

G22.3033-002 Scripting Languages

5/29/2008
Objects in VBA
Properties, Call-backs

© Martin Hirzel G22.3033-002 NYU 5/29/2008 1

Outline

- Using objects in VBA
- Application-embedded scripting
- Classes and properties
- Visual programming and callbacks

© Martin Hirzel G22.3033-002 NYU 5/29/2008 2

VBA

Using Objects

```

Dim a1 As Apple      ' declare variables a1 and a2
Dim a2 As Apple      ' of class Apple
Set a1 = New Apple   ' allocate new objects, assign
Set a2 = New Apple   ' references (need "Set" keyword)
a1.color = "green"   ' set property, differently
a2.color = "red"     ' for a1 and a2
a1.prepare("slice") ' call method, passing
a2.prepare("squeeze") ' string parameter
    
```

© Martin Hirzel G22.3033-002 NYU 5/29/2008 3

Concepts

Classes and Objects

© Martin Hirzel G22.3033-002 NYU 5/29/2008 4

VBA

Abbreviated Member Access

```

With ActiveWindow.Selection.SlideRange.Shapes.Title
    .Flip(msoFlipVertical) ' call method
    .Rotation = 15         ' assign to property
End With
    
```

© Martin Hirzel G22.3033-002 NYU 5/29/2008 5

Concepts

Properties vs. Fields

- Both: dot notation look&feel
 - Writable: `a1.color = "red"`
 - Readable: `Debug.print a1.color`
- Properties only: associated behavior
 - E.g., update graphical representation
- Properties only: may be indexed, like arrays
 - `cake.ingredient("topping") = a1`
- Other languages with properties:
 - E.g., PHP, Delphi, C#

© Martin Hirzel G22.3033-002 NYU 5/29/2008 6

Concepts

Common Uses of Properties

<p><u>Simple (field-like)</u></p> <ul style="list-style-type: none"> • Visual update • Invariant checking <ul style="list-style-type: none"> – Filter illegal values – Read-only – Copy on write • Logging 	<p><u>Indexed (array-like)</u></p> <ul style="list-style-type: none"> • Collections <ul style="list-style-type: none"> – Resizable array – Hash map • Persistence <ul style="list-style-type: none"> – File – Database – Cookie
---	--

© Martin Hirzel G22.3033-002 NYU 5/29/2008 7

VBA

Collections

```

Dim col As Slides
Set col = ActivePresentation.Slides
Dim i As Integer
Debug.Print "for-loop, indexed property access"
For i = 1 To col.Count
    Debug.Print col.Item(i).Name
Next i
Debug.Print "for-loop, default property access"
For i = 1 To col.Count
    Debug.Print col(i).Name
Next i
Dim s As Slide
Debug.Print "for-each loop"
For Each s In col
    Debug.Print s.Name
Next s
    
```

© Martin Hirzel G22.3033-002 NYU 5/29/2008 8

Soap-box

Progressive Disclosure

- We only looked at how to use classes, but not how to define classes
 - That's sufficient for most VBA tasks!
- Language design encourages this:
 - Learn small subset of language to do most important tasks
 - Learn a little more to do a little more
- *Progressive disclosure* user experience

© Martin Hirzel G22.3033-002 NYU 5/29/2008 9

Outline

- Using objects in VBA
- Application-embedded scripting
- Classes and properties
- Visual programming and callbacks

© Martin Hirzel G22.3033-002 NYU 5/29/2008 10

VBA

Where Is My Code?

