Delphi

Criando Units

Jocélio Passos
joceliodpassos@bol.com.br

Criando Units

- **Unit**
- Arquivo fonte em Pascal não conectado a um formulário
- Módulo compilado separadamente
- Vantagens
  - Estruturar o Sistema
  - Reusabilidade
- Arquivo isolado permitindo que qualquer projeto use
- Você pode escrever suas próprias procedures, funcions, DLL´s e componentes

Criando Units

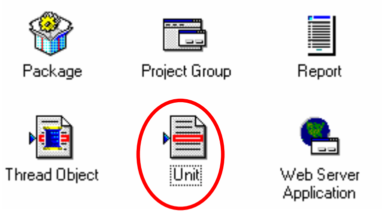
- Quando compilamos, executamos ou checamos a sintaxe de um projeto o compilador Delphi produz um **arquivo intermediário** para cada unit.
- Por default cada unit é gravada em um arquivo separado em formato binário como o mesmo nome da unit, mas com extensão
- DCU
- Delphi Compiled Unit
- Arquivos Unit armazenam procedures e functions.

Exemplo de Units

- **unit** Unit1; { unit identifier }
- **interface**
- **uses** SysUtils, Windows, Messages, Classes, Graphics, Controls, Forms, Dialogs; { uses clause }
- **type** TForm1 = class(TForm) { class declaration }
- private { Private declarations }
- public { Public declarations }
- end;
- var Form1: TForm1; { instance declaration }
- **implementation**
- **{$R *.DFM}** { compiler directive to link form file }
- end.

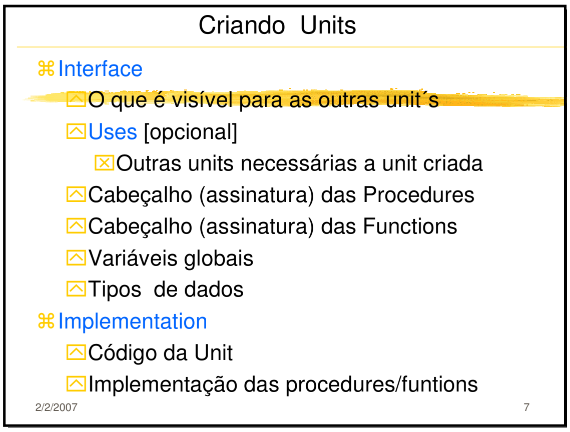
Criando Units

- Após abrir o projeto : UNIDADE.DPR
- Crie uma nova Unit
  - File, New, Other ..., Aba New, **Unit**



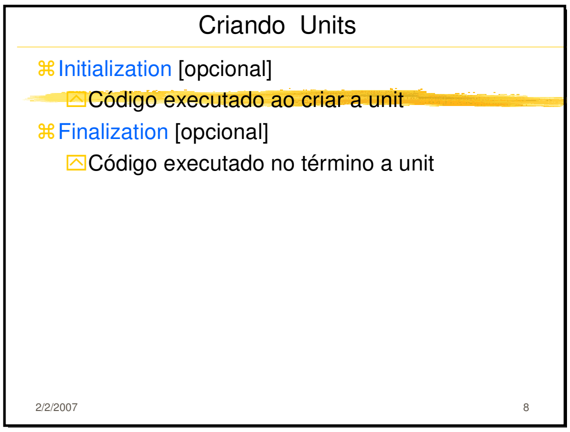
Criando Units

- **unit** Unit1;
- **interface**
- **implementation**
- **end.**



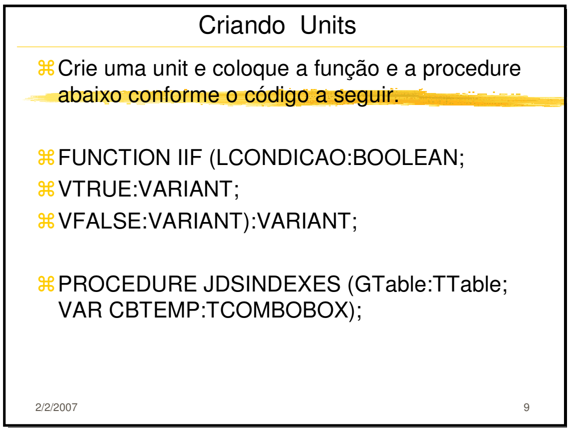
Criando Units

- Interface
  - O que é visível para as outras unit´s
  - Uses [opcional]
    - Outras units necessárias a unit criada
  - Cabeçalho (assinatura) das Procedures
  - Cabeçalho (assinatura) das Functions
  - Variáveis globais
  - Tipos de dados
- Implementation
  - Código da Unit
  - Implementação das procedures/funtions



