Real para String:

Definição: A conversão de real para string é o processo de transformar um valor numérico de ponto flutuante em uma representação textual.

Exemplo em Delphi:

[USADO PARA MOSTRAR RESULTADO EM LABEL](#)

```
``delphi
var
  preco: Double;
  precoTexto: String;
begin
  preco := 10.50;
```

```
precoTexto := FloatToStr(preco);  
end;  
...
```

Neste exemplo, a variável "preco" é do tipo Double e possui o valor 10.50. A função `FloatToStr()` é utilizada para converter esse valor real em uma string, e o resultado é atribuído à variável "precoTexto".

3. Conversão de String para Inteiro:

Definição: A conversão de string para inteiro é o processo de transformar uma representação textual em um valor numérico inteiro.

Exemplo em Delphi:

```
``delphi
```

NESSE CASO, UTILIZA O DIGITADO PARA FAZER
UMA CONTA, POR EXEMPLO

```
var
```

```
idadeTexto: String;
```

```
idade: Integer;
```

```
begin
```

```
idadeTexto := '25';
```

```
idade := StrToInt(idadeTexto);
```

```
end;
```

```
...
```

Neste exemplo, a variável "idadeTexto" é do tipo String e possui o valor '25'. A função `StrToInt()` é utilizada para converter essa string em um valor inteiro, e o resultado é atribuído à variável "idade".

Esses são exemplos de como realizar conversões de variáveis nos tipos inteiro, real e string em Delphi. Essas operações são úteis para manipular diferentes tipos de dados em um programa Delphi.

1. StrToFloat:

Definição: A função `StrToFloat` converte uma `string` em um valor de ponto flutuante (real).

Exemplo:

```
``delphi
var
    strNumero: String;
    numero: Double;
begin
    strNumero := '10.5';
    numero := StrToFloat(strNumero);
end;
``
```

2. FloatToStr:

Definição: A função `FloatToStr` converte um valor de ponto flutuante (real) em uma `string`.

Exemplo:

```
``delphi
var
    numero: Double;
    strNumero: String;
begin
    numero := 10.5;
    strNumero := FloatToStr(numero);
end;
```

...

3. IntToStr:

Definição: A função `IntToStr` converte um valor inteiro em uma string.

Exemplo:

```
``delphi
var
  valorInteiro: Integer;
  strValor: String;
begin
  valorInteiro := 42;
  strValor := IntToStr(valorInteiro);
end;
...
```

4. StrToInt:

Definição: A função `StrToInt` converte uma string em um valor inteiro.

Exemplo:

```
``delphi
var
  strNumero: String;
  numero: Integer;
begin
  strNumero := '42';
  numero := StrToInt(strNumero);
end;
...
```

Essas funções são úteis para converter valores entre diferentes tipos de dados em Delphi. Elas permitem que você manipule e processe dados de forma flexível em seus programas.

Principais componentes delphi:

Claro! Aqui estão as definições e exemplos dos componentes em Delphi, bem como da função `ShowMessage`:

1. Label:

Definição: O componente **Label** é usado para exibir **texto estático** em um formulário Delphi. Ele é usado principalmente para **fornecer informações aos usuários ou identificar outros controles.**

Exemplo:

```
``delphi
object Label1: TLabel
  Left = 20
  Top = 20
  Caption = 'Bem-vindo ao meu programa!'
end
``
```

2. Edit:

Definição: O componente **Edit** é usado para **permitir que o usuário insira e edite texto em um formulário Delphi**. Ele é usado em campos de entrada de dados.

Exemplo:

```
``delphi
object Edit1: TEdit
  Left = 20
  Top = 50
  Width = 150
end
``
```

3. Button:

Definição: O componente **Button** é usado para **criar botões clicáveis em um formulário Delphi**. Eles são usados para acionar ações quando clicados pelo usuário.

Exemplo:

```
``delphi
object Button1: TButton
  Left = 20
  Top = 100
  Caption = 'Clique aqui'
end
``
```

4. ListBox:

Definição: O componente `ListBox` é usado para exibir uma lista de itens que o usuário pode selecionar. Ele é usado para apresentar uma lista de opções ou dados ao usuário.

Exemplo:

```
```delphi
object ListBox1: TListBox
 Left = 20
 Top = 150
 Width = 150
 Items.Strings = (
 'Item 1'
 'Item 2'
 'Item 3'
)
end
```
```

5. `ShowMessage`:

Definição: A função `ShowMessage` exibe uma mensagem de alerta ou informativa em uma janela de diálogo modal. Ela é frequentemente usada para fornecer feedback aos usuários em aplicativos Delphi.

