



FACULTAD DE INFORMÁTICA

Aplicaciones gráficas en C++ Builder

Baltasar Fernández Manjón

<http://www.fdi.ucm.es/profesor/balta/>



Nombres de ficheros en C++ Builder

- Ficheros de proyecto
 - NombreProyecto.bpr ; NombreProyecto.cpp ; NombreProyecto.res
- Ficheros de formulario
 - NombreUnidad.dfm ; NombreUnidad.cpp ; NombreUnidad.h
- Ficheros de paquet
 - VCL50.bpl (paquete con la biblioteca de Borland de componentes visuales)
- Ficheros con las opciones de escritorio
 - NombreProyecto.dsk
- Ficheros de backup - *.~cp ; *.~bp
- Turbo Debugger *.TDW

Laboratorio de programación II (Facultad de Informática)

- 1



VCL como colección de objetos

- Los componentes de la biblioteca VCL son objetos que se han creado en Delphi
 - Se almacenan en el monton (heap) y se referencian mediante punteros
 - (MiBoton->caption = "Salir")
- Son objetos persistentes (almacenables) que tienen interfaces que permiten su configuración o modificación tanto en diseño como en ejecución
- Esto es posible mediante el mecanismo de propiedades
 - Permite su configuración en diseño
 - Permite crear/asignar manejadores de eventos tanto en diseño como en ejecución

Laboratorio de programación II (Facultad de Informática)

- 2



VCL

- La biblioteca está formada por objetos que tienen
 - Propiedades
 - Métodos
 - Eventos
- Todos las clases de objetos comienzan por T para indicar tipo de datos

Laboratorio de programación II (Facultad de Informática)

- 3

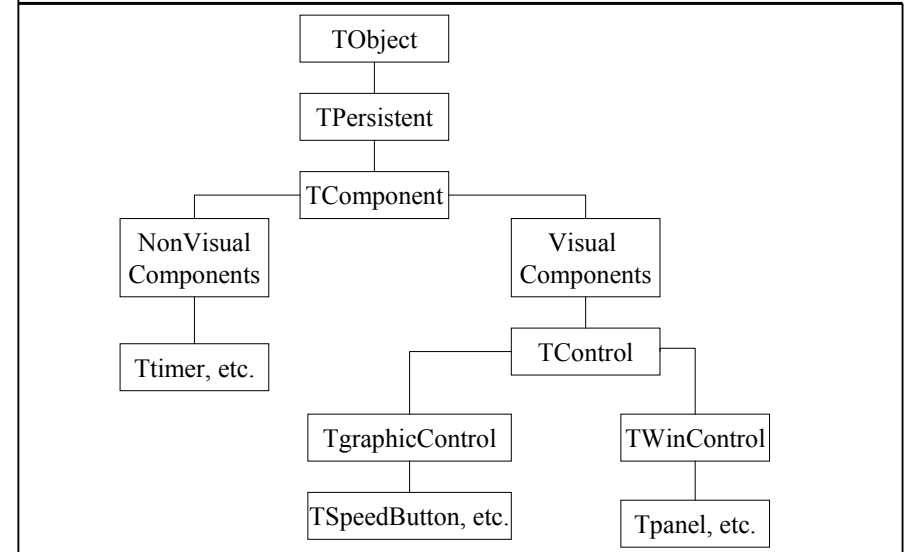


VCL introduce nuevos aspectos

- Nuevas palabras reservadas
 - `__classid`
 - `__closure`
 - `__fastcall`
 - `__property`
 - `__published`
- Nuevos tipos de datos
 - E.g. `AnsiString`, `Set`, `OPENARRAY`
- Un mecanismo de propiedades de los objetos
 - Permite el acceso controlado a los datos internos de los objetos
 - Formaliza la funcionalidad de acceso/mutación
- Mejora la gestión de excepciones



Jerarquía de objetos en VCL



Jerarquía de objetos en VCL

- TObject
 - Objeto base de la jerarquía de objetos gráficos
 - No hay herencia múltiple
 - Establece las características y el comportamiento común a todos los objetos gráficos
 - respuesta a creación/destrucción de objetos
 - identificación del tipo en tiempo de ejecución (incluido identificación de ancestros)
 - soporte de gestión de mensajes
- TPersistent
 - Es una clase abstracta que define los métodos que permiten almacenar de modo permanente los objetos en un flujo (stream)
 - No define propiedades o eventos especiales pero define ciertos métodos para el Component Writer
 - `LoadFromStream`, `LoadFromFile`
 - `SaveToStream`, `SaveToFile`



