

DIDASCALIA,C,254

Obiettivo

Come Funziona

Unit Heading

Unit Heading

Interface Part - Definizione

Interface Part - Uses Clause

Interface Part - Inizio blocco Type

Interface Part - Nome Componente

Interface Part - Tipo Componente

Interface Part - Livelli di accesso

Interface Part - Livelli di accesso

Interface Part - Livelli di accesso

Interface Part - Livelli di accesso

Interface Part - Fine blocco Type

Interface Part - Prototipo

Interface Part - Completa

Implementation Part - Definizione

Implementation Part - Procedure Register

Procedure Register - Comando

Fine Unit

Riassunto

Nota

Definizione variabili private.

Definizione dei metodi

Definizione delle nuove propriet...

Nota - Definizione delle nuove propriet...

Facoltativo - Definizione delle vecchie propriet...

Scrittura Metodo

Conclusioni

CODICE,C,254

NUI LU

	-1 0
	-1 0
	-1 0
unit MiaUnit1;	0 2
interface	
	2 2
uses	
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;	4 3
Type	
	7 1
TMioVCL1 = class(TComponent)	8 1
TMioVCL1 = class(TComponent)	-1 0
private	
{ Private declarations }	9 2
protected	
{ Protected declarations }	12 2
public	
{ Public declarations }	14 2
published	
{ Published declarations }	17 2
end;	23 2
procedure Register;	
	25 2
	-1 0
implementation	
	27 2
procedure Register;	
begin	
RegisterComponents('Samples', [TMioVCL1]);	
end;	34 5
RegisterComponents('Samples', [TMioVCL1]);	-1 1
end.	39 2
	-1 0

## Sheet1

	-1	0
N1,N2,R:Integer;	11	1
Procedure Execute;		
	16	1
Property Num1:Integer Read N1 Write N1;		
Property Num2:Integer Read N2 Write N2;		
Property Risultato:Integer Read R Write R;		
	19	3
	-1	0
Property Name;	22	1
Procedure TMioVCL1.Execute;		
Begin		
R:=N1+N2;		
End;		
	29	4
	-1	0

DESCRIZION,C,254

Creeremo un componente invisibile con tre propriet... ed un solo metodo.  
Le propriet... saranno Num1,Num2,Risultato;  
Il metodo sar... Execute.

Chiamando il metodo Execute, viene aggiornato il valore della propriet... Risultato dato dalla somma di Num1 e Num2.

Specifica il nome della unit sia come file fisico su disco che come riferimento per il comando 'Uses'. Tale nome deve essere u

Inseriamo quindi il nome della unit, preceduta dalla parola riservata 'Unit'. In questo esempio il nome Š 'MiaUnit1'.

Iniziamo a scrivere l'Interface part della unit. Come gi... sappiamo, questa sezione specifica le parti visibili e quindi utilizzabili

Per il momento nell'uses clause inseriremo solo le unit standard necessarie per la creazione del nostro componente.

Ora Š necessario definire il tipo di componente che intendiamo creare. Questa parte inizia con la parola chiave 'Type' e termin  
Definiamo il nome ed il tipo di oggetto che intendiamo creare. Il nome pu• essere qualsiasi, in questo caso Š 'TMioVCL1'. Se s  
Definiamo poi il tipo di oggetto che intendiamo creare in questo modo :  
NomeComponente = Class(TAntenato)  
TAntenato in questo caso corrisponde a TComponent.

Definiamo il livello di accesso ai dati pi— restrittivo. Tutto ci• che verr... dichiarato nella sezione 'Private', sar... accessibile so

Tutto ci• che verr... dichiarato nella sezione 'Protected', sar... accessibile solo all'interno della Unit e potr... essere 'ereditato' e

Come ricorderete il terzo livello di accesso ai dati Š il tipo 'Public'. Quello che viene definito in questa sezione Š accessibile a

Infine apriamo la sezione 'Published' del componente. Qui verranno definite le nuove propriet... e gli eventi del componente.

Come avevamo detto, concludiamo il blocco Type con la parola chiave 'End;'

Definiamo poi il prototipo della procedura che ci permetter... di registrare il nostro nuovo componente nella ComponentPalette  
A questo punto la struttura dell'Interface part della unit Š stata definita. Continueremo con la definizione della sezione 'Impleme

L'Implementation part della unit ha inizio con la parola riservata 'Implementation'. Termina con la parola riservata 'End.', che in

Definiamo la procedura 'Register'. Questa procedura verr... automaticamente eseguita al momento dell'installazione del comp

La procedura 'Register' prevede almeno 1 chiamata alla procedura 'RegisterComponents'. Come parametri imposteremo la se

Come al solito terminiamo la unit con la parola riservata 'End.'

Abbiamo creato la struttura generale della unit componente. Da notare che fino ad ora abbiamo specificato il nome dalla Unit,

## Sheet1

Avremmo ottenuto lo stesso risultato utilizzando il Component Expert di Delphi ed impostando le seguenti propriet...:

Class Name= TMioVCL1

Ancestor Class= TComponent

Palette Page= Samples

Il nome della unit verrebbe impostato automaticamente da Delphi.

Ora iniziamo con la definizione vera e propria delle nuove caratteristiche del componente. Per questo esempio avremo bisogno

Come avevamo detto in precedenza, definiamo il prototipo del metodo 'Execute' nella sezione 'Public' del componente.

Infine definiamo le nuove propriet... del componente. Queste verranno visualizzate automaticamente nell'Object Inspector di Delphi.  
Property NomeVariabile:TipoVariabile Read VariabileInterna Write VariabileInterna;

La sintassi utilizzata in questo caso è corretta ma non completa, vedremo in seguito le altre opzioni.

È possibile ridichiarare le propriet... ereditate dal componente Antenato. Queste propriet... verrebbero comunque visualizzate

Scriviamo in fine la procedura Execute, come se fosse una qualsiasi procedura. A questo punto il nostro primo componente è pronto.  
Abbiamo creato un VCL perfettamente funzionante, ma forse un po' 'scomodo' da utilizzare. Vediamo nella Lezione1b alcune soluzioni