© Martin Hirzel G22.3033-002 NYU 5/29/2008 11

VBA

Structure of a VBA Application

- Projects
 - Files: .ppt, .xls, .doc
 - Normal.dot for Word
 - Personal Macro Workbook for Excel (Window→Unhide)
- Modules
 - Regular module
 - Class module
 - Form with code sheet
- Subroutines
 - Function or Sub (Macro)

© Martin Hirzel G22.3033-002 NYU 5/29/2008 12

VBA

Scopes and Visibility

- Project dependencies
 - Visual Basic Editor → Tools → References
 - Then, can call across projects:
`project.module.subroutine`
 - Cyclic references are not allowed
- Visibility
 - "Instancing" property of class module
 - Public/Private modifiers of declarations

© Martin Hirzel G22.3033-002 NYU 5/29/2008 13

Reference

Powerpoint Object Model

- The complete object model is much larger
- See Visual Basic help in editor
- Also in MSDN library:
 - Office development
 - Microsoft Office 2003
 - Office 2003
 - VBA reference
 - Powerpoint help
 - Object model

© Martin Hirzel G22.3033-002 NYU 5/29/2008 14

Concepts

Object Model

- Object-oriented API for embedded scripts
- Other examples:
 - Object models for other Microsoft apps
 - DOM = document object model for XML

© Martin Hirzel G22.3033-002 NYU 5/29/2008 15

Outline

- Using objects in VBA
- Application-embedded scripting
- Classes and properties
- Visual programming and callbacks

© Martin Hirzel G22.3033-002 NYU 5/29/2008 16

VBA

Defining Classes

```

Properties
Public color As String
Public weight As Double

Public Function pluck() As String
    pluck = Me.color & " apple"
End Function

Methods
Public Sub prepare (how As String)
    Dim dish As String
    Select Case how
        Case "slice"
            dish = "salad"
        Case "squeeze"
            dish = "juice"
        Case Else
            dish = "dessert"
    End Select
    Debug.Print "yum, " & pluck() & " " & dish
End Sub
    
```

© Martin Hirzel G22.3033-002 NYU 5/29/2008 17

Reference

Documentation of OO in VBA

- MSDN library
 - Development tools and languages
 - Visual Studio 6.0
 - Visual Basic 6.0
 - Product documentation
 - Using Visual Basic
 - Programmer's guide
 - Part 2: What can you do with Visual Basic
 - Programming with objects

© Martin Hirzel G22.3033-002 NYU 5/29/2008 18

VBA

Defining Properties

- Syntax
 - Property Get *id* ([*arg*"]) ... End Property
 - Property Let *id* ([*arg*'], *arg*) ... End Property
 - Property Set *id* ([*arg*'], *arg*) ... End Property
- Remarks
 - To return value from **Get**, assign to *id*
 - **Let** assigns regular value, **Set** assigns object
 - Extra arguments, if any, are for indexing
 - Each routine can be **Private** or **Public**
 - **Public** variable is just shortcut for routines

© Martin Hirzel G22.3033-002 NYU 5/29/2008 19

Concepts

Inheritance

© Martin Hirzel G22.3033-002 NYU 5/29/2008 20

VBA

Inheritance in VBA

```

Class module "Fruit" (Interface)
Public weight As Double
Public Function pluck() As String 'empty routine
End Function
Public Sub prepare(how As String) 'empty routine
End Sub
    
```

Would be abstract in other languages

Implements Fruit

```

Class module "Apple" (subclass)
Public color As String
Private wght As Double
Private Property Let Fruit_weight(ByVal RHS As Double)
wght = RHS
End Property
Private Property Get Fruit_weight() As Double
Fruit_weight = wght
End Property
Private Function Fruit_pluck() As String
Fruit_pluck = color & " apple"
End Function
Private Sub Fruit_prepare(how As String)
Debug.Print how & "d " & Fruit_pluck()
End Sub
    
```

"Implements" in subclass identifies superclass

Override private with mangled name

© Martin Hirzel G22.3033-002 NYU 5/29/2008 21

Concepts

Pure Interface Inheritance

- Interface inheritance (in VBA):
 - Subclass must provide own implementation for all methods in superclass
 - Can use object of subclass where object of superclass is expected
- Implementation inheritance (not in VBA):
 - Subclass can leave some methods from superclass unchanged
 - Those method implementations are reused
- To reuse code in VBA, must call it by hand

