

Aula 01 - Conteúdo

- 1) Apresentação da disciplina de Estrutura de Dados e Algoritmos – Ementa, objetivo, programa, critérios de avaliação, bibliografia básica etc.
- 2) Revisão da linguagem Object Pascal
- 3) Definição de registros, classes e objetos
- 4) Apontadores e Variáveis Dinâmicas
- 5) Exercícios

Revisão da Linguagem Object Pascal

Material anexo.

Definição de registros, classes e objetos

Registro

Registro: coleção de componentes de tipos arbitrários, logicamente relacionados. Um componente individual é denominado campo.

Exemplo 1.1 – record (arquivos ProjEx101.dpr e Ex101.pas)

```
unit Ex101;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

  RegEst = record
    Nome,Endereco: string;
    RA, NCred: integer;
    MF: real;
  end;

var
  Form1: TForm1;
  Est,Melhor: RegEst;

implementation
{$R *.DFM}
Procedure Inic_Dados;
begin
  Est.Nome:='Fujiro Nakombi';
  Est.Endereco:='Av. Liberdade';
  Est.MF := 7.6;
  Est.NCred := 25;
  Est.RA := 3901;
  Melhor.Nome := 'Bruce Wayne';
```

```

    Melhor.Endereco:='R. Gotan City';
    Melhor.MF := 6.5;
    Melhor.NCred := 30;
    Melhor.RA := 3920;
end;

procedure principal;
begin
    Inic_Dados;
    if ( Est.MF >= 5 ) then ShowMessage('Aprovado: '+ Est.Nome);
    if ( Est.MF > Melhor.MF ) then
    begin
        Melhor.Nome:=Est.Nome;
        Melhor.Endereco:=Est.Endereco;
        Melhor.NCred := Est.NCred;
        Melhor.MF := Est.MF;
        Melhor.RA := Est.RA;
    end;
    ShowMessage('Melhor aluno: '+IntToStr(Melhor.RA)+'-'+Melhor.Nome);
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
    principal;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
    Form1.Close;
end;
end.

```

Exemplo 1.2 – class (arquivos ProjEx102.dpr e Ex102.pas)

```

unit Ex102;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls;

type
    TForm1 = class(TForm)
        Button1: TButton;
        Button2: TButton;
        procedure Button1Click(Sender: TObject);
        procedure Button2Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

    RegEst = class
        Nome,Endereco: string;
        RA, NCred: integer;
        MF: real;
        Constructor Inicializa(no,ender:string;reg,nc:integer;med:real);
        Procedure DefDados(no,ender:string;reg,nc:integer;med:real);
    end;

var
    Form1: TForm1;
    Est,Melhor: RegEst;

```

```

implementation
{$R *.DFM}

Constructor
RegEst.Inicializa(no,ender:string;reg,nc:integer;med:real);
begin
    Nome:=no;
    Endereco:=ender;
    MF := med;
    NCred := nc;
    RA := reg;
end;

Procedure RegEst.DefDados(no,ender:string;reg,nc:integer;med:real);
begin
    Nome:=no;
    Endereco:=ender;
    MF := med;
    NCred := nc;
    RA := reg;
end;

procedure principal;
begin
    Est:=RegEst.Inicializa('Fujiro Nakombi','Av.
Liberdade',3901,25,7.6);
    Melhor:=RegEst.Inicializa('Bruce Wayne','R. Gotham',3920,30,6.5);
    if ( Est.MF >= 5 ) then ShowMessage('Aprovado: '+ Est.Nome);
    if ( Est.MF > Melhor.MF ) then
        begin
            Melhor.DefDados(Est.Nome,Est.Endereco,Est.RA,Est.NCred,Est.MF);
        end;
    ShowMessage('Melhor aluno: '+IntToStr(Melhor.RA)+'-'+Melhor.Nome);
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
    principal;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
    Form1.Close;
end;

end.