Exemplo:

```
```delphi
 ShowMessage('Olá, mundo!');
```
```

Esses são exemplos básicos de como usar os componentes `Label`, `Edit`, `Button` e `ListBox` em um aplicativo Delphi, além da função `ShowMessage` para exibir mensagens aos usuários.

Propriedades principais dos objetos citados:

1. Name:

Definição: A propriedade `Name` é usada para definir o nome único de um componente no formulário Delphi. É útil para identificar e referenciar o componente em tempo de design e em código-fonte.

Exemplo:

```
``delphi
object Button1: TButton
    Name = 'btnOK'
    Caption = 'OK'
end
...
```

2. Caption:

Definição: A propriedade `Caption` é usada para definir o texto exibido em um componente, como um botão, label ou janela. É usado principalmente para fornecer informações ou instruções aos usuários.

Exemplo:

```
``delphi
object Button1: TButton
    Caption = 'Clique aqui'
end
...
```

3. Text:

Definição: A propriedade `Text` é usada para definir ou obter o texto contido em um componente que suporta entrada ou exibição de texto, como um edit ou memo. É usado para exibir e manipular o texto em tempo de execução.

Exemplo:

```
``delphi
object Edit1: TEdit
    Text = 'Texto inicial'
end
``
```

Essas propriedades são fundamentais para personalizar e controlar o comportamento dos componentes em um aplicativo Delphi. Elas permitem definir nomes, textos e outras características visuais e funcionais dos componentes.

The screenshot shows a Delphi application window titled "Cálculo de Média Anual". The window contains a form with the following elements:

- A label "Nome do Aluno:" followed by a text input field.
- Four labels "Bimestre 1:", "Bimestre 2:", "Bimestre 3:", and "Bimestre 4:" each followed by a text input field.
- A label "Média" followed by a text input field.
- A label "Status" followed by a text input field.
- Two buttons: "Calcular" and "Limpar".

At the bottom of the window, there is a status bar with the text "Iblhora" and "Ibldata".

unit UnitMedia; // Declaração da unit (unidade) onde o formulário está localizado

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ExtCtrls; //

Declaração das units que serão utilizadas

type

Tfrmmedia = class(TForm) // Declaração da classe do formulário

Panel1: TPanel; // Painel para agrupar componentes

Panel2: TPanel; // Painel para agrupar componentes

Panel3: TPanel; // Painel para agrupar componentes

lbalunos: TListBox; // ListBox para exibir dados dos alunos

Timer1: TTimer; // Timer para atualizar a data e hora

lblhora: TLabel; // Label para exibir a hora

lbldata: TLabel; // Label para exibir a data

Panel4: TPanel; // Painel para agrupar componentes

Label7: TLabel; // Label para título

Label1: TLabel; // Label para título

Label2: TLabel; // Label para título

Label4: TLabel; // Label para título

Label3: TLabel; // Label para título

edtn1: TEdit; // Edit para entrada da nota 1

edtn2: TEdit; // Edit para entrada da nota 2

edtn3: TEdit; // Edit para entrada da nota 3

edtn4: TEdit; // Edit para entrada da nota 4

edtmedia: TEdit; // Edit para exibir a média

edtstatus: TEdit; // Edit para exibir o status (aprovado ou reprovado)

```

    btncalcular: TButton; // Botão para calcular a média
    Label6: TLabel; // Label para título
    Label5: TLabel; // Label para título
    btnlimpar: TButton; // Botão para limpar os campos
    edtnome: TEdit; // Edit para entrada do nome do aluno

    procedure btncalcularClick(Sender: TObject); // Evento de clique do
botão de calcular

    procedure btnlimparClick(Sender: TObject); // Evento de clique do
botão de limpar

    procedure Timer1Timer(Sender: TObject); // Evento do timer para
atualizar a data e hora

    procedure FormClose(Sender: TObject; var Action: TCloseAction); //
Evento de fechamento do formulário

private
    { Private declarations }
public
    { Public declarations }
end;

var
    frmmedia: Tfrmmedia; // Variável global do formulário

    nota1,nota2,nota3,nota4,media:real; // Declaração de variáveis para
notas e média

    status:string; // Declaração de variável para o status (aprovado ou
reprovado)

implementation

{$R *.dfm} // Inclusão do arquivo de recursos do formulário

