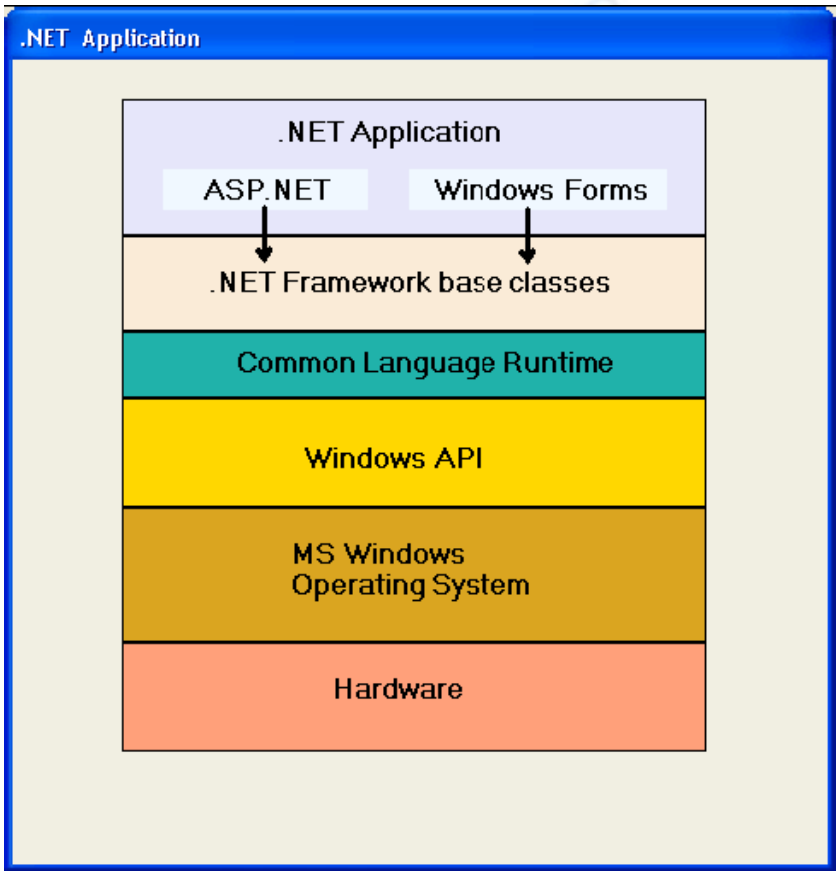
MỤC LỤC

Chương 1. Tìm hiểu môi trường .NET và Visual Basic.NET

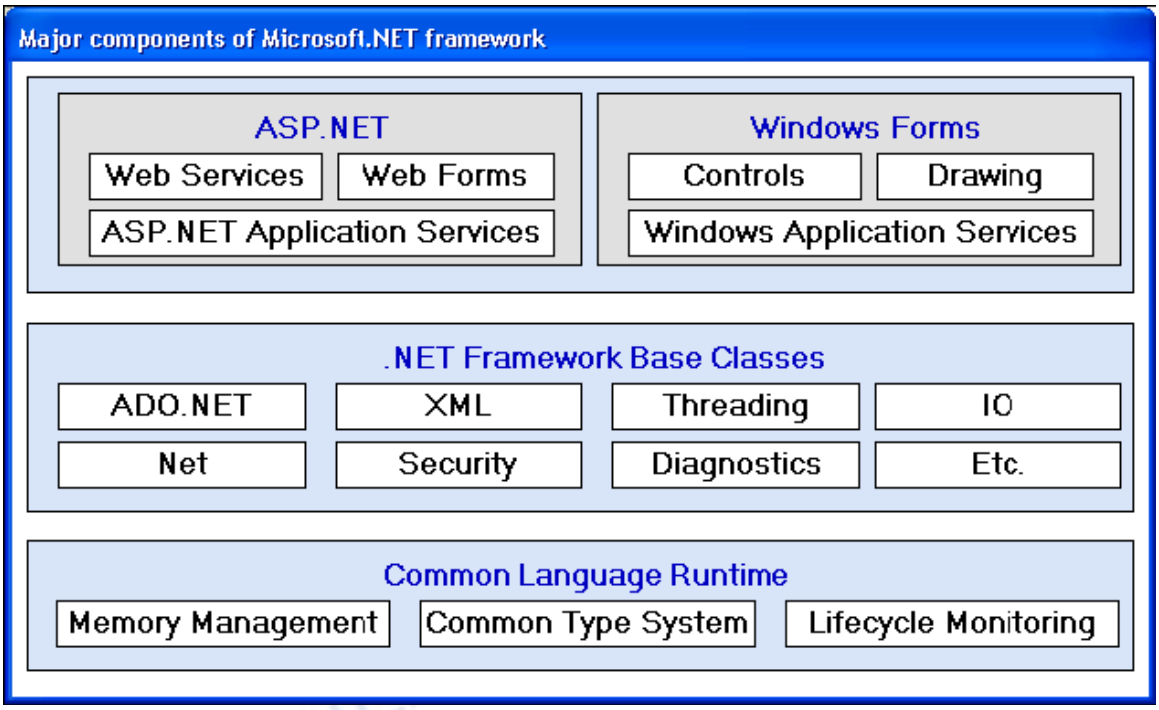
1. Giới thiệu về .NET Framework

.NET được phát triển từ đầu năm 1998, lúc đầu có tên là **Next Generation Windows Services (NGWS)**. Nó là hạ tầng cơ bản được chuẩn hoá, độc lập với ngôn ngữ lập trình, cho phép ta xây dựng, tích hợp, biên dịch, triển khai, chạy các dịch vụ Web, XML, tiện ích hay thực hiện chương trình đa cấu trúc (phát triển bằng các ngôn ngữ lập trình hỗ trợ .NET) trên hệ điều hành có cài đặt .NET Framework. Nó sẽ trở thành một phần của MicroSoft Windows và sẽ được xuất qua các platform khác, có thể ngay cả Unix. Mô hình ứng dụng của .NET Framework



2. Các thành phần của MicroSoft .NET framework

Các phần chính của Microsoft.NET Framework:



.NET Framework bao gồm 2 phần chính là Common Language Runtime (CLR), và .NET Framework Class Library (FCL).

CLR là thành phần chính của .NET Framework, quản lý mã có thể thực thi của chương trình, quản lý các tiến trình, quản lý tiểu trình (Threading), quản lý bộ nhớ, cung cấp dịch vụ để biên dịch, tích hợp và tác vụ truy nhập từ xa (Remoting)