```

Exemplo 1.3 – vetor de class (arquivos ProjEx103.dpr e Ex103.pas)

```

unit Ex103;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls;

type
    TForm1 = class(TForm)
        Button1: TButton;
        Button2: TButton;
        procedure Button1Click(Sender: TObject);
        procedure Button2Click(Sender: TObject);
    end;

```

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

Peca = class
  Cod: integer;
  Desc: string;
  Preco: real;
  Constructor Inicializa(c:integer;d:string;p:real);
  Procedure SetDados(c:integer;d:string;p:real);
  Procedure MostraTela;
end;

var
  Form1: TForm1;
  Estoque: array[1..5] of Peca;

implementation
{$R *.DFM}

Constructor Peca.Inicializa(c:integer;d:string;p:real);
begin
  Cod:=c; Desc:=d; Preco:=p;
end;

Procedure Peca.SetDados(c:integer;d:string;p:real);
begin
  Cod:=c; Desc:=d; Preco:=p;
end;

Procedure Peca.MostraTela;
begin
  ShowMessage(IntToStr(Cod)+'-'+Desc+' R$'+CurrToStr(Preco));
end;

Procedure Principal;
var k:integer;
begin
  Estoque[1]:=Peca.Inicializa(10,'TV',450.80);
  Estoque[2]:=Peca.Inicializa(22,'Video',380);
  Estoque[3]:=Peca.Inicializa(37,'Som',715.60);
  Estoque[4]:=Peca.Inicializa(41,'DVD',1200.45);
  Estoque[5]:=Peca.Inicializa(55,'FAX',600);
  for k:=1 to 5 do
    Estoque[k].MostraTela;
  end;

  procedure TForm1.Button1Click(Sender: TObject);
  begin
    Principal;
  end;

  procedure TForm1.Button2Click(Sender: TObject);
  begin
    Form1.Close;
  end;
end.
```

Exemplo 1.4 – class aninhada (arquivos ProjEx104.dpr e Ex104.pas)

```
unit Ex104;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

  Data = class
    ano,mes,dia:integer;
    Constructor Inicializa(a,m,d:integer);
    Function MontaString: string;
  end;

  Cheque = class
    Num: integer;
    Valor: real;
    ChData: Data;
    Constructor Inicializa(n,a,m,d:integer;v:real);
    Procedure MostraTela;
  end;

var
  Form1: TForm1;
  ch1, ch2: Cheque;

implementation
{$R *.DFM}
Constructor Data.Inicializa(a,m,d:integer);
begin
  ano:=a; mes:=m; dia:=d;
end;

Function Data.MontaString: string;
begin
  result:=IntToStr(dia)+'/'+IntToStr(mes)+'/'+IntToStr(ano);
end;

Constructor Cheque.Inicializa(n,a,m,d:integer;v:real);
begin
  Num:=n; Valor:=v;
  ChData := Data.Inicializa(a,m,d);
end;

Procedure Cheque.MostraTela;
begin
  ShowMessage(IntToStr(Num)+' : R$'+CurrToStr(Valor)+
    #13+ChData.MontaString);
end;
```

```

Procedure Principal;
begin
  ch1 := Cheque.Inicializa(100,2000,07,29,1500.60);
  ch2 := Cheque.Inicializa(320,2000,05,12,700);
  if ( ch1.Valor > ch2.Valor ) then ch1.MostraTela
  else ch2.MostraTela;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  Principal;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  Form1.Close;
end;
end.

```

Passagem de Parâmetros

Exemplo 1.5 - Passagem de parâmetros por valor e por referência
(arquivos ProjEx105.dpr e Ex105.pas)

```

unit Ex105;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  x,y: integer;

implementation
{$R *.DFM}

Procedure FuncValor(a,b:integer);
begin
  a := a + 5;
  b := b + 2;
end;

Procedure FuncRef(var a,b:integer);
begin
  a := a + 5;
  b := b + 2;
end;

```

```

Procedure Principal;
begin
  x:=10; y:=6;
  ShowMessage('Inicialmente x = '+IntToStr(x)+' y = '+IntToStr(y));
  FuncValor(x,y);
  ShowMessage('Após FuncValor: x = '+IntToStr(x)+' y = '+IntToStr(y));
  FuncRef(x,y);
  ShowMessage('Após FuncRef: x = '+IntToStr(x)+' y = '+IntToStr(y));
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  Principal;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  Form1.Close;
end;

end.

```

Apontadores e Variáveis Dinâmicas

Def: uma variável dinâmica é uma variável criada e destruída explicitamente durante a execução do programa.

A variável dinâmica não é declarada na parte de declaração de variáveis, pois a mesma não existe antes da execução do programa. A variável dinâmica é referenciada através de um endereço de memória. O endereço de memória é armazenado numa variável chamada apontador (pointer) que deve ser declarada e compatível com o tipo de variável dinâmica ao qual está relacionada.

Um apontador contém o endereço de um objeto sendo então possível acessar o objeto indiretamente através do apontador.

Exemplo 1.6 – Apontadores/endereço de memória (arquivos ProjEx106.dpr e Ex106.pas)