```

```
procedure Tfrmmedia.btncalcularClick(Sender: TObject); // Procedimento  
para calcular a média
```

```
begin
```

```
    nota1:=strtofloat(edtn1.Text); // Conversão da nota 1 para float
```

```
    nota2:=strtofloat(edtn2.Text); // Conversão da nota 2 para float
```

```
    nota3:=strtofloat(edtn3.Text); // Conversão da nota 3 para float
```

```
    nota4:=strtofloat(edtn4.Text); // Conversão da nota 4 para float
```

```
    media:= (nota1+nota2+nota3+nota4)/4; // Cálculo da média
```

```
    if media>=7 then // Verificação se a média é maior ou igual a 7
```

```
        status:='Aprovado' // Se sim, aluno está aprovado
```

```
    else
```

```
        status:='Reprovado'; // Se não, aluno está reprovado
```

```
    edtmedia.Text:=floattostr (media); // Conversão da média para string e  
exibição no campo de texto
```

```
    edtstatus.Text:= status; // Exibição do status no campo de texto
```

```
    lbalunos.Items.Add('-----'); // Adição de linha  
separadora na ListBox
```

```
    lbalunos.Items.Add(edtnome.Text); // Adição do nome do aluno na  
ListBox
```

```
    lbalunos.Items.Add('Bimestre 1: '+ edtn1.Text); // Adição da nota 1 na  
ListBox
```

```
    lbalunos.Items.Add('Bimestre 2: '+ edtn2.Text); // Adição da nota 2 na  
ListBox
```

```
    lbalunos.Items.Add('Bimestre 3: ' + edtn3.Text); // Adição da nota 3 na  
ListBox
```

```
    lbalunos.Items.Add('Bimestre 4: ' + edtn4.Text); // Adição da nota 4 na  
ListBox
```

```
    lbalunos.Items.Add('Média' + edtmedia.Text); // Adição da média na  
ListBox
```

```
lbalunos.Items.Add('Status' + edtstatus.Text); // Adição do status na  
ListBox
```

```
lbalunos.Items.Add('-----'); // Adição de linha  
separadora na ListBox
```

```
end;
```

```
procedure Tfrmmedia.btnlimparClick(Sender: TObject); // Procedimento  
para limpar os campos
```

```
begin
```

```
edtn1.Clear; // Limpeza do campo de nota 1
```

```
edtn2.Clear; // Limpeza do campo de nota 2
```

```
edtn3.Clear; // Limpeza do campo de nota 3
```

```
edtn4.Clear; // Limpeza do campo de nota 4
```

```
edtmedia.Clear; // Limpeza do campo de média
```

```
edtstatus.Clear; // Limpeza do campo de status
```

```
lbalunos.Clear; // Limpeza da ListBox
```

```
edtnome.Clear; // Limpeza do campo de nome
```

```
end;
```

```
procedure Tfrmmedia.FormClose(Sender: TObject; var Action:  
TCloseAction); // Procedimento para fechar o formulário
```

```
begin
```

```
showmessage('Obrigado por utilizar meu cálculo de média! By Prof.  
José Carlos'); // Exibição de mensagem de agradecimento
```

```
end;
```

```
procedure Tfrmmedia.Timer1Timer(Sender: TObject); // Procedimento  
para atualizar a data e hora
```

```
begin
```

```
lbldata.Caption:=datetostr(date); // Atualização da data na Label
```

```
lblhora.Caption:=timetostr(time); // Atualização da hora na Label
```

```
end;
```



```
end. // Fim da implementação da unit
```

Código sem comentários:

```
unit UnitMedia;
```

```
interface
```

```
uses
```

```
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,  
    System.Classes, Vcl.Graphics,  
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ExtCtrls;
```

```
type
```

```
    Tfrmmedia = class(TForm)
```

```
        Panel1: TPanel;
```

```
        Panel2: TPanel;
```

```
        Panel3: TPanel;
```

```
        lbalunos: TListBox;
```

```
        Timer1: TTimer;
```

```
        lblhora: TLabel;
```

```
        lbldata: TLabel;
```

```
        Panel4: TPanel;
```

```
        Label7: TLabel;
```

```
        Label1: TLabel;
```

```
        Label2: TLabel;
```

```
        Label4: TLabel;
```

```
        Label3: TLabel;
```

```
        edtn1: TEdit;
```

```
        edtn2: TEdit;
```

```
        edtn3: TEdit;
```

```

edtn4: TEdit;
edtmedia: TEdit;
edtstatus: TEdit;
btncalcular: TButton;
Label6: TLabel;
Label5: TLabel;
btnlimpar: TButton;
edtnome: TEdit;
procedure btncalcularClick(Sender: TObject);
procedure btnlimparClick(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    frmmedia: Tfrmmedia;

    nota1,nota2,nota3,nota4,media:real;
    status:string;

implementation

{$R *.dfm}

procedure Tfrmmedia.btncalcularClick(Sender: TObject);
begin