FCL bao gồm tất cả các dịch vụ giao tiếp người sử dụng, điều khiển, truy cập dữ liệu, XML, Threading, bảo mật

3. Tìm hiểu về Metadata

Metadata là một thuật ngữ mà người lập trình sử dụng dùng để mô tả dữ liệu về dữ liệu. Môi trường .NET sử dụng Metadata để mô tả những đối tượng, các gói assembly, dữ liệu ADO.NET. Nét đặc trưng của môi trường .NET là những đối tượng có thể tự mô tả. Một chương trình có thể truy vấn thông tin về một đối tượng để xác định phương thức, thuộc tính, trường mà đối tượng sẵn sàng hỗ trợ cho chương trình sử dụng.

.NET có một file đặc biệt gọi là assembly sử dụng metadata để mô tả thông tin về những lớp mà ứng dụng cung cấp và những thành phần mà ứng dụng sử dụng (như các file dữ liệu), XML định nghĩa metadata, metadata định nghĩa các dữ liệu khác.

4. Tìm hiểu về Common Language Runtime- CRL

Môi trường .NET được xây dựng dựa trên hai nền tảng mạnh và độc lập về ngôn ngữ với thư viện runtime, đó là thư viện thực thi ngôn ngữ chung (Common Language Runtime- CLR) và thư viện lớp (Class Libraries). Vì những thư viện này độc lập với ngôn ngữ, một lập trình viên Visual Basic.NET có cùng một tập hợp hàm, lớp, thủ tục để sử dụng như một lập trình viên C# hay C++.NET.

