

2011年计算机等级考试二级Delphi辅导讲义：Delphi自定义部件开发计算机二级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/646/2021_2022_2011_E5_B9_B4_E8_AE_A1_c97_646276.htm 导读：在本章中将阐述如何为Delphi应用程序编写部件。>>>点击查看此系列辅导讲义

汇总第十九章 Delphi自定义部件开发

Delphi除了支持使用可视化部件所见即所得地建立应用程序外，还支持为开发应用而设计自己的部件。在本章中将阐述如何为Delphi应用程序编写部件。这一章将达到两个目的：教你如何自定义部件

使你的部件成为Delphi环境的有机组合部分

19.1 Delphi部件原理

19.1.1 什么是部件

部件是Delphi应用程序的程序构件。尽管大多数部件代表用户界面的可见元素，但部件也可以是程序中的不可见元素，如数据库部件。为弄清什么是部件可以从三个方面来考察它：功能定义、技术定义和经验定义。

1. 部件的功能定义 从最终用户角度，部件是在Component Palette上选择的，并在窗体设计窗口和代码窗口中操作的元素。从部件编写者角度，部件是代码中的对象。在编写部件之前，你应用相当熟悉已有的Delphi部件，这样才能使你的部件适合用户的需要。编写部件的目标之一是使部件尽可能的类似其它部件。
2. 部件的技术定义 从最简单的角度看，部件是任何从TComponent继承的对象。TComponent定义了所有部件必须有的、最基本的行为。例如，出现在Component Palette上和窗体设计窗口中编辑的功能。但是TComponent并不知如何处理你的部件的具体功能，因此，你必须自己描述它。
3. 部件编写者自己的定义。在实际编程中，部件是能

插入Delphi开发环境的任何元素。它可能具有程序的各种复杂性。简而言之，只要能融入部件框架，部件就是你用代码编写的一切。部件定义只是接口描述，本章将详细阐述部件框架，说明部件的有限性，正如说明编程的有限性。本章不准备教你用所给语言编写每一种部件，只能告诉编定代码的方法和怎样使部件融入Delphi环境。

19.1.2 编写部件的不同之处

在Delphi环境中建立部件和在应用程序中使用部件有三个重要差别：

- 编写部件的过程是非可视化的
- 编写部件需要更深入的关于对象的知识
- 编写部件需要遵循更多的规则

1. 编写部件是非可视化的 编写部件与建立Delphi应用最明显的区别是部件编写完全以代码的形式进行，即非可视化的。因为Delphi应用的可视化设计需要已完成的部件，而建立这些部件就需要用Object Pascal 代码编写。虽然你无法使用可视化工具来建立部件，但你能运用 Delphi开发环境的所有编程特性如代码编辑器、集成化调试和对象浏览。

2. 编写部件需要更深的有关对象的知识 除了非可视化编程之外，建立部件和使用它们的最大区别是：当建立新部件时，需要从已存部件中继承产生一个新对象类型，并增加新的属性和方法。另一方面，部件使用者，在建立Delphi应用时，只是使用已有部件。在设计阶段通过改变部件属性和描述响应事件的方法来定制它们的行为。当继承产生一个新对象时，你有权访问祖先对象中对最终用户不可见的部分。这些部分被称为protected界面的。在很大部分的实现上，后代对象也需要调用他们的祖先对象的方法，因此，编写部件者应相当熟悉面向对象编程特性。

3. 编写部件要遵循更多的规则 编写部件过程比可视化应用生成采用更传统的编程方法，与使用已有

部件相比，有更多的规则要遵循。在开始编写自己的部件之前，最重要的事莫过于熟练应用Delphi自带的部件，以得到对命名规则以及部件用户所期望功能等的直观认识。部件用户期望部件做到的最重要的事情莫过于他们在任何时候能对部件做任何事。编写满足这些期望的部件并不难，只要预先想到和遵循规则。

19.1.3 建立部件过程概略 简而言之，建立自定义部件的过程包含下列几步：
建立包含新部件的库单元
从已有部件类型中继承得到新的部件类型
增加属性、方法和事件
用Delphi注册部件
为部件的属性方法和事件建立Help文件
如果完成这些工作，完整的部件包含下列4个文件
编译的库单元 (.DCU文件)
选择板位图 (.DCR文件)
Help文件 (.HLP文件)
Help-keyword文件 (.KWF文件)

