

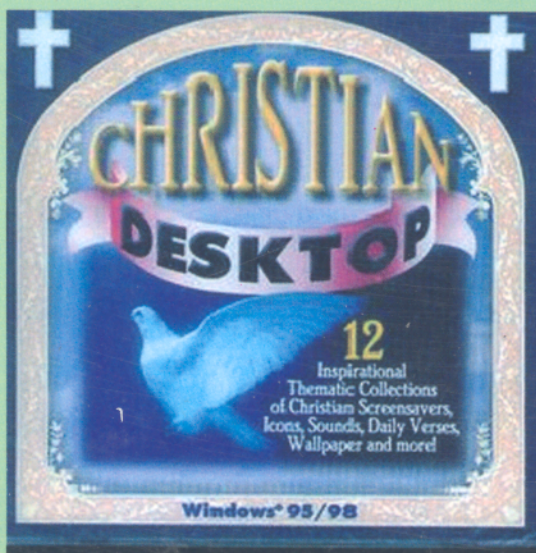


SỞ GIÁO DỤC VÀ ĐÀO TẠO HÀ NỘI
TRUNG TÂM TIN HỌC - NGOẠI NGỮ TRÍ ĐỨC

GIÁO TRÌNH

CƠ SỞ LẬP TRÌNH VISUAL BASIC

Ban biên tập: Trung tâm Tin học -
Ngoại ngữ Trí Đức



NHÀ XUẤT BẢN THỐNG KÊ - 2003

TRUNG TÂM NGOẠI NGỮ - TIN HỌC TRÍ ĐỨC

LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI VISUAL BASIC

**NHÀ XUẤT BẢN THỐNG KÊ
HÀ NỘI - 2003**

LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI VISUAL BASIC

Chịu trách nhiệm xuất bản:

CÁT VĂN THÀNH

In 1000 cuốn khổ 14,5 x 20,5 cm tại Xưởng in Nhà xuất bản Thống kê. Giấy phép xuất bản số 121-205/XB-QLXB do cục xuất bản cấp ngày 03 tháng 03 năm 2003. In xong và nộp lưu chiểu quý IV năm 2003.

Lời nói đầu

Cấu trúc dữ liệu và giải thuật là một trong những cuốn sách cơ sở trong chương trình tạo chuyên viên tin học và cử nhân tin học. Nội dung cuốn sách trình bày một số kiến thức cơ bản về cấu trúc dữ liệu và các giải thuật có liên quan, giúp cho sinh viên nâng cao kỹ năng lập trình, hỗ trợ khả năng đi sâu vào các môn học chuyên ngành như cơ sở dữ liệu, trí tuệ nhân tạo, hệ chuyên gia, chương trình dịch ... Nhằm xây dựng một cuốn sách vừa đảm bảo tính chuẩn mực, vừa đáp ứng nhu cầu thực hành của sinh viên. Chúng tôi đã tham khảo nhiều tài liệu giá trị của các tác giả trong và ngoài nước nhằm cung cấp kiến thức về môn một cách có hệ thống, nhiều vấn đề được minh họa trực quan và hướng dẫn theo từng bước lập trình cụ thể cho học viên.

Mặc dù rất nhiều cố gắng nhưng chắc chắn không thể tránh được thiếu sót. Nhóm tác giả rất mong nhận được các ý kiến đóng góp để nội dung cuốn sách ngày càng được hoàn thiện hơn nữa, trong lần tái bản sau.