Jerarquía de VCL

- TComponent
 - Padre de todos los componentes gráficos tanto visuales como no visuales
 - Sus propiedades son almacenables y pueden ser manipuladas durante el diseño de la aplicación mediante el inspector de objetos (Object Inspector)
 - TComponent puede contener a otros componentes
- Propiedades más importantes
 - Name
 - Identificador del componente
 - Components
 - Lista de los otros componentes que posee un componente
 - ComponentCount
 - Determina el número de componente que se poseen
 - Owner
 - Determina la destrucción de objetos
- En los proyectos normales la mayoría de los componentes pertenecen a un formulario y los formularios a la aplicación



Jerarquía VCL: TControl

- TControl
 - Define las propiedades, métodos y eventos comunes a todos los componentes visuales con los que habitualmente interactúa el usuario
- Un objeto TControl tiene la capacidad de mostrarse a sí mismo
 - Tiene propiedades asociadas a su tamaño y posición: Top, Left, Width, Height, ...
 - Tiene propiedades relacionadas con su apariencia y accesibilidad: Visible, Enabled, Color, Hint
 - La propiedad Font permite establecer la fuente utilizada en dicho control
 - El texto relativo al control se establece mediante las propiedades Text y Caption (título)
 - La propiedad Parent
 - Refiere al control que lo contiene
 - Proporciona un contexto para posicionamiento
 - Proporciona una ventana en la que se dibuja
 - Parent y Owner tienen distintas responsabilidades. Un componente siempre es una parte de su Parent (e.g. Un botón de herramienta en una barra de herramientas)



Jerarquía VCL: TControl

- TControl contiene los eventos (e.g. de ratón, selección y arrastre) necesarios para controles visuales
 - Estos eventos son: OnClick, OnDbClick, OnMouseDown, OnMouseMove, OnMouseUp, OnDragOver, OnDragDrop, and OnEndDrag



Jerarquía VCL: TWinControl

- TWinControl
 - Encapsula los controles estándar y el manejador de una ventana
 - La funcionalidad de estos controles se puede manipular mediante las propiedades y métodos proporcionados sin necesidad de utilizar las funciones del API Win32 que proporciona Windows.
 - Las tres características básicas de los controles TWinControl son:
 - Tienen un manejador de ventana
 - Reciben el foco de la interacción
 - Pueden contener a otros controles
- Por tanto tienen propiedades, funciones y eventos relacionados con dichas características



Jerarquía VCL: TGraphicControl

- TGraphicControl
 - Un control gráfico que no tiene un manejador de ventana y no puede recibir el foco de la interacción.
 - Tampoco puede contener a (o ser padre de) otros controles
 - Proporciona un mecanismo para el dibujo personalizado del componente
 - El método virtual Paint que se reescribe en sus descendientes
 - Su propiedad principal es el objeto Canvas
 - Se utilizan cuando se desea mostrar algo en el formulario pero no son necesarias todas las funcionalidades de las ventanas normales
 - Las ventajas son que usan menos recursos y se dibujan más rápidamente
 - Los objetos TGraphicControl si pueden responder a eventos de ratón y por lo tanto tienen manejadores o gestores de eventos



Funciones de utilidad

- `ShowMessage(AnsiString("Hola que tal ") + otraCadena + ".");`
 - Muestra un cuadro de mensaje con un botón de OK
- `AnsiString cadenaEntrada = InputBox("Titulo del cuadro de diálogo", "Solicitud al usuario", "cadena por defecto");`
 - Llama a `InputBox` para crear un cuadro de diálogo con un campo editable en el que el usuario puede introducir una cadena de caracteres. El cuadro tiene dos botones uno de OK y otro de cancelación.
 - Si el usuario cancela se devuelve la cadena por defecto. Si el usuario pulsa el botón OK se devuelve la cadena del campo editable.



TApplication

- Encapsula una aplicación Windows
 - Simplifica aspectos de la programación de aplicaciones como es la gestión de mensajes, el formulario que actúa como ventana principal, disposición de ayuda, etc
- Todas las aplicaciones de *C++ Builder* tienen un puntero a un objeto *TApplication* denominado *Application*, que se crea automáticamente
 - Algunas de sus propiedades pueden establecerse en tiempo de diseño mediante `Forms` o `Application` después de seleccionar `Project | Options`



TForm

- Representa los formularios en VCL
- Los formularios son los elementos que se emplean como ventanas
 - Ventanas principales de la aplicación
 - Cuadros de diálogo
 - Ventanas secundarias