```

unit Ex106;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    ButtonExemploA: TButton;
    ButtonSair: TButton;
    Label1: TLabel;
    LabelX: TLabel;
    Label3: TLabel;
    LabelY: TLabel;
    ButtonExemploB: TButton;
    Label2: TLabel;
    ButtonExemploC: TButton;
    Label4: TLabel;
    procedure ButtonExemploAClick(Sender: TObject);
    procedure ButtonSairClick(Sender: TObject);
    procedure SetLabelX(str: string);
    procedure SetLabelY(str: string);
  end;

```

```

    procedure ButtonExemploBClick(Sender: TObject);
    procedure ButtonExemploCClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;

implementation
{$R *.DFM}

Procedure ExemploA;
var
    x,y: integer;
    px,py: ^integer;
begin
    ShowMessage('Inicialmente x = 5 e y = 9');
    x:=5;
    Form1.SetLabelX(IntToStr(x));
    y:=9;
    Form1.SetLabelY(IntToStr(y));
    px := @x;
    py := @y;
    px^ := px^ + 10;
    py^ := 2 * py^;
    ShowMessage('Pressione OK para obter os valores de x e y após
operação com apontadores:');
    Form1.SetLabelX(IntToStr(x));
    Form1.SetLabelY(IntToStr(y));
end;

Procedure ExemploB;
var
    k,x: integer;
    pa: ^integer;
    a: array [0..9] of integer;
begin
    for k:=0 to 9 do a[k] := k*k;
    pa := @a[0]; //equivalente a pa = a;
    x := pa^;    //equivalente a x = a[0];
    ShowMessage('x = a[0] = '+IntToStr(x));
    Inc(pa);     //pa = pa + 1
    x := pa^;    // equivalente a x = a[1];
    ShowMessage('x = a[1] = '+IntToStr(x));
    Inc(pa,3);   // pa = pa + 3;
    x := pa^;    // equivalente a x = a[4];
    ShowMessage('x = a[4] = '+IntToStr(x));
    Inc(pa,5);   //pa = pa + 5;
    x := pa^;    // equivalente a x = a[9];
    ShowMessage('x = a[9] = '+IntToStr(x));
    Inc(pa,-2);  // pa= pa - 2;
    x := pa^;    // equivalente a x = a[7];
    ShowMessage('x = a[7] = '+IntToStr(x));
end;

Procedure ExemploC;
var
    x: char;

```



```

    pa: ^char;
    a: string;
begin
    a := 'ABCDEFGHJIJ';
    pa := @a[1]; //equivalente a pa = a;
    x := pa^;    //equivalente a x = a[1];
    ShowMessage('x = a[1] = '+ x);
    Inc(pa);     //pa = pa + 1
    x := pa^;    // equivalente a x = a[2];
    ShowMessage('x = a[2] = '+ x);
    Inc(pa,3);   // pa = pa + 3;
    x := pa^;    // equivalente a x = a[5];
    ShowMessage('x = a[5] = '+x);
    Inc(pa,5);   //pa = pa + 5;
    x := pa^;    // equivalente a x = a[10];
    ShowMessage('x = a[10] = '+x);
    Inc(pa,-2);  // pa= pa - 2;
    x := pa^;    // equivalente a x = a[8];
    ShowMessage('x = a[8] = '+x);
end;

procedure TForm1.ButtonExemploAClick(Sender: TObject);
begin
    ExemploA;
end;

procedure TForm1.ButtonExemploBClick(Sender: TObject);
begin
    ExemploB;
end;

procedure TForm1.ButtonExemploCClick(Sender: TObject);
begin
    ExemploC;
end;

procedure TForm1.ButtonSairClick(Sender: TObject);
begin
    Form1.Close;
end;

Procedure TForm1.SetLabelX(str: string);
begin
    LabelX.Caption:=str;
end;

Procedure TForm1.SetLabelY(str: string);
begin
    LabelY.Caption:=str;
end;
end.

```

Exemplo 1.7 - Alocação Dinâmica de vetores
(arquivos ProjEx107.dpr e Ex107.pas)