Mọi ngôn ngữ lập trình đều được biên dịch ra ngôn ngữ trung gian Microsoft Intermediate Language (IL) nhờ bắt buộc mọi ngôn ngữ đều phải dùng cùng các kiểu dữ liệu chung (gọi là Common Type System). Vì vậy, cho phép các ngôn ngữ có thể hợp tác nhau một cách thông suốt. Tức là trong .NET, các chương trình VB.NET có thể kế thừa các chương trình C# và ngược lại một cách hoàn toàn tự nhiên.

5. Tổ chức thư viện đối tượng bằng không gian tên (namespaces)

Trong môi trường lập trình hướng đối tượng, người lập trình thường sử dụng các lớp để mô tả đối tượng. Để tổ chức các lớp tốt hơn và để đơn giản, người lập trình thường nhóm các lớp vào trong thư viện lớp. Thông thường, các lớp trong thư viện lớp sẽ liên hệ với nhau để xử lý một nhiệm vụ nào đó. Khi số lớp trong một thư viện tăng lên, khả năng xung đột tên rất dễ xảy ra. Để giảm bớt xung đột tên tiềm ẩn có thể xảy ra, lập trình viên lên sử dụng không gian tên (namespaces) để tổ chức các lớp đối tượng.

Namespaces là một cách đặt tên để giúp sắp đặt các lớp ta dùng trong chương trình một cách thứ tự để thuận lợi cho việc tìm kiếm chúng. Tất cả mã chương trình trong .NET, viết bằng Visual Basic.NET, C# hay ngôn ngữ nào khác, đều được chứa trong các namespace. Điểm này cũng áp dụng cho thư viện lớp của .NET. Chẳng hạn, các lớp của WinForms đều nằm trong System.Windows.Forms namespace. Và các lớp dùng cho tập hợp như Queue, Stack, Hashtable .v.v.. đều nằm trong System.Collections namespace.

Mỗi namespace chứa những phần của mã lệnh thích hợp cho nó trong chương trình nói chung, trong .NET ta có thể có hai lớp có cùng một tên nhưng nằm trong hai namespaces khác nhau. Ta có thể dùng namespaces bằng cách chỉ ra tên của

namespace rồi đến tên lớp , tiếp theo là tên lớp dẫn xuất …hoặc có thể dùng từ khoá Import để chỉ ra tên namespace chứa lớp mà ta đề cập đến . Nhưng điều tiên quyết là ta phải tham chiếu đến file assembly chứa namespace mà ta muốn dùng. Ta thực hiện việc ấy bằng cách chọn Project | Add References. Khi hộp thoại Add References hiện ra, chọn Tab .NET cho standard .NET components hay Tab Projects cho dữ liệu của một .NET project khác, chọn dữ liệu bạn muốn rồi Click nút Select, sau đó Click nút OK. Chẳng hạn ta muốn sử dụng các phương thức read và write từ studio để nhập dữ liệu và hiển thị dữ liệu trong ứng dụng dạng Console (ứng dụng có giao diện dạng text g iống như MS- DOS trước đây). Namespace ta cần sẽ là System.Console. Trong cách viết thứ nhất ta sẽ viết lệnh như sau để viết hàng chữ "Chào thế giới":

System.Console.WriteLine("Hello world!"). Nếu ta dùng từ khoá Import bằng cách sử dụng câu lệnh Imports System.Console ở đầu phần mã lệnh trong module, ta có thể code gọn hơn: WriteLine ("Hello world!")