Nhóm tác giả

MỤC LỤC

CHƯƠNG I: TỔNG QUAN VỀ VISUAL BASIC 3

I. KHỞI ĐỘNG CHƯƠNG TRÌNH VISUAL BASIC.....7

II. TẠO THƯ MỤC LÀM VIỆC.....8

III. GIỚI THIỆU GIAO DIỆN CỬA SỔ CỦA VISUAL BASIC 8

IV. TẠO/LUU PROJECT LÀM VIỆC..... 12

V. LÀM VIỆC VỚI CỬA SỔ PROJECT 16

VI. GIỚI THIỆU TỔNG QUAN VỀ FORM..... 17

CHƯƠNG II: FORM VÀ CÁC CONTROL TRONG VISUAL BASIC..... 26

I. GIỚI THIỆU26

II. CÁC THAO TÁC CƠ BẢN TRÊN FORM.....28

III. CÁC THUỘC TÍNH CHUNG30

IV. GIỚI THIỆU CÁC CONTROL CƠ BẢN TRONG TOOLBOX.....32

CHƯƠNG III: THIẾT KẾ MENU 48

I. THIẾT KẾ GIAO DIỆN.....52

II. VIẾT LỆNH CHO CÁC THỦ TỤC ĐÁP ỨNG BIẾN CỎ CHỌN MỤC MENU.....65

CHƯƠNG IV: SỰ KIỆN TRÊN FORM	69
I. SỰ KIỆN TRÊN FORM.....	69
II. SỰ KIỆN TRÊN NHÃN (LABEL).....	71
III. SỰ KIỆN TRÊN HỘP VĂN BẢN (TEXTBOX).....	72
IV. SỰ KIỆN TRÊN NÚT (BUTTON).....	73
V. SỰ KIỆN TRÊN HỘP KIỂM TRA (CHECK BOX).....	74
VI. SỰ KIỆN TRÊN NÚT TÙY CHỌN (OPTION BUTTON).....	77
VII. SỰ KIỆN TRÊN HỘP LIỆT KÊ (LIST BOX).....	79
VIII. SỰ KIỆN TRÊN HỘP LIỆT KÊ THẢ (COMBOBOX).....	80
 CHƯƠNG V: VIẾT LỆNH.....	 82
I. GIỚI THIỆU	82
II. CÁC KIỂU DỮ LIỆU TRONG VISUAL BASIC	83
III. KHAI BÁO BIẾN	90
IV. CÁC CẤU TRÚC ĐIỀU KHIỂN TRONG VISUAL BASIC.....	92
V. HÀM VÀ THỦ TỤC.....	98
 CHƯƠNG VI: DEBUG	 104
I. GIỚI THIỆU	104
II. DESIGN TIME, RUN TIME, VÀ BREAK MODE.....	106
III. CÁCH SỬ DỤNG CÁC CÔNG CỤ DEBUG	109

CHƯƠNG VII: KẾT NỐI CƠ SỞ DỮ LIỆU VỚI ADO..... 118

I. GIỚI THIỆU KỸ THUẬT LẬP TRÌNH ADO.....	119
II. SỬ DỤNG ADO DATA CONTROL.....	121
III. SỬ DỤNG OBJECT INTERFACE.....	139

CHƯƠNG VIII: MỘT SỐ ĐIỀU KHIỂN KẾT BUỘC DỮ LIỆU..... 161

I. DATACOMBO VÀ DATALIST.....	161
II. DATAGRID.....	176

CHƯƠNG IX: REPORT..... 195

I. MỞ ĐẦU.....	195
II. VÍ DỤ 1.....	198
III. VÍ DỤ 2.....	207
IV. THÊM MỘT FIELD TÍNH TOÁN (CALCULATED FIELD) VÀO DATA REPORT.....	212
V. THÊM MỘT FUNCTION CONTROL VÀO DATA REPORT.....	214
VI. NHÓM (GROUP) CÁC THÔNG TIN TRONG DATA REPORT.....	219

Chương I:

TỔNG QUAN VỀ VISUAL BASIC

Visual Basic là một công cụ lập trình trực quan của Microsoft, giúp ta có thể xây dựng nhanh các ứng dụng trên Windows. Khác với các môi trường lập trình hướng thủ tục trước đây, Visual Basic là môi trường lập trình hướng sự kiện trên Windows.

Visual Basic cung cấp một bộ công cụ hoàn chỉnh để đơn giản hóa việc triển khai lập trình ứng dụng.

Có 3 ấn bản VB6: Learning, Professional và Enterprise. Trong đó ấn bản Learning không thông dụng lắm. Ấn bản Professional cung cấp đầy đủ những gì ta cần để triển khai một ứng dụng, nhất là các Control ActiveX. Ấn bản Enterprise là ấn bản Professional cộng thêm các công cụ Back Office (SQL Server, Microsoft Transaction Server, Internet Information Server).

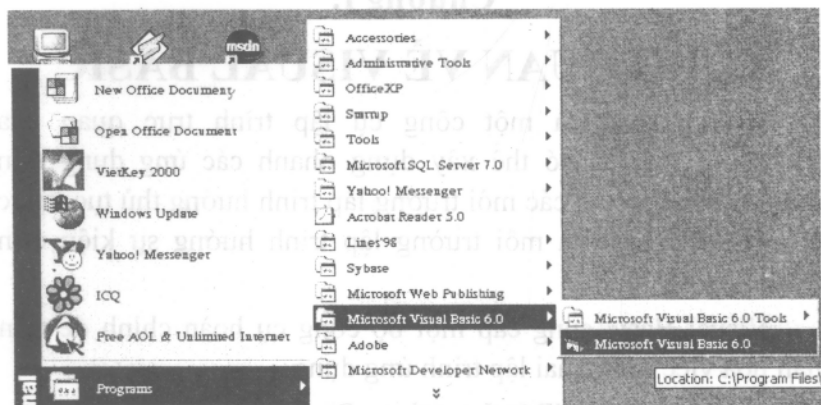
Khi viết một chương trình trong Visual Basic, ta phải qua hai bước: thiết kế giao diện và viết lệnh. Khi thiết kế giao diện, ta sử dụng các công cụ do Visual Basic cung cấp sẵn. Các công cụ này cho phép thiết kế giao diện bằng chuột lẫn bàn phím.