```

unit Ex107;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls;

```

```

type
  TForm1 = class(TForm)
    ButtonAlocar: TButton;
    EditTamVet: TEdit;
    Label1: TLabel;
    ButtonSair: TButton;
    procedure ButtonAlocarClick(Sender: TObject);
    procedure ButtonSairClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  x: array of integer;

implementation
{$R *.DFM}

procedure TForm1.ButtonAlocarClick(Sender: TObject);
  var tam: integer;
begin
  tam := StrToInt(EditTamVet.text);
  if tam > 0 then
  begin
    SetLength(x,tam);
    ShowMessage('Tamanho do vetor x: ' + IntToStr(Low(x)) + ' a ' +
      IntToStr(High(x)));
  end
  else
    ShowMessage('O tamanho do vetor deve ser maior que 0');
end;

procedure TForm1.ButtonSairClick(Sender: TObject);
begin
  Form1.Close;
end;

end.

```

Exemplo 1.8 - Alocação Dinâmica de estruturas: Record e class
(arquivos ProjEx108.dpr e Ex108.pas)

```

unit Ex108;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    ButtonExecutarA: TButton;
    ButtonSair: TButton;
    ButtonExecutarB: TButton;
    procedure ButtonExecutarAClick(Sender: TObject);
    procedure ButtonSairClick(Sender: TObject);
    procedure ButtonExecutarBClick(Sender: TObject);
  private
    { Private declarations }

```

```
public
{ Public declarations }
end;

PLivro = ^Livro;
Livro = record
    tit, autor: string;
    cod: integer;
    preco: real;
    prox: PLivro;
end;

Produto = class
    cod: integer;
    desc: string;
    preco: real;
    prox: Produto;
end;

var
    Form1: TForm1;
    InicLivro: PLivro;
    InicProd: Produto;

implementation

{$R *.DFM}

Procedure ExemploA;
var str: string;
    p1,p2,p3: PLivro;
begin
    New(InicLivro);
    InicLivro^.tit := 'Cem Anos de Solidão';
    InicLivro^.autor := 'Gabriel Garcia Marques';
    InicLivro^.cod := 10; InicLivro^.preco := 45.60;
    InicLivro^.prox := nil;
    New(p1);
    p1^.tit := 'O Primo Basílio';
    p1^.autor := 'Eça de Queiroz';
    p1^.cod := 20; p1^.preco := 60;
    p1^.prox := nil;
    New(p2);
    p2^.tit := 'Memórias Póstumas de Brás Cubas';
    p2^.autor := 'Machado de Assis';
    p2^.cod := 30; p2^.preco := 55;
    p2^.prox := nil;
    New(p3);
    p3^.tit := 'Iracema';
    p3^.autor := 'José de Alencar';
    p3^.cod := 40; p3^.preco := 38;
    p3^.prox := nil;
    //Ligando as estruturas
    InicLivro^.prox := p1;
    p1^.prox := p2;
    p2^.prox := p3;
    //Percorrendo a lista
    p1 := InicLivro;
    while (p1 <> nil) do
    begin
        str := IntToStr(p1^.cod) + ': ' + p1^.tit;
```

```
    str := str + #13 + p1^.autor + ' R$ '+CurrToStr(p1^.preco);
    ShowMessage(str);
    p1 := p1^.prox;
end;
end;

Procedure ExemploB;
var str: string;
    p1: Produto;
begin
    InicProd := Produto.Create;
    InicProd.cod := 100;
    InicProd.desc := 'Televisão';
    InicProd.preco := 650;
    InicProd.prox := Produto.Create;
    p1 := InicProd.prox;
    p1.cod := 200;
    p1.desc := 'DVD';
    p1.preco := 1250;
    p1.prox := Produto.Create;
    p1 := p1.prox;
    p1.cod := 300;
    p1.desc := 'Som';
    p1.preco := 400;
    p1.prox := nil;
    //Percorrendo a lista
    p1 := InicProd;
    while (p1 <> nil) do
    begin
        showMessage(IntToStr(p1.cod)+' : '+p1.desc +
            ' R$ '+CurrToStr(p1.preco));
        p1 := p1.prox;
    end;
end;

procedure TForm1.ButtonExecutarAClick(Sender: TObject);
begin
    ExemploA;
end;