19.2 Delphi部件编程方法 19.2.1 Delphi部件编程概述 19.2.1.1

Delphi可视部件类库 Delphi的部件都是可视部件类库 (VCL) 的对象继承树的一部分，下面列出组成VCL的对象的关系

。TComponent是VCL中每一个部件的共同祖先

。TComponent提供了Delphi部件正常工作的最基本的属性和事件。库中的各条分支提供了其它的更专一的功能。当建立部件时，通过从对象树中已有的对象继承获得新对象，并将其加入VCL中。

19.2.1.2 建立部件的起点 部件是你在设计时想操作的任意程序元素。建立新部件意味着从已有类型中继承得到新的部件对象类。建立新部件的主要途径如下：
修改已有的控制
建立原始控制
建立图形控制
建立Windows控制的子类
建立非可视部件
下表列出了不同建立途径的起始类

表19.1 定义部件的起始点

途径起始

类

修改已有部件 任何已有部件，如TButton、TListBox
或抽象部件对象如TCustomListBox 建立原始控制
TCustomControl 建立图形控制 TGraphicControl 建立窗口控制的
子类 TWinControl 建立非可视部件 TComponent

也

可以继承非部件的其它对象，但无法在窗体设计窗口中操作它们。Delphi包括许多这种对象，如TINIFile、TFont等。

1. 修改已有控制 建立部件的最简单的方法是继承一个已有的、可用的部件并定制它。可以从Delphi提供的任何部件中继承。例如，可以改变标准控制的缺省属性值，如TButton。有些控制，如Listbox和Grid等有许多相同变量，在这种情况下，Delphi提供了抽象控制类型，从该类型出发可定制出许多的类型。例如，你也许想建立TListBox的特殊类型，这种部件没有标准TListBox的某些属性，你不能将属性从一个祖先类型中移去，因此你需要从比TListBox更高层次的部件继承。例如TCustomListBox，该部件实现了TCustomListBox的所有属性但没有公布(Publishing)它们。当从一个诸如TCustomListBox的抽象类中继承时，你公布那些你想使之可获得的属性而让其它的保护起来(protected)。
2. 建立原始控制 标准控制是在运行时可见的。这些标准控制都从TWinControl，继承来的，当你建立原始控制时，你使用TCustomControl作为起始点。标准控制的关键特征是它具有窗口句柄，句柄保存在属性Handle中，这种控制：
 - 能接受输入焦点
 - 能将句柄传送给Windows API函数如果控制不需要接受输入焦点，你可把它做成图形控制，这可能节省系统资源。
3. 建立图形控制

图形控制非常类似定制的控制，但它们没有窗口句柄，因此不占有系统资源。对图形控制最大的限制是它们不能接收输入焦点。你需要从TGraphicControl继承，它提供了作图的Canvas和能处理WM_PAINT消息，你需要覆盖Paint方法。

4. 继承窗口控制 Windows中有一种称之为窗口类的概念，类似于面向对象的对象和类的概念。窗口类是Windows中相同窗口或控制的不同实例之间共享的信息集合。当你用传统的Windows编程方法创建一种新的控制，你要定义一个新的窗口类，并在Windows中注册。你也能基于已有的窗口类创建新的窗口类。这就称为从窗口类继承。在传统的Windows编程中，如果你想建立客户化的控制，你就必须将其做在动态链接库里，就象标准Windows控制，并且提供一个访问界面。使用Delphi，你能创建一个部件包装在已有窗口类之上。如果你已有客户化控制的库，并想使其运行在你的Delphi应用中，那你就能创建一个使你能使用已有控制和获得新的控制的部件。在库单元StdCtrls中有许多这样的例子。

5. 建立非可视化的部件 抽象对象类型TComponent是所有部件的基础类型。从TComponent直接继承所创建的部件就是非可视化部件。你编写的大多数部件都是可视控制。TComponent定义了部件在FormDesigner中所需的基本的属性和方法。因此，从TComponent继承来的任何部件都具备设计能力。非可视部件相当少，主要用它们作为非可视程序单元（如数据库单元）和对话框的界面。

19.2.1.3 建立新部件的方法 建立新部件的方法有两种：
手工建立部件 使用Component Expert 一旦完成建立后，就得到所需的最小功能单位的部件，并可以安装在Component Palette上。安装完后，你就能将新部件放置在

窗体窗口，并可在设计阶段和运行阶段进行测试。你还能为部件增加新的特征、更新选择板、重新测试。

1. 手工创建部件

显然创建部件最容易的方法是使用Component Expert。然而，你也能通过手工来完成相同步骤。手工创建部件需要下列三步：

- 创建新的库单元
- 继承一个部件对象
- 注册部件

创建新的库单元 库单元是Object Pascal代码的独立编译单位。每一个窗体有自己的库单元。大多数部件（在逻辑上是一组）也有自己的库单元。当你建立部件时，你可以为部件创建一个库单元，也可将新的部件加在已有的库单元中。

