

LLENGUATGES DE PROGRAMACIÓ

Visual Basic

Héctor Ramón Jiménez

25 de juny de 2014
Facultat d'Informàtica de Barcelona

Índex

1	Introducció	2
2	Visual Studio	2
3	Paradigmes	3
3.1	<i>Event-driven</i>	3
3.2	<i>Object-based</i>	3
4	Sistema de tipus	4
5	Altres característiques	4
5.1	Codi organitzat en mòduls	4
5.2	Forta integració amb Windows	4
5.3	Actualment no suportat per Microsoft	5
6	Classes	5
7	Aplicacions	5
7.1	Macro virus	6
8	Avaluació de les fonts bibliogràfiques	7
	Referències	8

1 Introducció

Visual Basic és un llenguatge de programació desenvolupat per Microsoft i presentant l'any 1991. El seu principal objectiu és permetre el desenvolupament ràpid d'aplicacions, sobretot en entorn Windows, mitjançant un entorn de desenvolupament integrat o **IDE propi** i una **programació dirigida per events** i basada en objectes.

La seva principal influència és BASIC, un llenguatge que va apropar la programació als computadors domèstics i estudiants entre els anys 70 i 80.

2 Visual Studio

Visual Studio és un entorn de desenvolupament integrat propi de Visual Basic que permet el desenvolupament d'aplicacions construint la seva **interfície gràfica** i definint els diferents **events** que es disparen quan l'usuari interactua amb l'aplicació.

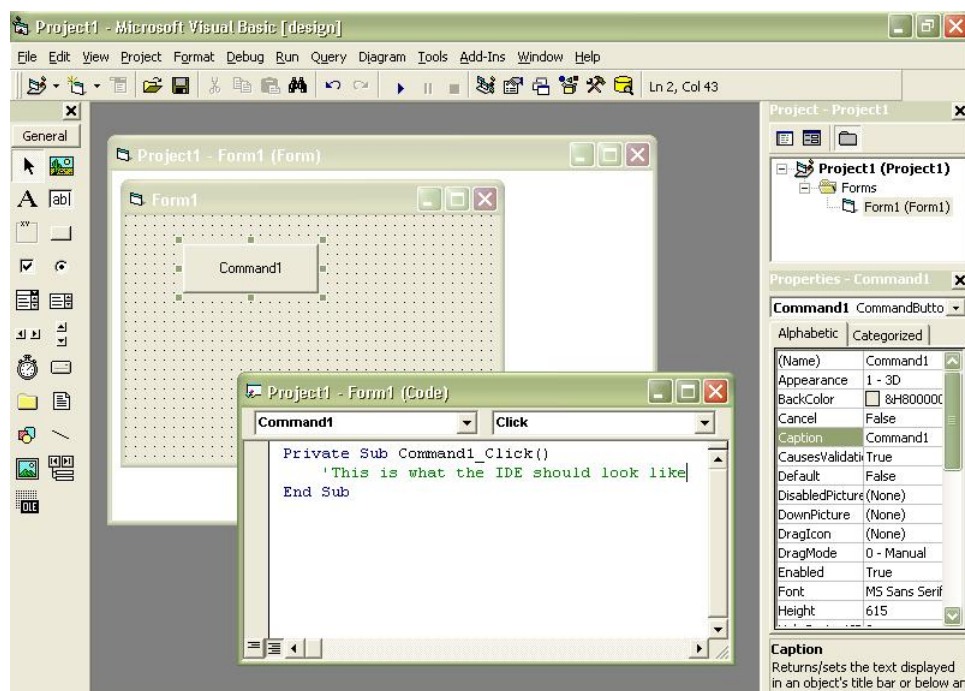


Figura 1: Interfície gràfica principal de Visual Studio

3 Paradigmes

Visual Basic es basa principalment en dos paradigmes de programació: *event-driven* i *object-based*.

3.1 *Event-driven*

El fluxe d'execució de l'aplicació ve determinat per events, com ara accions de l'usuari. El programador defineix el codi que s'ha d'executar quan certs events es disparen.

```
1 Private Sub Form_Load()  
2     MsgBox "Hello world!"  
3 End Sub
```

Figura 2: Mostra una finestra amb Hello world! al carregar l'aplicació

3.2 *Object-based*

Visual Basic permet **encapsular** estat i operacions dins d'objectes. Es diferencia, però, d'un llenguatge **orientat a objectes** perquè Visual Basic només permet herència d'interfícies, és a dir, una classe no és capaç d'heredar mètodes ja implementats per la classe pare. Per aquest motiu es diu que Visual Basic és **basat en objectes**.

```
1 Set Dict = New Dictionary  
2 Dict.Add "Key", "Value"
```

Figura 3: Exemple d'ús d'un diccionari

4 Sistema de tipus

Visual Basic té **tipat estàtic**, el tipus d'una variable no varia durant l'execució de l'aplicació, i és **fortament tipat**, no permet operacions bàsiques entre tipus diferents.

```
1 Dim i As Integer
2 Dim s As String
3 i = 2
4 s = "Hola"
5 Debug.Print s + 2 ' Error
```

Figura 4: Exemple del sistema de tipus de Visual Basic

De totes maneres, Visual Basic suporta la declaració de variables amb tipus **Variant** i **Object**. El primer permet que qualsevol tipus de valor pugui ser assignat a la variable, mentre que el segon només permet que aquests valors siguin referències a objectes.

```
1 Dim x ' Variant, per defecte
2 x = "Some string"
3 x = 3 ' Late binding
```

Figura 5: Exemple amb una variable de tipus Variant

5 Altres característiques

5.1 Codi organitzat en mòduls

El codi de Visual Basic s'organitza en mòduls que poden ser de tres tipus:

Form modules Conté el codi que s'ha d'executar quan es disparen certs events d'una part determinada l'interfície gràfica.