Dưới đây là một số namespaces thông dụng:

Namespace	Mục đích
System.CodeDOM	Chứa các lớp mà chương trình có thể sử dụng để biểu diễn tài liệu mã nguồn
System.Colection	Chứa các lớp định nghĩa các cấu trúc dữ liệu dạng tập hợp như: Mảng, danh sách, hàng đợi, ngăn xếp,…
System.Configuration	Chứa các lớp xử lý thông tin cấu hình .NET Framework
System.Data	Chứa các lớp đối tượng cơ sở dữ liệu
System.Diagnostics	Chứa các lớp để chuẩn đoán, gỡ lỗi
System.Drectory.Services	Chứa các lớp dịch vụ thư mục cung cấp khả năng giao tiếp với hệ thống Active Directory
System.Drawing	Chứa các lớp hỗ trợ tính năng đồ hoạ
System.Globalization	Chứa các lớp hỗ trợ khả năng quốc tế hoá như ngày tháng, thời gian, tiền tệ
System.IO	Chứa các lớp hỗ trợ thao tác vào/ra dữ liệu
System.Management	Chứa các lớp quản lý giao diện với dịch vụ thiết bị của

	windows
System.Messaging	Chứa các lớp thực hiện gửi/nhận thông điệp qua hàng đợi
System.NET	Chứa các lớp thực hiện thao tác xử lý mạng
System.Reflection	Chứa các lớp thực hiện truy vấn khả năng của một đối tượng
System.Runtime	Chứa các lớp dùng để gọi tương tác với mã assembly, các đối tượng COM, các đối tượng từ xa
System.Security	Chứa các lớp xử lý về bảo mật và mã hoá
System.Text	Chứa các lớp xử lý ký tự ASCII, Unicode, UTF
System.Timers	Chứa các lớp hỗ trợ sự kiện Timer
System.Web	Chứa các lớp hỗ trợ truy xuất Web
System.Windows.Forms	Chứa các lớp tạo cửa sổ giao diện của ứng dụng dạng Windows Form
System.XML	Chứa các lớp xử lý dữ liệu XML

6. Tìm hiểu mã IL của .NET

Thông thường khi biên dịch chương trình, trình biên dịch sẽ chuyển mã nguồn (source) thành mã máy (các chỉ thị CPU). Các chỉ thị này phụ thuộc vào từng loại CPU, như Intel, Macintosh,...Để hỗ trợ chương trình chạy trên nhiều máy, Java tạo nên một máy ảo. Các chương trình biên dịch ra mã máy ảo và máy ảo của Java chuyển dịch mã máy ảo thành chỉ thị CPU mà máy ảo chạy trên đó. .NET hoạt động tương tự như vậy, các chương trình .NET không biên dịch ra ra mã máy như các file .EXE thông thường, thay vào đó, chương trình được biên dịch ra mã IL (Intermediamte Language- mã trung gian). Chính vì vậy, chương trình .EXE của .NET hoàn toàn không giống như chương trình .EXE của windows trước đây. Những chương trình .EXE của .NET phải chạy trên bộ khung .NET Framework, có thể coi đây là máy ảo dùng diễn dịch mã IL thành chỉ thị CPUđể thực thi chương trình.

Để xem mã IL của chương trình .NET sau khi biên dịch, ta sử dụng chương trình ILDASM.EXE như sau: C:\ildasm <tên_chương_trình.NET>.exe

7. Đóng gói chương trình .NET trong gói Assembly

Trong .NET các chương trình được ngắt ra thành các đơn vị gói (được gọi với thuật ngữ Assembly). Một gói assembly có thể chứa mã chương trình, thư viện lớp hoặc thông tin về đối tượng (được lưu dạng metadata).