Trong chương này, ta chỉ cần nắm rõ cách sử dụng các công cụ sao cho thành thạo.

I. KHỞI ĐỘNG CHƯƠNG TRÌNH VISUAL BASIC

Ta có thể khởi động chương trình Visual Basic theo cách sau:

Nhấp nút Start → Programs → Microsoft Visual Studio 6.0
→ Microsoft Visual Basic 6.0



- Hình 1.1 -

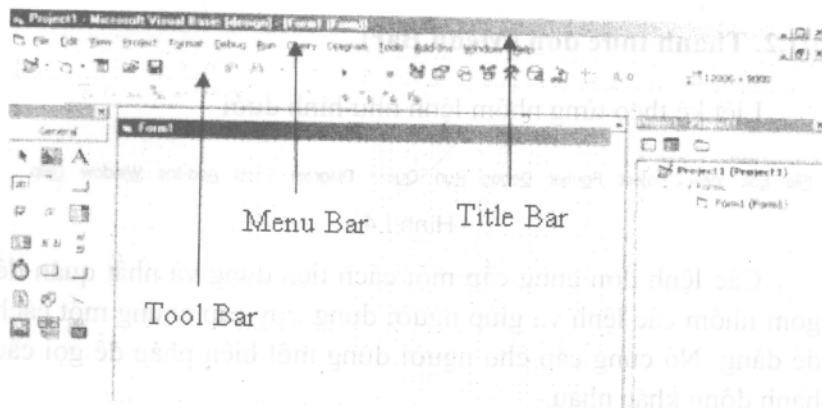
II. TẠO THƯ MỤC LÀM VIỆC

Trước khi lập trình, ta cần tạo một thư mục chứa các chương trình làm việc.

Ví dụ: lưu các chương trình trong thư mục: **D:\VB**.

III. GIỚI THIỆU GIAO DIỆN CỬA SỔ CỦA VISUAL BASIC

Sau khi khởi động, cửa sổ Visual Basic có dạng như hình bên dưới :






- Hình 1.2 -







III.1. Thanh tiêu đề (Title Bar)

Thẻ hiện tên của chương trình là Microsoft Visual Basic

Project1 - Microsoft Visual Basic [design]

- Hình 1.3 -

Góc phải thanh tiêu đề có 3 nút điều khiển   

- Nút thu nhỏ (minimize window)  : thu nhỏ cửa sổ làm việc
- Nút phóng to và phục hồi (maximize/restore window)  /  :
 - + Nếu cửa sổ đang ở trạng thái bị thu hẹp kích thước, nút maximize () giúp phóng to cửa sổ (cửa sổ ở trạng thái lớn nhất)
 - + Nếu cửa sổ đang ở trạng thái lớn nhất về kích thước, nút Restore window () giúp ta có thể thu hẹp bớt kích thước cửa sổ.
- Nút đóng cửa sổ (Close)  : khi nhấn vào nút này, cửa sổ làm việc của Visual Basic sẽ bị đóng lại.

III.2. Thanh thực đơn (Menu Bar)

Liệt kê theo từng nhóm lệnh như hình dưới

File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help

- Hình 1.4 -

Các lệnh đơn cung cấp một cách tiện dụng và nhất quán để gom nhóm các lệnh và giúp người dùng truy cập chúng một cách dễ dàng. Nó cung cấp cho người dùng một biện pháp để gọi các hành động khác nhau.

Thanh lệnh đơn (menu bar) xuất hiện ngay dưới thanh tiêu đề (title bar) và chứa một hoặc nhiều tùy chọn lệnh đơn, mỗi tùy chọn trên thanh lệnh đơn định danh một phạm trù các tác vụ.

Xuất hiện ngay dưới mỗi tùy chọn lệnh đơn có thể là một hoặc nhiều tùy chọn. Các tùy chọn xuất hiện trên thanh lệnh đơn được gọi là các tùy chọn lệnh đơn chính và những tùy chọn dưới chúng được gọi là các tùy chọn con.

Ví dụ vài lệnh đơn của Visual Basic:

- + File: có các tùy chọn con tạo thuận lợi cho các tác vụ điều quản tập tin như mở, đóng, lưu các tập tin
- + Edit: có các tùy chọn con tạo thuận lợi cho các tác vụ hiệu chỉnh như cắt, chép, dán văn bản.
- + Help: có các tùy chọn con hỗ trợ trợ giúp trực tuyến cho nhiều công việc khác nhau trong ứng dụng.

.....