procedure TForm1.ButtonExecutarBClick(Sender: TObject);
begin
    ExemploB;
end;

procedure TForm1.ButtonSairClick(Sender: TObject);
begin
    Form1.Close;
end;

end.
```

Exercícios

1.01) Qual a saída produzida pelos programas abaixo.

- a) `procedure procA;`
 `var x,y: integer;`
 `begin`
 `x:=0; y:=2;`
 `while x < 10 do`
 `begin`
 `y := 2 * y; writeln(y);`
 `x := x + 2;`
 `end;`
 `end;`
- b) `Procedure procB;`
 `var a: integer;`
 `begin`
 `for a :=36 to 0 do`
 `begin`
 `writeln(a);`
 `a := a div 2;`
 `end;`
 `end;`
- c) `Procedure procC`
 `var k: integer;`
 `v: array[0..4] of integer;`
 `begin`
 `v[0]:=6; v[1]:=5; v[2]:=9; v[3]:=23; v[4]:=16;`
 `for k:=0 to 4 do`
 `begin`
 `writeln(v[k] mod 4);`
 `k := k + 1;`
 `end;`
 `end;`

1.02) Escreva a função `function contac(s: string; char c):integer` que conta o número de ocorrências do caracter `c` na string `s`.

Exemplo: para `s = 'abxdexfgxhixjk'` e `c = 'x'` a saída da função será 4.

1.03) Escreva o procedimento `procedure substr(s1,s2: string; n,tam:integer)` que copia a substring de `s1` com início na posição `n` e de tamanho `tam` na variável string `s2`. Exemplo: seja `s1 = "abcxyzwtdef"`, `n = 3` e `tam = 5` então `s2` deverá conter a string "xyzwt".

1.04) Elabore um programa que calcule as raízes de uma equação do 2º grau. O programa deve ler os valores de `a`, `b` e `c` e imprimir os valores de `x1` e `x2`.

1.05) Escreva um programa que leia 5 variáveis inteiras e imprima o valor da maior e da menor delas.

1.06) Elabore um programa que calcule a área de algumas figuras geométricas. O programa deve apresentar um menu com as seguintes opções:

- 1) Quadrado
- 2) Retângulo

- 3) Círculo
- 4) Trapézio
- 5) Sair

O cálculo das áreas deve ser feito por funções que recebem os dados necessários para o cálculo como parâmetro e retornam a área da figura. Desta forma, o programa deverá ter no mínimo 5 funções: uma função contendo o menu e mais as 4 funções para cálculo das áreas

Fórmulas das áreas:

área do quadrado: $A = \text{Lado} \times \text{Lado}$

área do retângulo: $A = \text{Base} \times \text{Altura} / 2$

área do círculo: $A = \text{Pi} \times \text{Raio} \times \text{Raio}$

área do trapézio: $A = (\text{BaseMaior} + \text{BaseMenor}) \times \text{Altura} / 2$

1.07) Faça um programa que leia uma frase e mostre na tela a frase com as palavras na ordem inversa. Exemplo:

Entrada: EU ADORO PROGRAMAR EM PASCAL

Saída: PASCAL EM PROGRAMAR ADORO EU

1.08) Elabore um programa em Pascal que:

- a) leia o tamanho de um vetor A e de um vetor B, ambos contendo dados do tipo real, e alocue espaço para os mesmos (alocação dinâmica).
- b) inicialize os vetores A e B.
- c) alocue espaço para um vetor C de real cujo tamanho é a soma dos tamanhos de A e de B
- d) inicialize o vetor C com os dados de A e com os dados de B, nesta ordem.

1.09) Escreva as instruções em Pascal utilizadas para definir um vetor capaz de armazenar 20 elementos de uma estrutura (record) com seguintes campos:

- RG (inteiro)
- Nome (string)
- End. (string)
- Peso (real)
- Aniversário (Data)

Obs.: criar uma estrutura (record) para os dados acima e outra para data.

1.10) Defina uma estrutura (class) para armazenar as fitas de uma locadora. Elabore uma programa para ler e imprimir um vetor de 10 elementos desta estrutura. Abaixo temos a descrição dos campos a serem armazenados.

- Cod (inteiro)
- Nome do Filme (string)
- Duração (real)
- Data de Empréstimo (Data)
- Valor a pagar (real);