File assembly có thể có phần mở rộng là EXE hoặc DLL. Nếu một assembly là thư viện lớp nó có thể dùng riêng cho ứng dụng hoặc dùng chung giữa các ứng dụng. Các assembly dùng chung được đặt trong thư mục C:\WinNT\Assembly

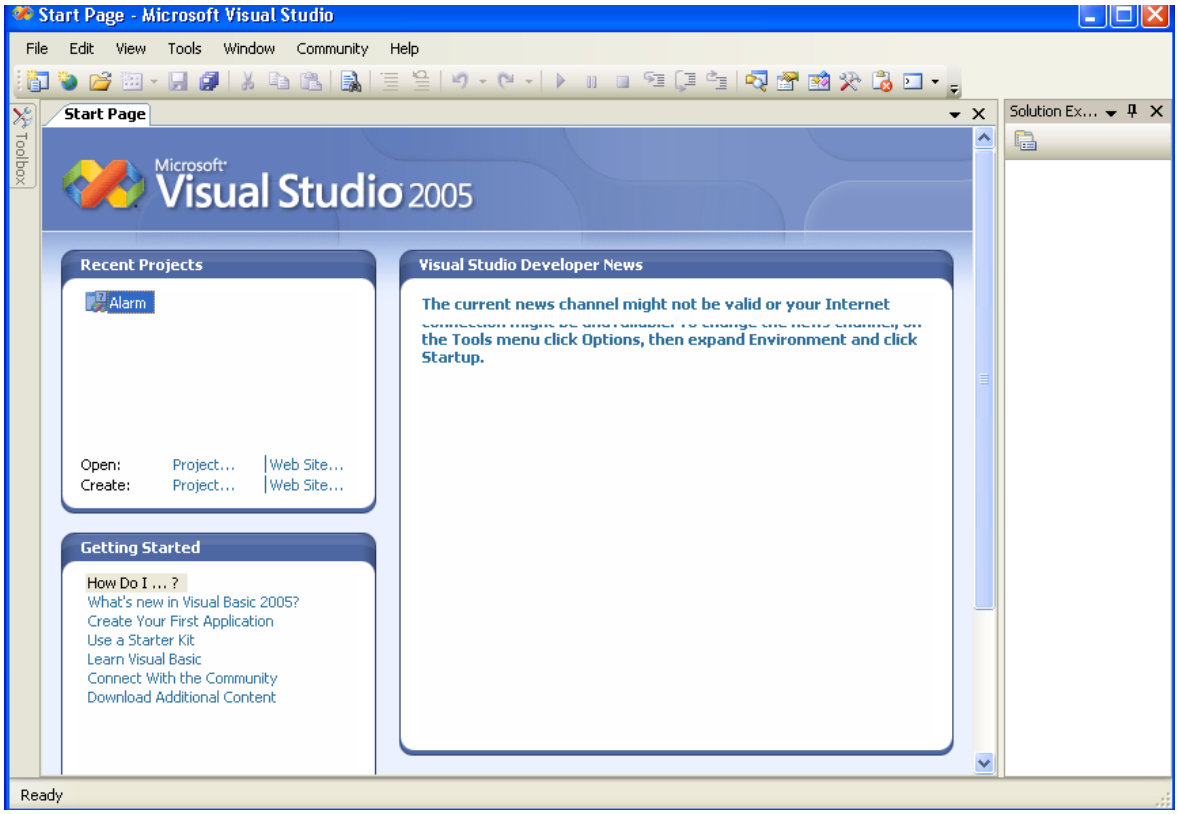
8. Giới thiệu Visual Studio.NET

8.1. Môi trường phát triển Visual studio.NET

Visual Studio.NET là bộ công cụ giúp phát triển các ứng dụng từ ứng dụng desktop đến ứng dụng mạng. Nó là môi trường để phát triển ứng dụng sử dụng tất cả các ngôn ngữ như Visual Basic.NET, Visual C^{++}, Visual C#.Net hết sức mạnh mẽ và thuận lợi.