III.3. Thanh công cụ (Tool Bar)

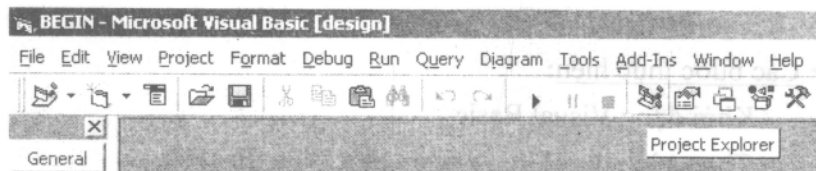
Là các lệnh được thể hiện dưới dạng biểu tượng giúp người sử dụng có thể thi hành một lệnh nhanh hơn.

- Thanh công cụ chứa các nút thanh công cụ, cho phép nhanh chóng truy cập đến các lệnh thường dùng nhất.
- Các nút trong thanh công cụ cũng tương tự như các mục trong thanh thực đơn. Các nút trong thanh công cụ cung cấp cho người dùng một lối tắt đối với các hành động mà ta phải chọn qua nhiều bước bằng lệnh đơn.



- Hình 1.5 -

- Các nút trong thanh công cụ có một ô gợi nhớ (gọi là Tool Tip) – là những cửa sổ nhỏ bật ra chứa một mô tả vắn bản ngắn gọn về công dụng và tên gọi của nút trong thanh công cụ. Khi người dùng đưa chuột đến một nút, cửa sổ gợi nhớ sẽ bật ra vắn bản nhắc nhở.



- Hình 1.6 -

➤ Các tính chất của thanh công cụ:

Các tính chất duy nhất đối với một điều khiển thanh công cụ gồm:

- + Style: xác định hình dáng của điều khiển
- + TextAlignment: xác định vị trí của văn bản tương đối với nút
- + ToolTipText: ô gợi nhớ tên công cụ

➤ Các phương pháp của thanh công cụ:

Dưới đây là vài phương pháp thường dùng:

- + Move

+ Drag

➤ Các sự kiện của thanh công cụ:

Dưới đây là sự kiện thường dùng:

+ ButtonClick: xảy ra khi người dùng nhấp lên một đối tượng Object trong một điều khiển ToolBar

➤ Vài điểm cần lưu ý về thanh công cụ:

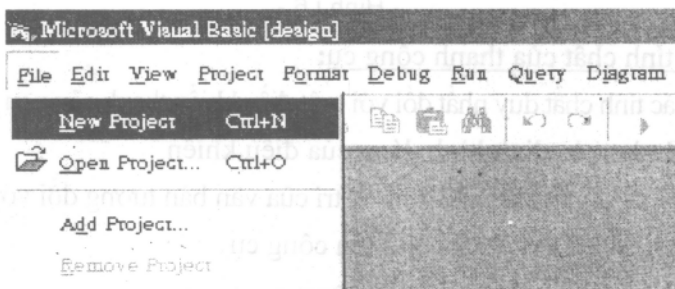
- Một thanh công cụ cung cấp các lối tắt đối với các hành động (hành động đó có thể được gọi bằng cách dùng thanh lệnh đơn). Tuy nhiên, thanh công cụ có thể cung cấp các tùy chọn không có trong các lệnh đơn
- Hầu hết các tùy chọn trên một thanh công cụ đều xuất hiện dưới dạng các nút, ngoài ra một tùy chọn thanh công cụ cũng có thể hiển thị dưới dạng một hộp liệt kê

IV. TẠO / LƯU PROJECT LÀM VIỆC

IV.1. Tạo Project

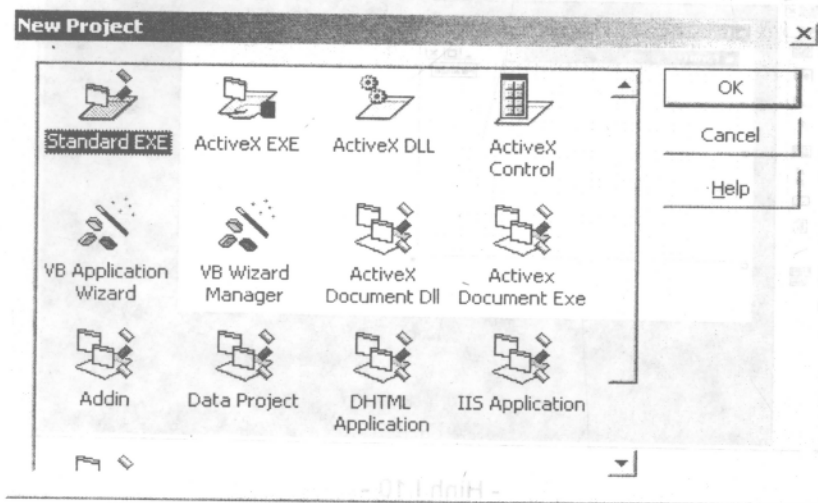
➤ Các bước thực hiện:

- Khởi động Visual Basic
- Chọn trình đơn **File** → chọn mục **New Project**



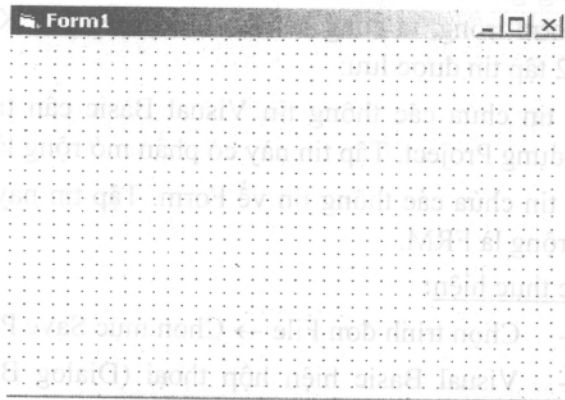
- Hình 1.7-

- Trong hộp thoại New Project, chọn loại Project muốn tạo, ở đây ta sẽ chọn Standard EXE như hình sau



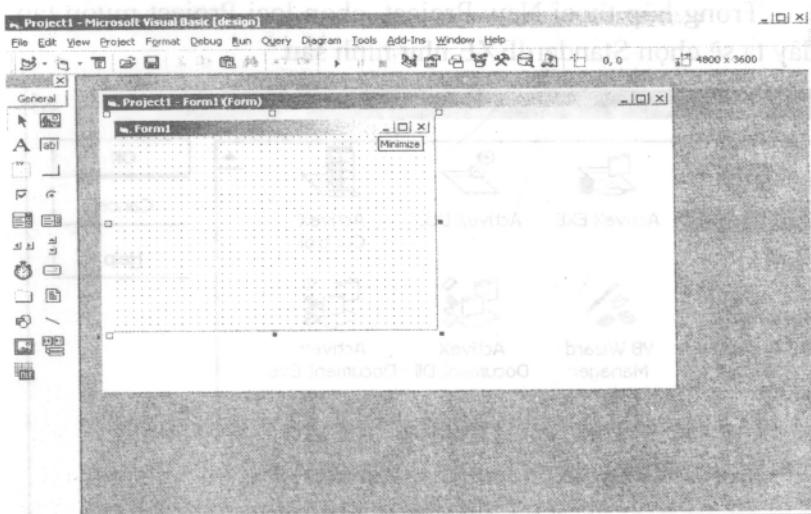
- Hình 1.8 -

- Visual Basic sẽ hiện một cửa sổ trên màn hình. Đây là **Form** (cửa sổ) trống với tiêu đề (caption) là Form1.



- Hình 1.9 -

Cửa sổ chương trình lúc này có dạng:



- Hình 1.10 -

IV.2. Lưu Project

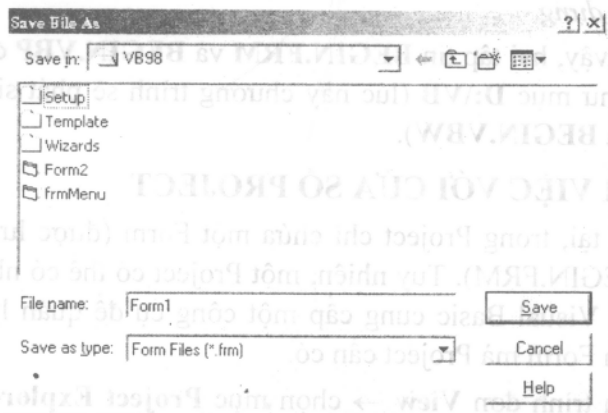
Một trong những vấn đề quan trọng khi viết chương trình là lưu lại những gì mình đã làm. Mặc dù chưa có thao tác gì thay đổi trong Form trống, ta cũng nên lưu Form lên đĩa. Khi ta lưu Project, có 2 tập tin được lưu:

1. Tập tin chứa các thông tin Visual Basic cần trong việc xây dựng Project. Tập tin này có phần mở rộng là VBP.
2. Tập tin chứa các thông tin về Form. Tập tin này có phần mở rộng là FRM.

➤ Các bước thực hiện:

- Chọn trình đơn File → Chọn mục Save Project As
- Visual Basic hiện hộp thoại (Dialog Box) Save File As

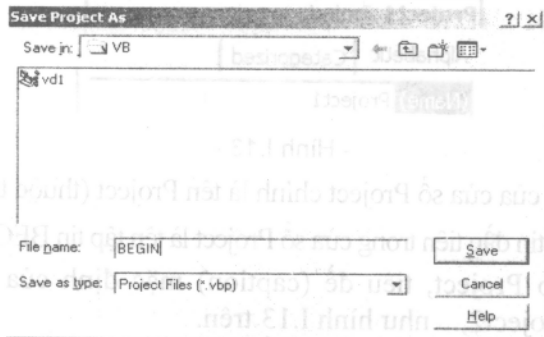
Chọn thư mục **D:\VB** làm thư mục lưu và thay đổi tên mặc định của Form trống từ **Form1.FRM** thành **BEGIN.FRM**



- Hình I.11 -

Lưu ý: Không nên dùng tên mặc định Visual Basic đặt cho Form, ta nên đặt tên Form gần gũi với các chức năng của Form bạn đang muốn xây dựng.