Criando Units

- Initialization [opcional]
  - Código executado ao criar a unit
- Finalization [opcional]
  - Código executado no término a unit

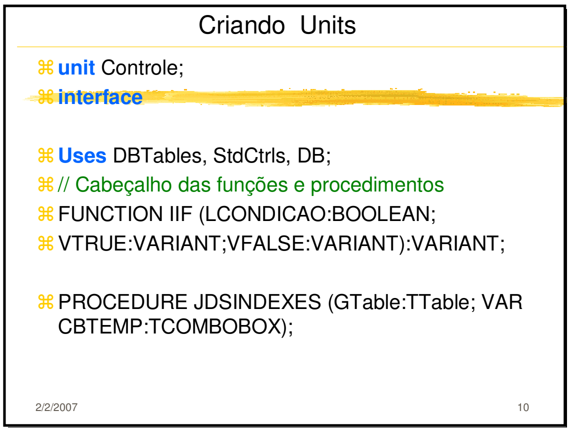


Criando Units

- Crie uma unit e coloque a função e a procedure abaixo conforme o código a seguir.

- FUNCTION IIF (LCONDICAO:BOOLEAN;
- VTRUE:VARIANT;
- VFALSE:VARIANT):VARIANT;

- PROCEDURE JDSINDEXES (GTable:TTable; VAR CBTEMP:TCOMBOBOX);

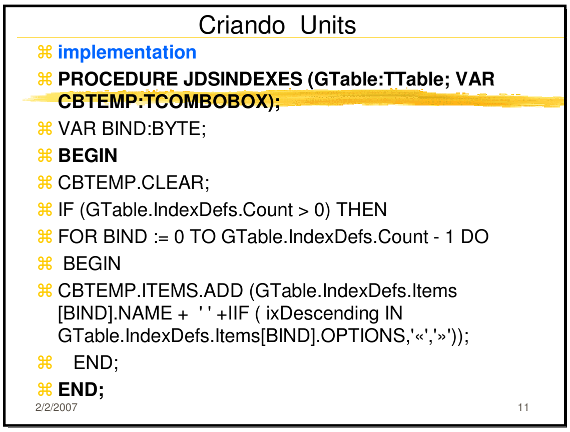


Criando Units

- **unit** Controle;
- **interface**

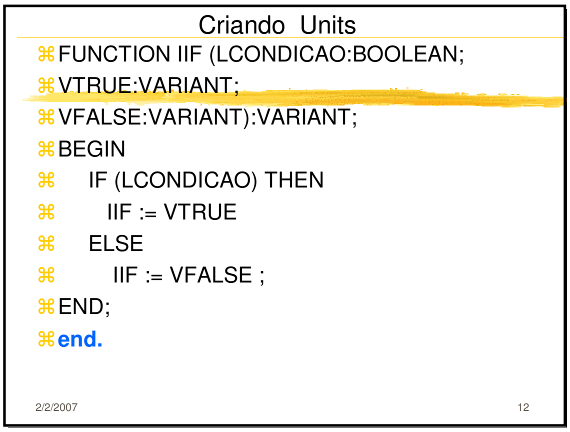
- **Uses** DBTables, StdCtrls, DB;
- // Cabeçalho das funções e procedimentos
- FUNCTION IIF (LCONDICAO:BOOLEAN;
- VTRUE:VARIANT;VFALSE:VARIANT):VARIANT;

- PROCEDURE JDSINDEXES (GTable:TTable; VAR CBTEMP:TCOMBOBOX);



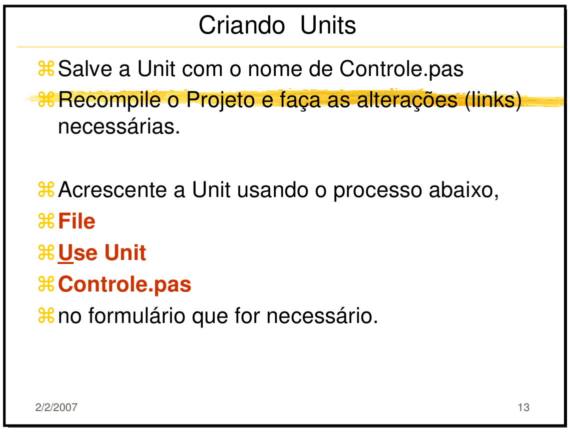
Criando Units

- **implementation**
- **PROCEDURE JDSINDEXES (GTable:TTable; VAR CBTEMP:TCOMBOBOX);**
- VAR BIND:BYTE;
- **BEGIN**
- CBTEMP.CLEAR;
- IF (GTable.IndexDefs.Count > 0) THEN
- FOR BIND := 0 TO GTable.IndexDefs.Count - 1 DO
- BEGIN
- CBTEMP.ITEMS.ADD (GTable.IndexDefs.Items [BIND].NAME + ' ' +IIF (ixDescending IN GTable.IndexDefs.Items[BIND].OPTIONS,'«','»'));
-    END;
- **END;**