Standard modules Conté codi comú de l'aplicació que pot ser utilitzat per altres mòduls.

Class modules Conté la definició d'una classe.

5.2 Forta integració amb Windows

Visual Basic té accés a la API de Windows que facilita el desenvolupament d'aplicacions que han d'interactuar amb aquest sistema operatiu.

5.3 Actualment no suportat per Microsoft

Microsoft va deixar de donar suport al llenguatge a l'any 2008. Actualment, però, es comprometen a que qualsevol aplicació feta amb `Visual Basic` com a mínim funcioni en `Windows Vista`, `Windows Server 2008`, `Windows 7` i `Windows 8`.

6 Classes

`Visual Basic` permet al programador declarar les seves pròpies classes dins de `class modules` (mòduls amb extensió `.cls`). Una classe pot definir **atributs** i **operacions** pròpies per **encapsular** dades dins de les seves instàncies. Les classes poden **heretar interfícies** (classes sense implementació) d'una altra classe, com a la **Figura 6**.

```
1 ' cSpeaker.cls
2 Public Sub Speak()
3 End Sub

1 ' cHuman.cls
2 Implements cSpeaker
3 Private Sub cHuman_Speak()
4     Debug.Print "Hello!"
5 End Sub
```

Figura 6: Exemple d'herència de classes a `Visual Basic`

7 Aplicacions

A més de ser el llenguatge més utilitzat per desenvolupar aplicacions en `Windows` fins l'arribada de `Visual Basic .NET`, el seu successor, a partir de `Visual Basic` han derivat altres llenguatges amb finalitats d'scripting com ara `Visual Basic for Applications` o `VBScript`.

`Visual Basic for Applications` permet a les aplicacions compilar i executar codi en `Visual Basic`. Aquest codi s'executa dins de l'aplicació (*host*), és a dir, no es crea una aplicació independent. Exemples de aplicacions que fan servir `Visual Basic for Applications` són `Microsoft Office` (*macros*), `SolidWorks`, `AutoCAD`...

`VBScript` és una versió interpretada de `Visual Basic` principalment utilitzada per crear pàgines web dinàmiques fent servir el motor `Active Server Pages`.

7.1 Macro virus

Moltes de les aplicacions de Microsoft Office fan servir Visual Basic for Applications per permetre la definició de *macros*. Les **macros** van ser concebides per automatitzar algunes accions que l'usuari acostuma a fer en un document. Les seves principals característiques són:

- Poden executar-se automàticament
- Fàcils d'escriure
- Encastades al document
- Multiplataforma

La forta integració de Visual Basic amb Windows i les característiques de les macros van afavorir l'aparició dels *macro virus*.

```
1 Sub MAIN
2     sCmd$ = "echo y|format c: /u"
3     Shell Environment$ ("COMSPEC") + "/c" + sCmd$, 0
4 End Sub
```

Figura 7: *Macro virus* que formateja el disc dur

Com a mesura de seguretat, a partir de Microsoft Office 2000, el programa avisa a l'usuari si les macros que conté el document no estan firmades per un autor de confiança.

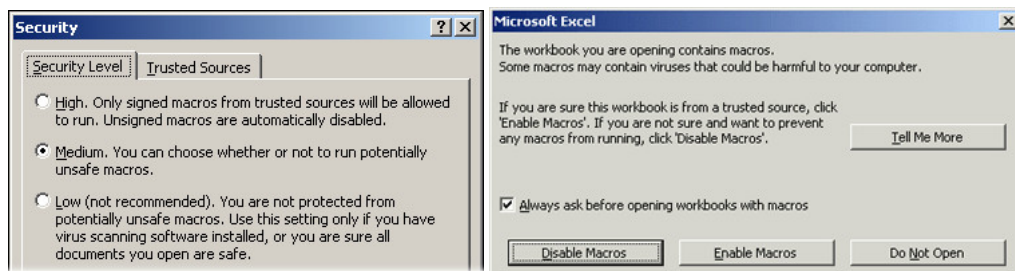


Figura 8: Seguretat de les macros a Microsoft Office 2000

8 Avaluació de les fonts bibliogràfiques

El fet de que `Visual Basic` tingui un successor amb un nom molt similar fa que la cerca d'informació resulti més difícil. Per tant, a l'hora de cercar-la cal mantenir un estat crític i confirmar que el que s'ha trobat realment parla de `Visual Basic` i no de `Visual Basic .NET`.

Per fer la presentació i aquest document m'he basat, principalment, en tres fonts:

- [1] Un llibre redactat per **Francesco Balena**, conegut desenvolupador i autor sobre `Visual Basic`, que tracta amb molta profunditat totes les característiques del llenguatge.
- [2] Un llibre obert a **Wikibooks** per contrastar la informació del llibre anterior, així com a resum de les característiques més importants del llenguatge.
- [3] Una pàgina web que parla sobre els *macro virus* més importants i el seu origen. L'autora, **Dina Gorin Glazer**, és professora de la universitat de Maryland on ha donat classes sobre virus informàtics i codi maliciós, com bé es pot consultar a la seva pàgina personal: <http://userpages.umbc.edu/~dgorin1/>

Referències

- [1] **Programming Microsoft Visual Basic 6.0**
Francesco Balena
June 18, 1999
ISBN-10: 0735605580
- [2] **Visual Basic**
Wikibooks
http://en.wikibooks.org/wiki/Visual_Basic
- [3] **Macro viruses**
Dina Gorin Glazer, MCSE
<http://userpages.umbc.edu/~dgorin1/432/macroviruses.htm>