- Lúc này, Visual Basic hiện tiếp hộp thoại **Save File As**, tập tin Project có tên mặc định là **Project1.VBP**. Ta đổi lại tên cho tập tin Project là **BEGIN.VBP** và lưu vào thư mục **D:\VB**.



- Hình I.12 -

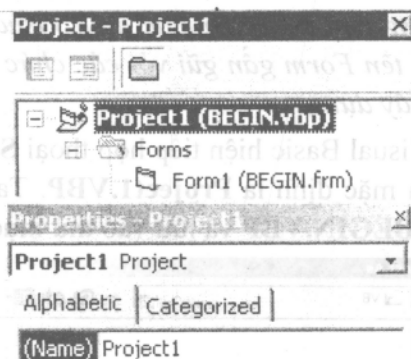
Lưu ý: Không nên dùng tên mặc định Visual Basic đặt cho Project, ta nên đặt tên Project gần gũi với chương trình ta đang muốn xây dựng.

Như vậy, hai tập tin **BEGIN.FRM** và **BEGIN.VBP** đã được lưu vào thư mục **D:\VB** (lúc này chương trình sẽ phát sinh một file phụ là **BEGIN.VBW**).

V. LÀM VIỆC VỚI CỬA SỔ PROJECT

Hiện tại, trong Project chỉ chứa một Form (được lưu thành tập tin **BEGIN.FRM**). Tuy nhiên, một Project có thể có nhiều tập tin Form. Visual Basic cung cấp một công cụ để quản lý tất cả các tập tin Form mà Project cần có.

Chọn trình đơn **View** → chọn mục **Project Explorer**. Cửa sổ Project xuất hiện như Hình bên dưới.



- Hình I.13 -

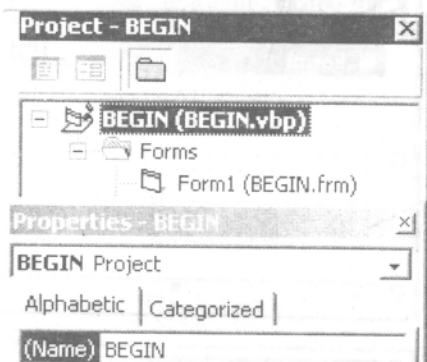
Tiêu đề của cửa sổ Project chính là tên Project (thuộc tính Name).

Tên tập tin đầu tiên trong cửa sổ Project là tên tập tin **BEGIN.FRM**.

Khi tạo Project, tiêu đề (caption) mặc định của Project là **Project1**, **Project2**,... như hình I.13 trên.

Do đó, lúc này, tên cửa sổ Project cũng là **Project 1** như ta đã thấy trong hình.

Muốn đổi tên hiển thị này của Project, ta có thể chọn vào Project trong cửa sổ Project, lúc này ta thấy bên dưới sẽ xuất hiện một cửa sổ phụ, đó là cửa sổ **Properties** của Project, tại đây, ta có thể thay đổi thuộc tính **Name** của Project



- Hình 1.14 -

Bây giờ, tên hiển thị của Project đã được đổi lại thành BEGIN, nên tên của cửa sổ Project lúc này cũng là BEGIN

VI. GIỚI THIỆU TỔNG QUAN VỀ FORM

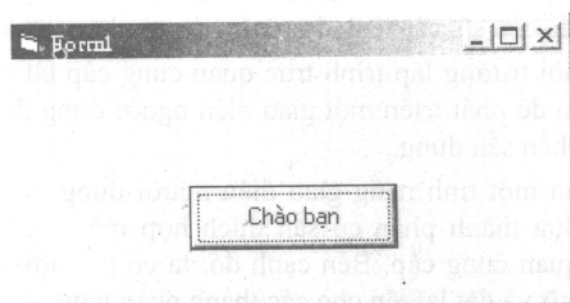
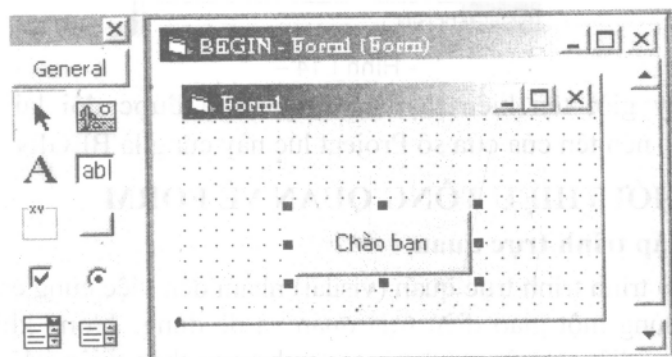
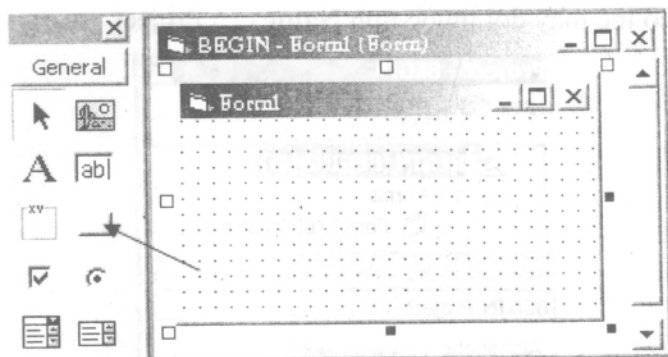
VI.1. Lập trình trực quan