Criando Units

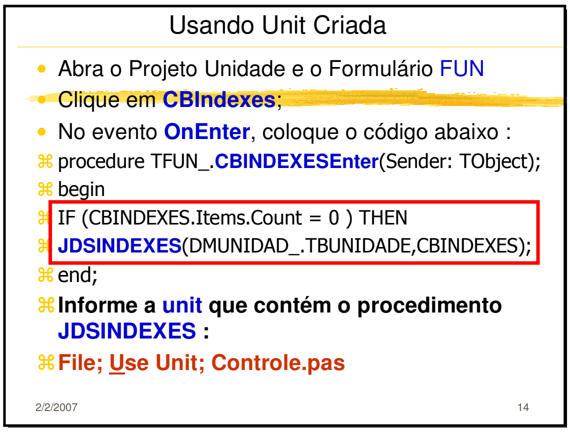
- FUNCTION IIF (LCONDICAO:BOOLEAN;
- VTRUE:VARIANT;
- VFALSE:VARIANT):VARIANT;
- BEGIN
-     IF (LCONDICAO) THEN
-        IIF := VTRUE
-     ELSE
-         IIF := VFALSE ;
- END;
- **end.**



Criando Units

- Salve a Unit com o nome de Controle.pas
- Recompile o Projeto e faça as alterações (links) necessárias.

- Acrescente a Unit usando o processo abaixo,
- **File**
- **Use Unit**
- **Controle.pas**
- no formulário que for necessário.

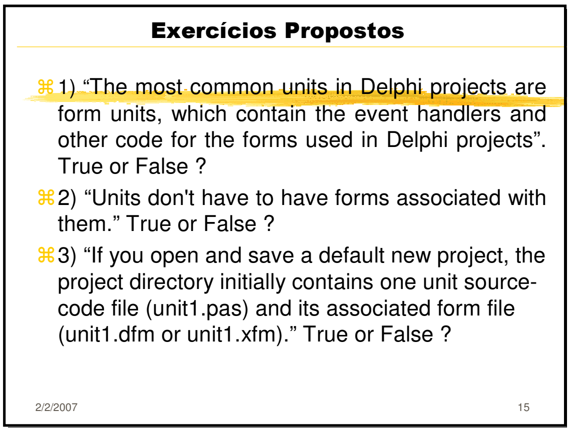


Usando Unit Criada

- Abra o Projeto Unidade e o Formulário FUN
- Clique em **CBIndexes**;
- No evento **OnEnter**, coloque o código abaixo :

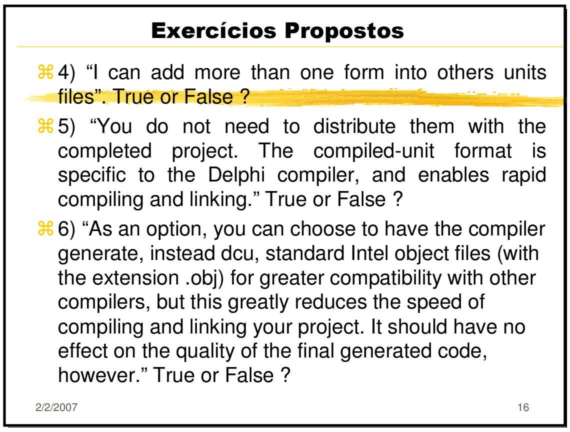
```
procedure TFUN_.CBINDEXESEnter(Sender: TObject);
begin
IF (CBINDEXES.Items.Count = 0 ) THEN
JDSINDEXES(DMUNIDAD_.TBUNIDADE,CBINDEXES);
end;
```

- **Informe a unit que contém o procedimento JDSINDEXES :**
- **File; Use Unit; Controle.pas**