© Martin Hirzel G22.3033-002 NYU 5/29/2008 22

VBA

Subtyping Example

```

Dim someFruit As Fruit
If ... Then
Set someFruit = New Apple
Else
Set someFruit = New Banana
End If
' compiler knows that the method exists,
' even if it doesn't know which version
' will get called
someFruit.prepare("slice")
    
```

© Martin Hirzel G22.3033-002 NYU 5/29/2008 23

Soap-box

Missing OO Features in VBA

- Implementation inheritance
- Constructors
 - Can write `Class_Initialize()` method
 - Or write subroutine in separate module to allocate `new` instance and initialize it
- Static fields and static methods

⇒ OO features changed in VB.NET

© Martin Hirzel G22.3033-002 NYU 5/29/2008 24

Outline

- Using objects in VBA
- Application-embedded scripting
- Classes and properties
- Visual programming and callbacks

© Martin Hirzel G22.3033-002 NYU 5/29/2008 25

VBA Using Dialogs

- Dialog = window to request user input
- User form = module defining dialog

```

Loading is optional Name of module
Load frmLemonStar frmLemonStar.Show
MsgBox "done, number of lines was " & _
    frmLemonStar.txtNumberOfLines.Text
Unload frmLemonStar
Unloading is optional
    
```

© Martin Hirzel G22.3033-002 NYU 5/29/2008 26

Concepts Hungarian Notation

- Convention: start identifier with indication of type

chk	Check box
cmd	Command button
frm	User form
fra	Frame
lst	List box
cmb	Combo box
menu	Menu
opt	Option button
lbl	Label
txt	Text box

© Martin Hirzel G22.3033-002 NYU 5/29/2008 27

VBA Dialog Example

© Martin Hirzel G22.3033-002 NYU 5/29/2008 28

VBA Drag&Drop Dialog Design

- Right-click on project→Insert→User form
- View→Properties Window
 - Change name, caption, maybe width/height
- View→Toolbox
 - Drag and drop controls onto user form
 - Rename, resize, set initial value
 - Set "default" or "cancel" property for buttons
- Right-click on form→Tab order
 - Order when user "tabs through" controls

© Martin Hirzel G22.3033-002 NYU 5/29/2008 29

VBA Writing Event Handlers

```

Sub paintLemonStar (nLines As Integer)
    Dim n As Integer, i As Integer
    n = nLines - 1
    For i = 0 To n
        Dim l As Shape
        Set l = ActiveWindow.Selection.SlideRange.Shapes.AddLine( _
            BeginX:=300, BeginY:=200 + i * 100 / n, _
            EndX :=400, EndY :=300 - i * 100 / n)
        l.Line.ForeColor.RGB = RGB(i * 255 / n, i * 255 / n, 0)
    Next i
End Sub
    
```

Auxiliary subroutine

```

Private Sub cmdPaint_Click ()
    paintLemonStar CInt(txtNumberOfLines.Text)
End Sub
    
```

Callbacks with mangled names

```

Private Sub cmdCancel_Click ()
End Sub
    
```

Retrieve user input

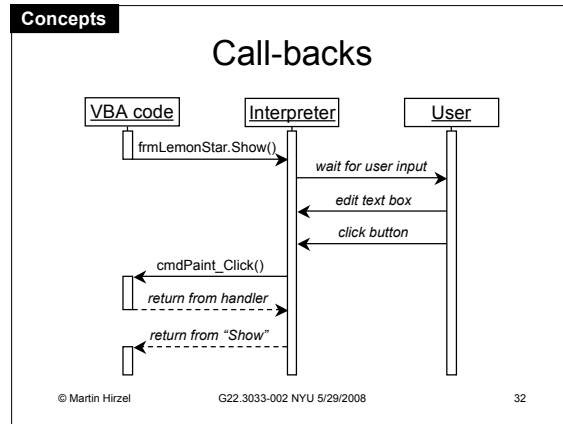
© Martin Hirzel G22.3033-002 NYU 5/29/2008 30