Lập trình trình trực quan (visual) nhắm đến việc cung cấp cho người dùng một giao diện trực quan và dễ dùng. Trong khi phát triển giao diện, ta cần sử dụng các tính năng thân thiện với người dùng như các cửa sổ, các lệnh đơn, các nút, các hộp liệt kê,...

Một môi trường lập trình trực quan cung cấp tất cả các tính năng cần có để phát triển một giao diện người dùng đồ họa dưới các thành phần sẵn dùng.

Khi cần một tính năng giao diện người dùng cụ thể, ta chỉ việc chọn lựa thành phần có sẵn thích hợp mà môi trường lập trình trực quan cung cấp. Bên cạnh đó, ta có thể dời vị trí, điều chỉnh kích cỡ và đặt lại tên cho các thành phần này.

Ví dụ: nếu ta cần dùng một nút, môi trường lập trình trực quan sẽ cung cấp cho ta nút đó. Tất cả những gì ta phải làm là lựa một nút rồi đặt nó lên màn hình tại vị trí thích hợp.



- Hình I.15 -

Thông thường, ta dùng chuột để lựa chọn và sắp đặt các thành phần cần thiết. Như vậy, môi trường lập trình trực quan cũng được gọi là môi trường nhấp thả.

Một môi trường lập trình trực quan tự động hóa tiến trình tạo một giao diện người dùng. Giao diện mà một môi trường lập trình trực quan cung cấp cho ta được gọi là giao diện trực quan. Dùng giao diện trực quan, ta sẽ thiết kế giao diện dễ dàng hơn mà không phải viết mã.

Ngoài ra, môi trường lập trình trực quan cũng cung cấp một biện pháp để kết hợp mã với mỗi thành phần.

VI.2. Giới thiệu FORM

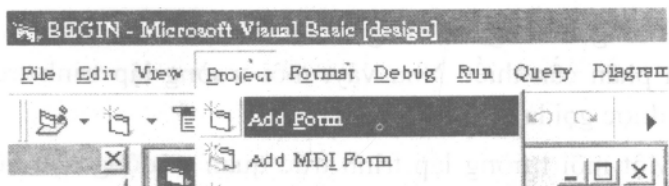
Hầu hết các chương trình trong Visual Basic đều có ít nhất một Form. Form còn gọi là biểu mẫu, là thành phần quan trọng nhất trong số tất cả các thành phần giao diện. Form là cơ sở cho việc thiết kế giao diện người dùng của ứng dụng. Tất cả các thành phần tạo thành giao diện người dùng đều được đặt trong Form.

Khi ứng dụng chạy, nó mở ra một Form. Sau khi Form được nạp vào bộ nhớ, tất cả các thành phần hàm chứa trong nó sẽ hiện diện và có thể truy cập.

VI.3. Cách tạo Form

- Sau khi tạo Project mới, môi trường triển khai lập trình cho sẵn ta một Form.
- Trong trường hợp ta muốn tạo một Form mới trên một Project đã mở sẵn, ta có 2 cách:

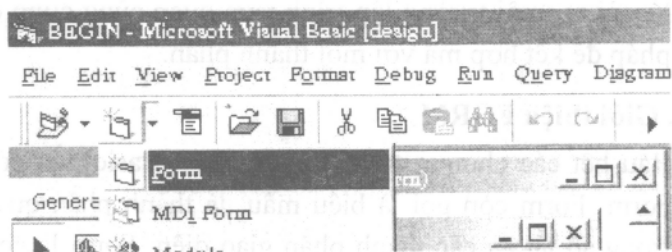
Cách 1: Dùng Menu: Vào trình đơn **Project** → chọn mục **Add Form** như hình sau:



- Hình I.16 -

Một Form mới sẽ được tạo ra và thêm vào trong Project.

