Программирование и основы алгоритмизации

Лекция 10

Основные классы и визуальные компоненты CBuilder

Основные классы

- TObject – объект
- TList - список
- TComponent – визуальный компонент
- TControl – Компонент для управления
- TStrings – строки (список строк)
- TCanvas – класс для рисования
- TGraphic – базовый класс для описания изображения
- TPicture - изображение
- TBitmap – изображение в формате BMP

Класс TObject (Help)

TObject is the ultimate ancestor of all VCL objects and components.

Unit

Systobj

Description

TObject encapsulates fundamental behavior common to VCL objects by introducing methods that

respond when object instances are created or destroyed.

return class-type and instance information on an object and runtime type information (RTTI) about its published properties.

support message handling.

Use TObject as an immediate base class when declaring simple objects that do not need to persist (are not saved and reloaded) and that do not need to be assigned to other objects.

Much of the capability of VCL objects is established by methods that TObject introduces. Many of these methods are used internally by the development environment and are not intended for users to call directly. Others are overridden in descendant objects that have more complex behavior.

Note: While TObject has no pure virtual methods, objects of this type are not normally instantiated.

Методы класса TObject

~TObject деструктор
AfterConstruction
BeforeDestruction
ClassInfo
ClassName
ClassNameIs
ClassParent
ClassType
CleanupInstance
DefaultHandler
Dispatch
FieldAddress
Free вызывает деструктор
FreeInstance
GetInterface
GetInterfaceEntry
GetInterfaceTable
InheritsFrom
InitInstance
InstanceSize
MethodAddress
MethodName
NewInstance
SafeCallException
TObject конструктор

Класс TList (Help)

TList stores an array of pointers.

Unit

Classes

Description

TList, which stores an array of pointers, is often used to maintain lists of objects. TList introduces properties and methods to

- Add or delete the objects in the list.
- Rearrange the objects in the list.
- Locate and access objects in the list.
- Sort the objects in the list.

Свойства класс TList

Capacity - ограничение на длину списка

Count - количество элементов в списке

Items - элементы списка

List - массив указателей на элементы

Методы класса TList

~TList - деструктор
Add - добавить элемент
Assign - копировать элемнты из одного списка в другой
Clear - почистить список
Delete - удалить элемент
Error - задать реакцию на ошибку при работе со списком
Exchange - обмен двух элементов значениями (перестановка)
Expand - расширение допустимого объема списка
Extract - исключение элемента из списка
First - получить указатель на 1-ый элемент
IndexOf - получить индекс элемента по его адресу
Insert - включить новый элемент
Last - получить указатель на последний элемент
Move - переместить элемент на другое место с заданным индексом
Pack - собирает в начале списка не пустые элементы
Remove - удаляет объект из списка
Sort - сортирует объекты в списке по именам
TList - конструктор

Пример с методом Add

```
typedef struct AList

{
  int I;
  char C;
} TAList;

typedef TAList* PAList;

void __fastcall TForm1::Button1Click(TObject
      *Sender)

{
  PAList AStruct;
  TList *MyList = new TList;
  // fill the TList
  AStruct = new TAList;
  AStruct->I = 100;
  AStruct->C = 'Z';
  MyList->Add(AStruct);
  AStruct = new TAList;
  AStruct->I = 100;
  AStruct->C = 'X';
  MyList->Add(AStruct);
  MyList->Add(AStruct);

  // Go through the list, writing the elements to the
  // canvas of a paintbox component.
  int Y = 10; // position on canvas
  for (int i = 0; i < MyList->Count; i++)

  {
    AStruct = (PAList) MyList->Items[i];
    PaintBox1->Canvas->TextOut(10, Y,
      IntToStr(AStruct->I));
    Y += 30;  // Increment Y Value again
    PaintBox1->Canvas->TextOut(10, Y, AStruct->C);
    Y += 30;  //Increment Y Value
  }

  // Clean up – must free memory for the items as well
      as the list
  for (int i = 0; i < MyList->Count; i++)
  {
    AStruct = (PAList) MyList->Items[i];
    delete AStruct;
  }
  delete MyList;

}
```

Пример с методом IndexOf

```
// Добавление элемента в сисок, если его
// там нет еще
if (MyList->IndexOf(TheObject)== -1)
  MyList->Add(TheObject);
```