VBA

Code Sheets

- Right-click→View code
 - Just like other modules
 - Contains event handlers as well as regular subroutines
- Double-click on control
 - Goes to handler, creating it if necessary
 - Caution: handler name not changed automatically with control name!
- Dynamic dialog: assign properties at runtime
 - E.g., `Visible=True` reveals hidden controls
 - `UserForm_Initialize` sets default, e.g. for list box

© Martin Hirzel G22.3033-002 NYU 5/29/2008 31



Reference

Common Controls and Events

- Controls:
 - Label, TextBox, ComboBox, ListBox, CheckBox, OptionButton, ToggleButton, Frame, CommandButton, TabStrip, MultiPage, ScrollBar, SpinButton, Image
- Events:
 - Change, Click, DblClick, Enter, Exit, Initialize, KeyPress, SpinDown, and many more
- In editor for user form code sheet:
 - Left drop-down list: control objects on this form
 - Right drop-down list: events on that object

© Martin Hirzel G22.3033-002 NYU 5/29/2008 33

Concepts

Callback Mechanisms

VBA form	Subroutine in form with mangled name
VBA class	WithEvent / RaiseEvent statements
Java	Pass object on which to call method
Perl, Python, JavaScript	Pass anonymous function (lambda)
C, C++	Pass function pointer
C++	Pass object on which to call “()” operator
SmallTalk	Pass code block
PHP	Pass name of function as string

© Martin Hirzel G22.3033-002 NYU 5/29/2008 34

VBA

Events on Classes

- In event source class id_{src} :
 - `Public Event id_event (arg*)`
 - `RaiseEvent id_event (expr*)`
- In event sink module:
 - `WithEvents id_handler As id_src`
 - `Sub id_handler_id_event (arg*) ... End Sub`

© Martin Hirzel G22.3033-002 NYU 5/29/2008 35

VBA

Reusing Dialog Boxes

- Don't write “open file” dialog from scratch!
- `Application.Dialogs (id) .Show`
 - E.g., `id = xlDialogOptionsGeneral`
 - To find others: Help→Visual Basic→Search “build-in dialog argument”
- `Display` instead of `Show` prevents handlers
 - Return value: -2=Close, -1=OK, 0=Cancel, >0 other command buttons
 - Retrieve user input from controls with `.Value`

© Martin Hirzel G22.3033-002 NYU 5/29/2008 36

Soap-box

How to Learn a Language

- I. Use peers & gurus
- II. Install tools
- III. Read tutorial
- IV. Find language & library reference
- V. Read example programs
- VI. Write example programs:
I/O, types, control flow, libraries
- VII. Understand error messages
- VIII. Practice

© Martin Hirzel G22.3033-002 NYU 5/29/2008 37

VBA

Common VBA Mistakes

Error	Message	Common cause
Runtime error '91'	Object variable or With block variable not set	Missing "Set" before object assignment
Compile error	Invalid character	Missing space before "_" at end of line
Compile error	Expected: =	Parentheses around arguments, missing "Call"
	<i>And many more ...</i>	

© Martin Hirzel G22.3033-002 NYU 5/29/2008 38

Soap-box

Suggestions for VBA Practice

- hw02 gets you points, but you may want to do more at your own leisure
- Powerpoint
 - Center shape on slide
 - Draw object model
 - Plot a polynomial
- Excel
 - Create graph, set fonts and grid preferences
 - Generate email from name+title+address sheet

© Martin Hirzel G22.3033-002 NYU 5/29/2008 39

Administrative

Last Slide

- First homework due tomorrow at 4pm
- Quiz at **end** of next lecture at 7:30pm
- Please subscribe to class mailing list
- Today's lecture
 - Putting the V and the A in VBA
 - Embedded scripts
 - Properties
 - Call-backs
- Next lecture
 - Associative arrays
 - Regular expressions
 - Basics of Perl

© Martin Hirzel G22.3033-002 NYU 5/29/2008 40