8.2. Khởi động Visual studio.NET

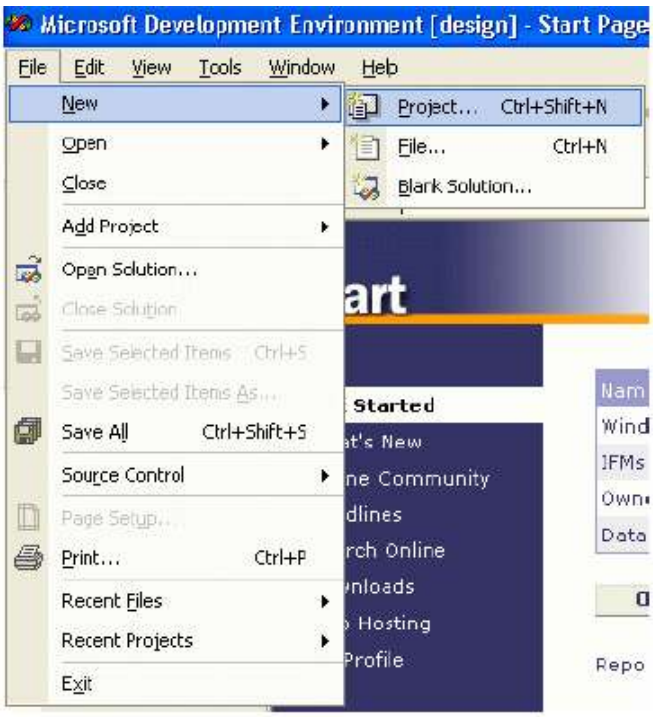
Việc khởi động Visual Studio.NET cũng tương tự như các phần mềm khác như MicroSoft Word hay Excel. Nếu lần đầu khởi động Visual Studio.NET sẽ yêu cầu xem bạn chạy nó với ưu tiên ứng dụng và ngôn ngữ nào. Để sử dụng ngôn ngữ lập trình Visual Basic.NET, bạn chọn Visual Basic và start Visual Studio.NET . Màn hình bắt đầu như sau:



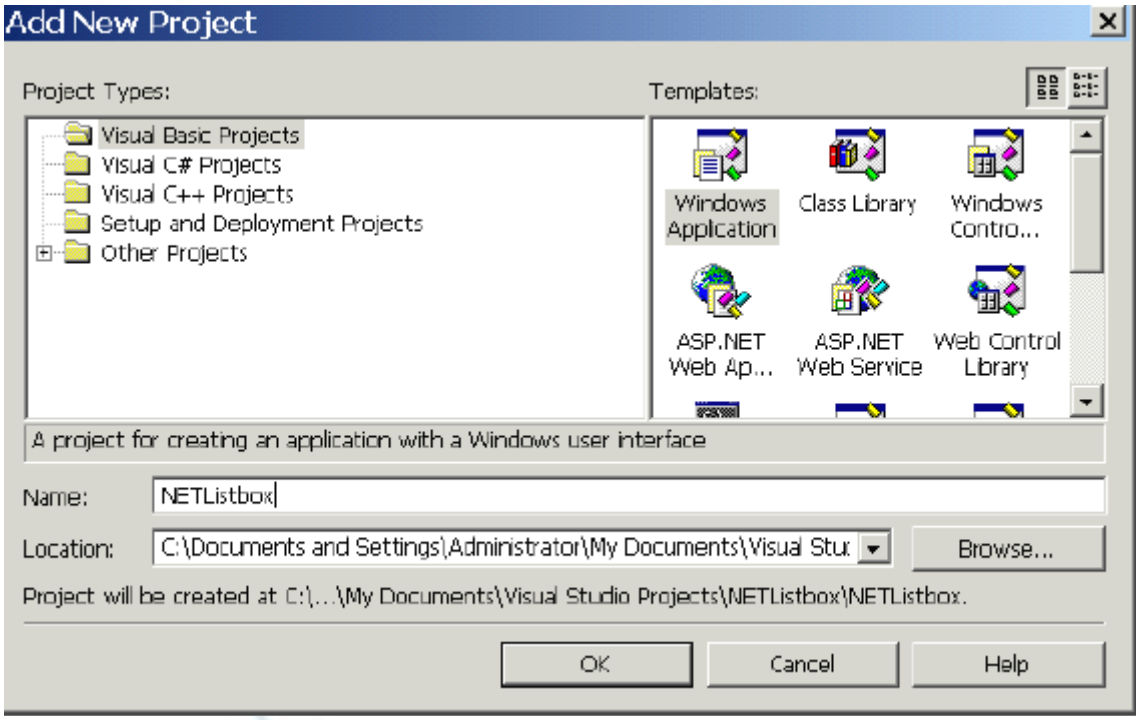
Nếu trang start page không hiện, bạn có thể làm nó xuất hiện bằng cách chọn menu View | Other Windows | Start Page.