为部件创建库单元，可选择File/New...，在New Items对话框中选择Unit，Delphi将创建一个新文件，并在代码编辑器中打开它。在已有库单元中增加部件，只须选择File/OPen为已有库单元选择源代码。在该库单元中只能包含部件代码，如果该库单元中有一个窗体，将产生错误。

继承一个部件对象 每个部件都是TComponent的后代对象。也可从TControl、TGraphicControl等继承。为继承一个部件对象，要将对象类型声明加在库单元的interface部分。例如，建立一个最简单的从TComponent直接继承非可视的部件，将下列的类型定义加在部件单元的interface部分。

```
type  
TNewComponent=class(TComponent) ..... end.
```

现在你能注册TNewComponent。但是新部件与TComponent没什么不同，你只创建了自己部件的框架。

注册部件 注册部件是为了告诉Delphi什么部件被加入部件库和加入Component Palette的哪一页。为了注册一个部件：在部件单元的interface部分增加一个Register过程。Register不带任何参数，因此声明很简单：

```
procedure Register.
```

如果你在已有部件的库单元中增加部

件，因为已有Register过程，因此不须要修改声明。在库单位的implementation部件编写Register过程为每一个你想注册的部件调用过程RegisterComponents，过程RegisterComponents带两个参数：Component Palette的页名和部件类型集。例如，注册名为TNewComponent的部件，并将其置于Component Palette的Samples页，在程序中使用下列过程：
procedure Register. begin RegisterComponents(Samples, [TNewComponent]). end. 一旦注册完毕，Delphi自动将部件图标显示在Component Palette上。

2. 使用Component Expert（部件专家）你能使用Component Expert创建新部件。使用Component Expert简化了创建新部件最初阶段的工作，因为你只需描述三件事：新部件的名字 祖先类型 新部件要加入的Component Palette页名 Component Expert执行了手工方式的相同工作：建立新的库单元 继承得到新部件对象 注册部件 但Component Expert不能在已有单元中增加部件。可选择File/New...，在New Items对话框中选择Component，就打开Component Expert对话框。填满Component Expert对话框的每一个域后，选择OK。Delphi建立包括新部件和Register过程的库单元，并自动增加uses语句。你应该立刻保存库单元，并给予其有意义的名字。

19.2.1.4. 测试未安装的部件 在将新部件安装在Component Palette之前就能测试部件运行时的动作。这对于调试新部件特别有用，而且还能用同样的技术测试任意部件，无论该部件是否出现在Component Palette上。从本质上说，你通过模仿用户将部件放置在窗体中的Delphi的动作来测试一个未安装的部件。可按下列步骤来测试未安装的部件

1. 在窗体单元

的uses语句中加入部件所在单元的名字 2. 在窗体中增加一个对象域来表示部件 这是自己增加部件和Delphi增加部件的方法的主要不同点。 你将对象域加在窗体类型声明底部的public部分。 Delphi则会将对象域加在底部声明的上面。 你不能将域加在Delphi管理的窗体类型的声明的上部。 在这一部分声明的对象域将相应存储在DFM文件中。 增加不在窗体中存在的部件名将产生DFM文件无效的错误。 3. 附上窗体的OnCreate事件处理过程 4. 在窗体的OnCreate处理过程中构造该部件 当调用部件的构造过程时，必须传递Owner参数（由Owner负责析构该部件）一般说来总是将Self作为Owner的传入参数。 在OnCreate中，Self是指窗体。 5. 给Component的Parent属性赋值 设置Parent属性往往是构造部件后要做的第一件事时。 Parent在形式上包含部件，一般来说Parent是窗体或者GoupBox、 Panel。 通常给Parent赋与Self，即窗体。 在设置部件的其它属性之前最好先给Parent赋值。 6. 按需要给部件的其它属性赋值 假设你想测试名为TNewComponent类型的新部件，库单元名为NewTest。 窗体库单元应该是这样的；

```
unit Unit1. interface uses SysUtils, Windows, Messages, Classes, Graphics, Controls, Forms, Dialogs, Newtest. type TForm1 = class(TForm) procedure FormCreate(Sender: TObject). private { private申明 } public { public申明 } NewComponent: TNewComponent. end. var Form1: TForm1. implementation {$R *.DFM } procedure TForm1.FormCreate ( Sender: TObject ) . begin NewComponent := TNewComponent.Create ( Self ). NewCompanent.Parent := Self. NewCompanent.Left := 12. end. end.
```

100Test 下载频道开通，各类考试题目直接下载。详细请

访问 www.100test.com