```

```
nota1:=strtofloat(edtn1.Text);
nota2:=strtofloat(edtn2.Text);
nota3:=strtofloat(edtn3.Text);
nota4:=strtofloat(edtn4.Text);
media:= (nota1+nota2+nota3+nota4)/4;
```

```
if media>=7 then
status:='Aprovado';
if media<7 then
status:='Reprovado';
```

```
edtmedia.Text:=floattostr (media);
```

```
edtstatus.Text:= status;
lbalunos.Items.Add('-----');
lbalunos.Items.Add(edtnome.Text);
lbalunos.Items.Add('Bimestre 1: '+ edtn1.Text);
lbalunos.Items.Add('Bimestre 2: '+ edtn2.Text);
lbalunos.Items.Add('Bimestre 3: ' + edtn3.Text);
lbalunos.Items.Add('Bimestre 4: ' + edtn4.Text);
lbalunos.Items.Add('Média' + edtmedia.Text);
lbalunos.Items.Add('Status' + edtstatus.Text);
```

```
lbalunos.Items.Add('-----');
```

```
end;
```

```
procedure Tfrmmedia.btnlimparClick(Sender: TObject);
begin
```

```

edtn1.Clear;
edtn2.Clear;
edtn3.Clear;
edtn4.Clear;
edtmedia.Clear;
edtstatus.Clear;
lbalunos.Clear;
edtnome.Clear;

end;

procedure Tfrmmedia.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
    showmessage('Obrigado por utilizar meu cálculo de média! By Prof. José
Carlos');
end;

procedure Tfrmmedia.Timer1Timer(Sender: TObject);
begin
    lbldata.Caption:=datetostr(date);
    lblhora.Caption:=timetostr(time);

end;

end.

```

A estrutura condicional, também conhecida como instrução condicional, é um

conceito fundamental na programação que permite tomar decisões com base em uma condição específica. No código comentado anteriormente, a estrutura condicional foi utilizada para determinar se um aluno está aprovado ou reprovado com base na média de suas notas. Aqui está a definição:

Estrutura Condicional:

A estrutura condicional é uma instrução de controle de fluxo que permite executar diferentes blocos de código com base em uma condição específica. Em Delphi, a estrutura condicional mais comum é a instrução `if`, que avalia uma expressão booleana e executa um bloco de código se a condição for verdadeira.

Exemplo:

```
``delphi
if media >= 7 then
    status := 'Aprovado'
else
    status := 'Reprovado';
...
```

Neste exemplo, se a média do aluno for maior ou igual a 7, a variável `status` recebe o valor 'Aprovado', caso contrário, recebe o valor 'Reprovado'. A estrutura condicional `if` permite que o programa tome uma decisão com base no resultado da expressão `media >= 7`. Se a condição for verdadeira, o bloco de código após o `then` é executado; caso contrário, o bloco de código após o `else` é executado (se presente).

Esta estrutura condicional permite que o programa selecione dinamicamente o caminho a ser seguido com base em uma condição específica, tornando o código mais flexível e adaptável.

Queridos alunos,

Estamos chegando perto da nossa prova do dia 17/04 e sei que todos vocês estão se esforçando muito para se prepararem da melhor forma possível. Quero lembrar a todos que o esforço e a dedicação que vocês têm demonstrado até agora serão recompensados.

Lembrem-se de revisar bem todo o conteúdo que estudamos juntos, tirem um tempo para resolver exercícios, revisar os exemplos e tirar todas as dúvidas que possam surgir. Acredito em cada um de vocês e sei que são capazes de alcançar excelentes resultados!

Mantenham o foco, a calma e confiem no trabalho que fizeram até agora. Estarei torcendo por cada um de vocês no dia da prova.

Boa sorte e vamos com tudo!

Prof. José Carlos.