Пример с методом Pack

```
void __fastcall TForm1::FormCreate(TObject *Sender)

{

  TList *MyList = new TList;

  MyList->Add("A string"); //Add a string

  MyList->Add(""); // Add an empty string
  MyList->Add(NULL); //Add NULL
  MyList->Add(""); // Add another empty string
  Edit1->Text = IntToStr(MyList->Count); // Put count into Edit1
  MyList->Pack();                        // Pack the list.
  Edit2->Text = IntToStr(MyList->Count); // Put count into Edit2
  delete MyList;  // Free memory for list
}
```

Пример с методом Remove

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
  TList *pList = new TList();
  AnsiString TheObject = "This is an object."
  try
  {
    pList->Add(TheObject); // add AnsiString instance to list
    MessageDlg("The list has " + IntToStr(pList->Count) + "objects",
              mtInformation, TMsgDlgButtons() << mbOK, 0);
    pList->Remove(TheObject);
    MessageDlg("The list has " + IntToStr(pList->Count) + "objects",
              mtInformation, TMsgDlgButtons() << mbOK, 0);

  }
  __finally
  {
    delete pList;
  }
}
```

Класс TComponent (Help)

TComponent is the common ancestor of all components that can appear in the form designer.

Unit

Classes

Description

Components are persistent objects that have the following capabilities:

The ability to appear on Component palette and be manipulated in the form designer.

The ability to own and manage other components.

Enhanced streaming and filing capabilities.

The ability to be converted into an ActiveX control or other COM object by wizards on the ActiveX page of the New Objects dialog.

Do not create instances of TComponent. Use TComponent as a base class when declaring non-visual components that can appear on the component palette and be used in the form designer. Properties and methods of TComponent provide basic behavior that descendant classes inherit as well as behavior that components can override to customize their behavior.

To create components which are visible to users at runtime, use TControl or its descendants as a base. To create controls based on Windows screen objects use TWinControl or its descendants as a base.

Класс TControl (Help)

TControl is the base class for all components that are visible at runtime.

Unit

Controls

Description

Controls are visual components, meaning the user can see them and possibly interact with them at runtime. All controls have properties, methods, and events that describe aspects of their appearance, such as the position of the control, the cursor or hint associated with the control, methods to paint or move the control, and events that respond to user actions.

TControl has many protected properties and methods that are used or published by its descendents.

Класс TStrings (Help)

TStrings is the base class for objects that represent a list of strings.

Unit

Classes

Description

Derive a class from TStrings to store and manipulate a list of strings. TStrings contains pure virtual methods and should not be directly instantiated.

Descendants of TStrings can represent several individual strings, such as the individual lines that appear in a list box. Some objects use descendants of TStrings to represent one long body of text so that it can be manipulated in smaller chunks.

TStrings introduces many properties and methods to

- Add or delete strings at specified positions in the list.
- Rearrange the strings in the list.
- Access the string at a particular location.
- Read the strings from or write the strings to a file or stream.
- Associate an object with each string in the list.

Свойства класса TStrings

Capacity - ограничение на кол-во строк
CommaText
Count - количество строк
DelimitedText
Delimiter
Names
Objects
QuoteChar
Strings - массив строк
StringsAdapter
Text - Строки как сплошной текст (строка)
UpdateCount
Values

Методы класса TStrings

~TStrings
Add
AddObject
AddStrings
Append
Assign
BeginUpdate
Clear
CompareStrings
DefineProperties
Delete
EndUpdate
Equals
Error
Exchange
ExtractName
Get
GetCapacity
GetCount
GetObject
GetText
GetTextStr
IndexOf
IndexOfName
IndexOfObject
Insert
InsertObject
LoadFromFile
LoadFromStream
Move
operator []
Put
PutObject

SaveToFile
SaveToStream
SetCapacity
SetText
SetTextStr
SetUpdateState
TStrings

Класс TGraphic (Help)

TGraphic is the abstract base class type for objects such as icons, bitmaps, and metafiles that can store and display visual images.

Unit

Graphics

Description

TGraphic is an abstract class that cannot be instantiated. Descendant graphics objects override many of the methods of TGraphic to address the needs of their particular file format and graphical characteristics. TGraphic also introduces methods that work with TPicture objects and the Clipboard. Properties of TGraphic provide information about the state and size of the graphic image.

When the type of graphic is known, store the graphic in its specific type object. Otherwise, use a TPicture object that can hold any type of TGraphic.

Свойства класса TGraphic

Empty

Height

Modified

Palette

PaletteModified

Transparent

Width

Методы класса TGraphic

~TGraphic
LoadFromClipboardFormat
LoadFromFile
LoadFromStream
SaveToClipboardFormat
SaveToFile
SaveToStream
TGraphic

Использование свойства Transparent

```
void __fastcall TForm1::Button1Click(TObject *Sender)

{
  Graphics::TBitmap *pBitmap = new Graphics::TBitmap();
  try
  {
    pBitmap->LoadFromFile("C:\\Program Files\\Common Files\\Borland
       Shared\\Images\\Splash\\256color\\factory.bmp");
    pBitmap->Transparent = true;
    pBitmap->TransparentColor = pBitmap->Canvas->Pixels[50,50];
    Form1->Canvas->Draw(0,0,pBitmap);
    pBitmap->TransparentMode = tmAuto;
    Form1->Canvas->Draw(50,50,pBitmap);
  }
  catch (...)