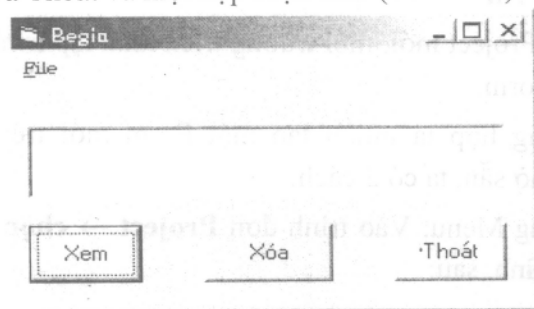
Cách 2: Dùng công cụ: Trên thanh công cụ, ta chọn vào công cụ **Add Form** như hình sau:



- Hình I.17 -

- Sau khi đã có Form trống, ta đặt các thành phần lên Form, thực hiện chỉnh sửa (phần này sẽ được nói kỹ trong chương sau).

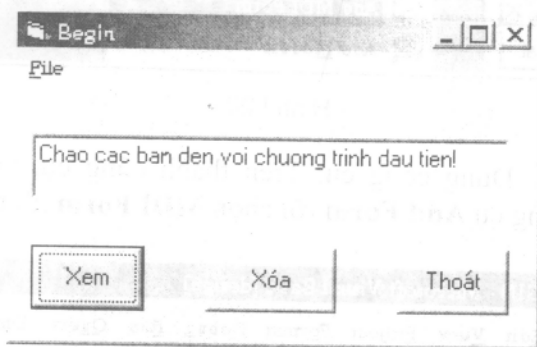
Ví dụ: ta xem xét ví dụ tạo chương trình BEGIN như sau. Khi chương trình BEGIN chạy: một cửa sổ xuất hiện như hình bên dưới. Bên trong cửa sổ có 3 nút lệnh (Command Button): Xem, Xóa và Thoát và một hộp soạn thảo (Text Box) tạm thời để trống.



- Hình I.18 -

➤ Hoạt động của chương trình:

- Khi nhấn nút Xem, hộp soạn thảo sẽ xuất hiện lời chào: “Chào các bạn đến với chương trình đầu tiên!”



- Hình 1.19 -

- Bấm nút Xóa, hộp soạn thảo bị xóa trắng như ban đầu
- Để kết thúc chương trình, ta nhấn nút Thoát.

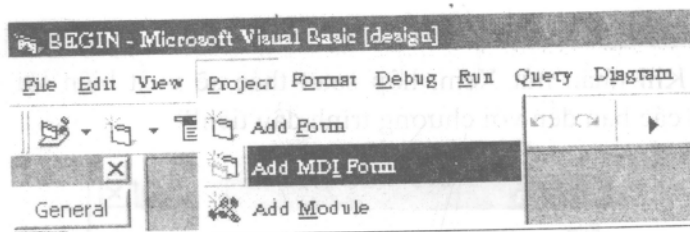
VI.4. MDI Form (Multiple Document Interface)

Đôi khi bạn muốn có một MDI Form, tức là một Form có thể chứa nhiều Form con bên trong. Dạng MDI Form này thường được dùng trong các application như wordprocessor để có thể mở nhiều document cùng một lúc, mỗi document được hiển thị trong một Form con.

➤ Cách tạo MDI Form:

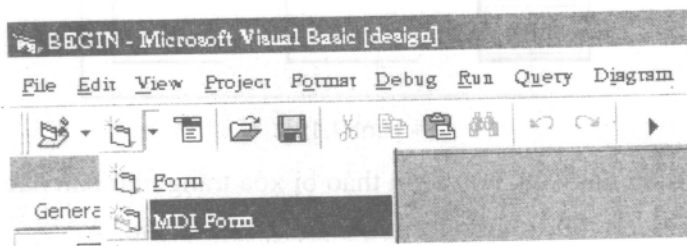
Để có một MDI Form ta có 2 cách:

Cách 1: Dùng Menu: vào trình đơn **Project** → chọn mục **Add MDI Form** như hình sau:



- Hình 1.20 -

Cách 2: Dùng công cụ: Trên thanh công cụ, ta chọn vào công cụ **Add Form** rồi chọn **MDI Form** như hình:



- Hình 1.21 -

Mỗi project VB chỉ có thể có tối đa một MDI Form. Muốn một Form trở thành một Form con bạn **set property MDI Child** của nó thành **True**. Khi chạy (At run-time) bạn không thể dấu một MDIChild Form (hide - biến nó thành invisible), nhưng có thể thu nhỏ (minimize) nó. Nếu bạn thật sự muốn dấu (hide) nó thì phải dùng một mẹo nhỏ là cho nó vị trí (top,left) số âm lớn hơn kích thước nó để nó nằm ngoài tầm hiển thị của Form. Trong một chương trình dùng MDI Form, khi bạn chọn (Click) MDI Form nó không nhảy ra phía trước và che các Form con, nó luôn chứa các form con gói gọn trong cửa sổ của nó (bạn thử click minimize cửa sổ MDI Form sẽ thấy tất cả đều thu nhỏ thành một biểu tượng trên thanh taskbar).