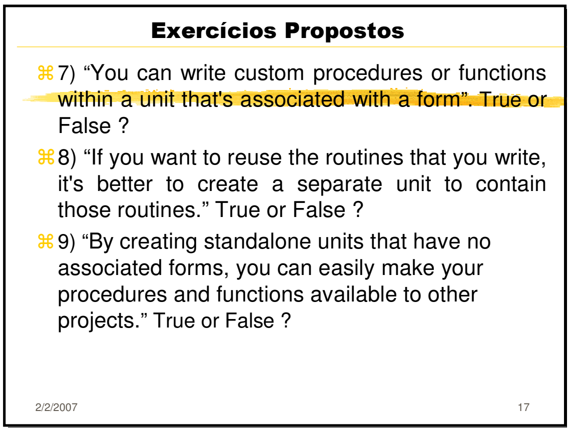
Exercícios Propostos

- 1) “The most common units in Delphi projects are form units, which contain the event handlers and other code for the forms used in Delphi projects”. True or False ?
- 2) “Units don't have to have forms associated with them.” True or False ?
- 3) “If you open and save a default new project, the project directory initially contains one unit source-code file (unit1.pas) and its associated form file (unit1.dfm or unit1.xfm).” True or False ?



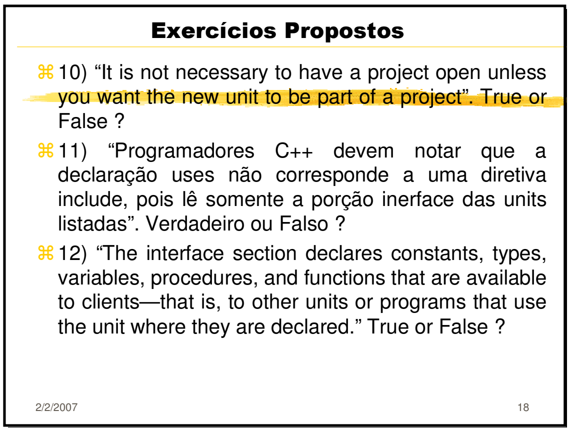
Exercícios Propostos

- 4) “I can add more than one form into others units files”. True or False ?
- 5) “You do not need to distribute them with the completed project. The compiled-unit format is specific to the Delphi compiler, and enables rapid compiling and linking.” True or False ?
- 6) “As an option, you can choose to have the compiler generate, instead dcu, standard Intel object files (with the extension .obj) for greater compatibility with other compilers, but this greatly reduces the speed of compiling and linking your project. It should have no effect on the quality of the final generated code, however.” True or False ?



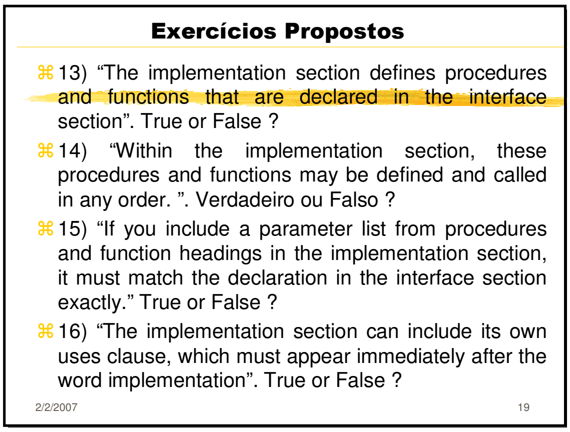
Exercícios Propostos

- 7) “You can write custom procedures or functions within a unit that's associated with a form”. True or False ?
- 8) “If you want to reuse the routines that you write, it's better to create a separate unit to contain those routines.” True or False ?
- 9) “By creating standalone units that have no associated forms, you can easily make your procedures and functions available to other projects.” True or False ?



Exercícios Propostos

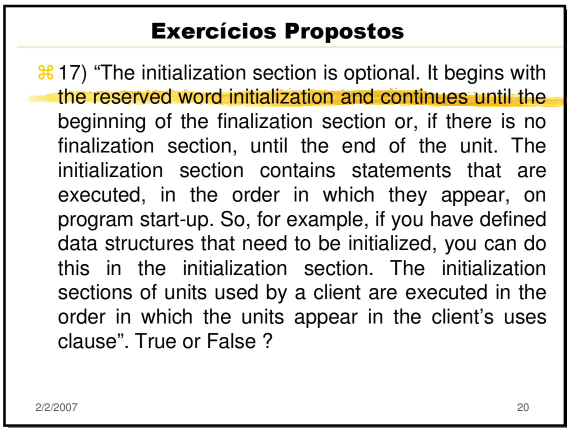
- 10) “It is not necessary to have a project open unless you want the new unit to be part of a project”. True or False ?
- 11) “Programadores C++ devem notar que a declaração uses não corresponde a uma diretiva include, pois lê somente a porção inerface das units listadas”. Verdadeiro ou Falso ?
- 12) “The interface section declares constants, types, variables, procedures, and functions that are available to clients—that is, to other units or programs that use the unit where they are declared.” True or False ?