  {
    ShowMessage("Could not load or display bitmap");
  }
  delete pBitmap;
}
```

Пример с методом Assign

This example converts a bitmap image to a jpeg file by using the Assign method.

```
void __fastcall TForm1::Button1Click(TObject *Sender)

{
  //Requires "jpeg.hpp" to be included in the source file
  TJPEGImage *jp = new TJPEGImage();
  try
  {
    jp->Assign(Image1->Picture->Bitmap);
    jp->SaveToFile("c:\\oneeye.jpg");
  }
  __finally
  {
    delete jp;
  }
}
```

TCanvas

- Canvas provides properties, events and methods that assist in creating an image by
  - Specifying the type of brush, pen and font to use.
  - Drawing and filling a variety of shapes and lines.
  - Writing text.
  - Rendering graphic images.
  - Enabling a response to changes in the current image.
  - Своства
    - » Brush
    - » Pen
    - » Penpos
    - » Font
    - » Pixels

Методы TCanvas

- Arc
- BrushCopy
- Chord
- CopyRect
- Draw
- DrawFocusRect
- Ellipse
- FillRect
- FloodFill
- FrameRect
- HandleAllocated
- LineTo
- Lock
- MoveTo
- Pie
- PolyBezier
- PolyBezierTo

Polygon
Polyline
Rectangle
Refresh
RoundRect
StretchDraw
TextExtent
TextHeight
TextOut
TextRect
TextWidth
TryLock
Unlock

Пример 1 с Canvas

```
Graphics::TBitmap *BrushBmp = new Graphics::TBitmap;
try
{
  BrushBmp->LoadFromFile("emblem2.bmp");
  Form1->Canvas->Brush->Bitmap = BrushBmp;
  Form1->Canvas->FillRect(Rect(0,0,100,100));
}
__finally
{
  Form1->Canvas->Brush->Bitmap = NULL;
  delete BrushBmp;
}
```

Пример 2 с Canvas

```
TCanvas *pCanvas = Image1->Canvas;
  pCanvas->Brush->Color = clRed;
  pCanvas->Brush->Style = bsDiagCross;
  pCanvas->Ellipse(0, 0, Image1->Width,
   Image1->Height);
```

Основные компоненты

Listbox, Checklistbox,

свойства: Items, Selected, ItemIndex, MultiSelect

```
Edit1->Text=ListBox1->Items->Strings[ListBox1->ItemIndex];
```

Button

Speedbutton

свойство: Glyph - иконка

Radiobutton

Checkbox

Label

Scrollbar

Panel

Opendialog

Savedialog

FindDialog

RichEdit

Пример с RichEdit и ClipBoard

```
void __fastcall TForm1::Button1Click(TObject
   *Sender)
{
  RichEdit1->SelectAll();
  RichEdit1->CopyToClipboard();
  Edit1->Clear();
  Edit1->PasteFromClipboard();
  RichEdit1->SetFocus();
}
```

Основные компоненты (2)

Mainmenu,

Popupmenu

ProgressBar

Свойства: min, max, step, stepit

HeaderControl

StatusBar

Свойство simpletext

Timer

Animate

Пример с Animate

```
TForm *TempForm = new TForm(this);
  TAnimate *pAnimate = new TAnimate(TempForm);
  pAnimate->Parent = TempForm;
  pAnimate->CommonAVI = aviCopyFiles;
  pAnimate->Active = true;
  TempForm->Show();
  // Simulate a lengthy process.  Alter this value
  // to accommodate your machine speed.
  for (int i = 0; i < 90000000; i++)
    Application->ProcessMessages();
  delete TempForm;
```

Основные компоненты (3)

- Image
  - Center
  - IncrementalDisplay
  - Picture
  - Propotional
  - Stretch
  - Transparent
  - Style:

bsSolid	bsCross
bsClear	bsDiagCross
bsBDiagonal	bsHorizontal
bsFDiagonal	bsVertical
- ImageList
- PaintBox
- Paint