Exercícios Propostos

- 13) “The implementation section defines procedures and functions that are declared in the interface section”. True or False ?
- 14) “Within the implementation section, these procedures and functions may be defined and called in any order. ”. Verdadeiro ou Falso ?
- 15) “If you include a parameter list from procedures and function headings in the implementation section, it must match the declaration in the interface section exactly.” True or False ?
- 16) “The implementation section can include its own uses clause, which must appear immediately after the word implementation”. True or False ?

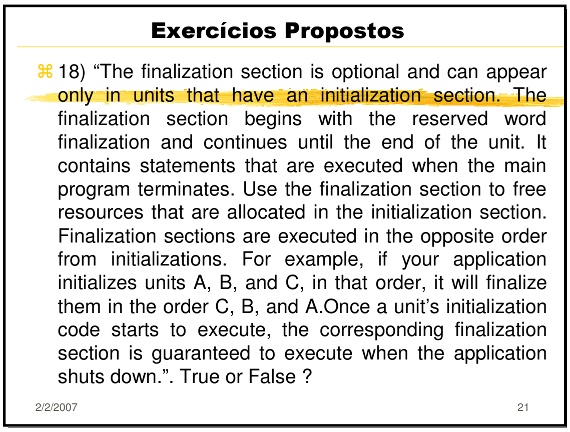
2/2/2007 19



Exercícios Propostos

- 17) “The initialization section is optional. It begins with the reserved word initialization and continues until the beginning of the finalization section or, if there is no finalization section, until the end of the unit. The initialization section contains statements that are executed, in the order in which they appear, on program start-up. So, for example, if you have defined data structures that need to be initialized, you can do this in the initialization section. The initialization sections of units used by a client are executed in the order in which the units appear in the client’s uses clause”. True or False ?

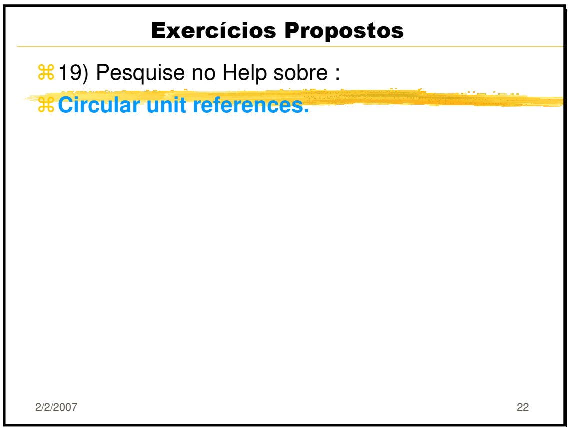
2/2/2007 20



Exercícios Propostos

- 18) “The finalization section is optional and can appear only in units that have an initialization section. The finalization section begins with the reserved word finalization and continues until the end of the unit. It contains statements that are executed when the main program terminates. Use the finalization section to free resources that are allocated in the initialization section. Finalization sections are executed in the opposite order from initializations. For example, if your application initializes units A, B, and C, in that order, it will finalize them in the order C, B, and A.Once a unit’s initialization code starts to execute, the corresponding finalization section is guaranteed to execute when the application shuts down.”. True or False ?

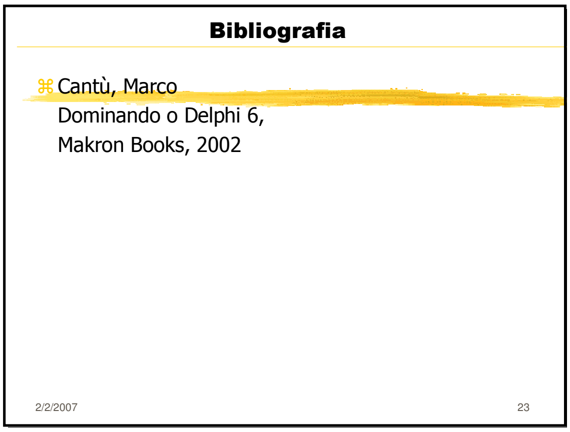
2/2/2007 21



Exercícios Propostos

- 19) Pesquise no Help sobre :
- **Circular unit references.**

2/2/2007 22



Bibliografia

- Cantù, Marco
  Dominando o Delphi 6,
  Makron Books, 2002

2/2/2007 23