8.3. Mở một dự án mới của Visual Basic

- Chọn File/New/ Project



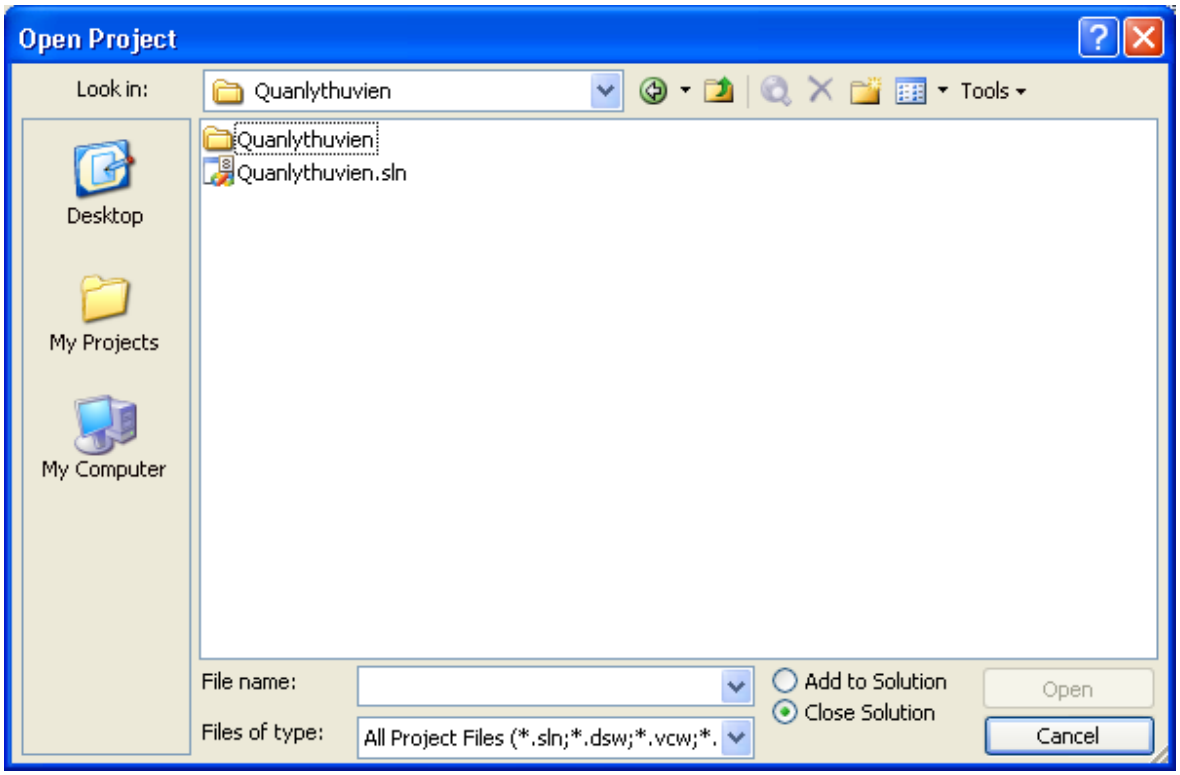
xuất hiện hộp thoại



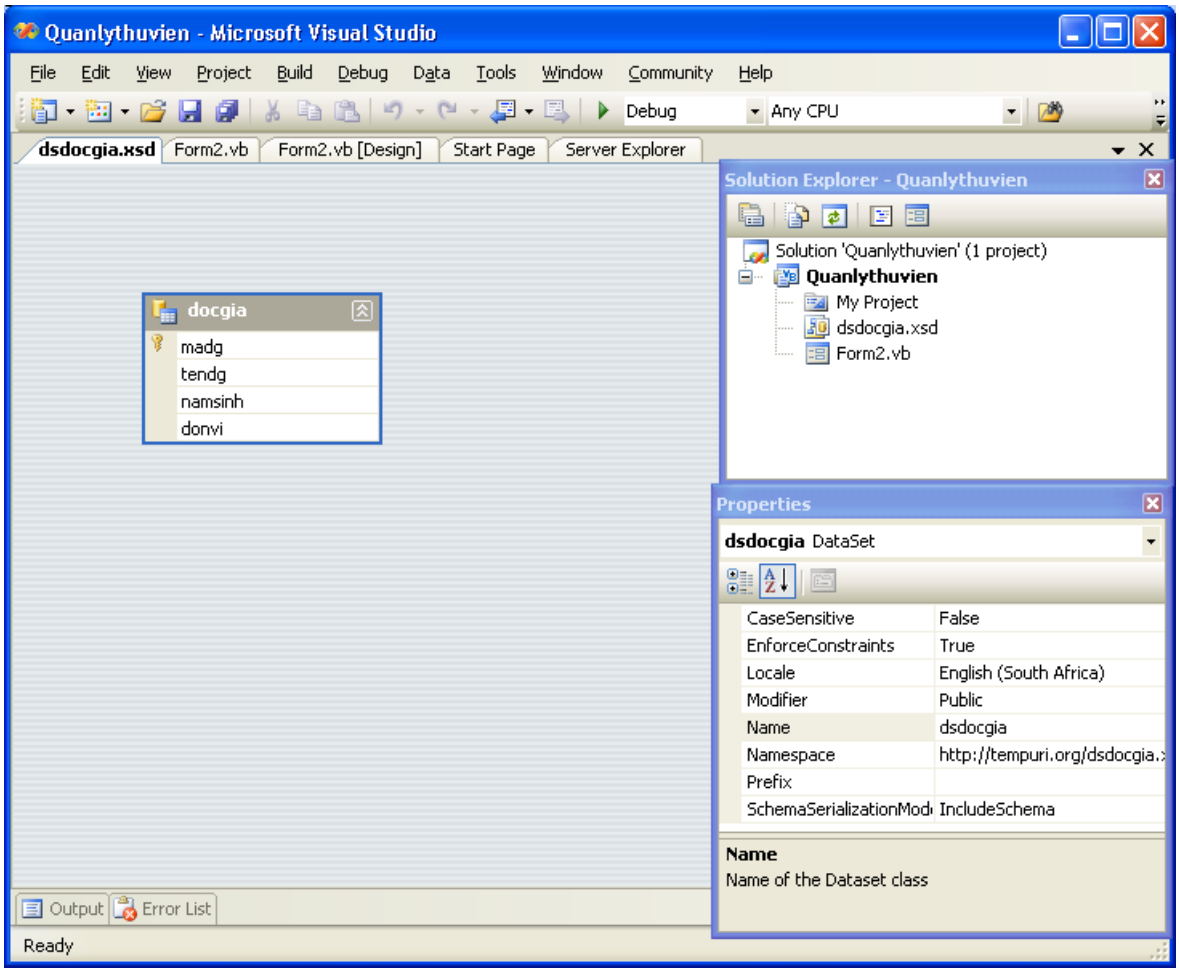
Chọn kiểu ứng dụng rồi Click nút OK

8.4. Mở một dự án cụ thể

- Click chuột vào project (dự án) của phần Open tại trang Start Page.
- Duyệt đến thư mục chứa dự án của bạn



- Mở file Quanlythuvien.sln. Khi này trang Start Page sẽ tạm ẩn đi, một cửa sổ mới xuất hiện:

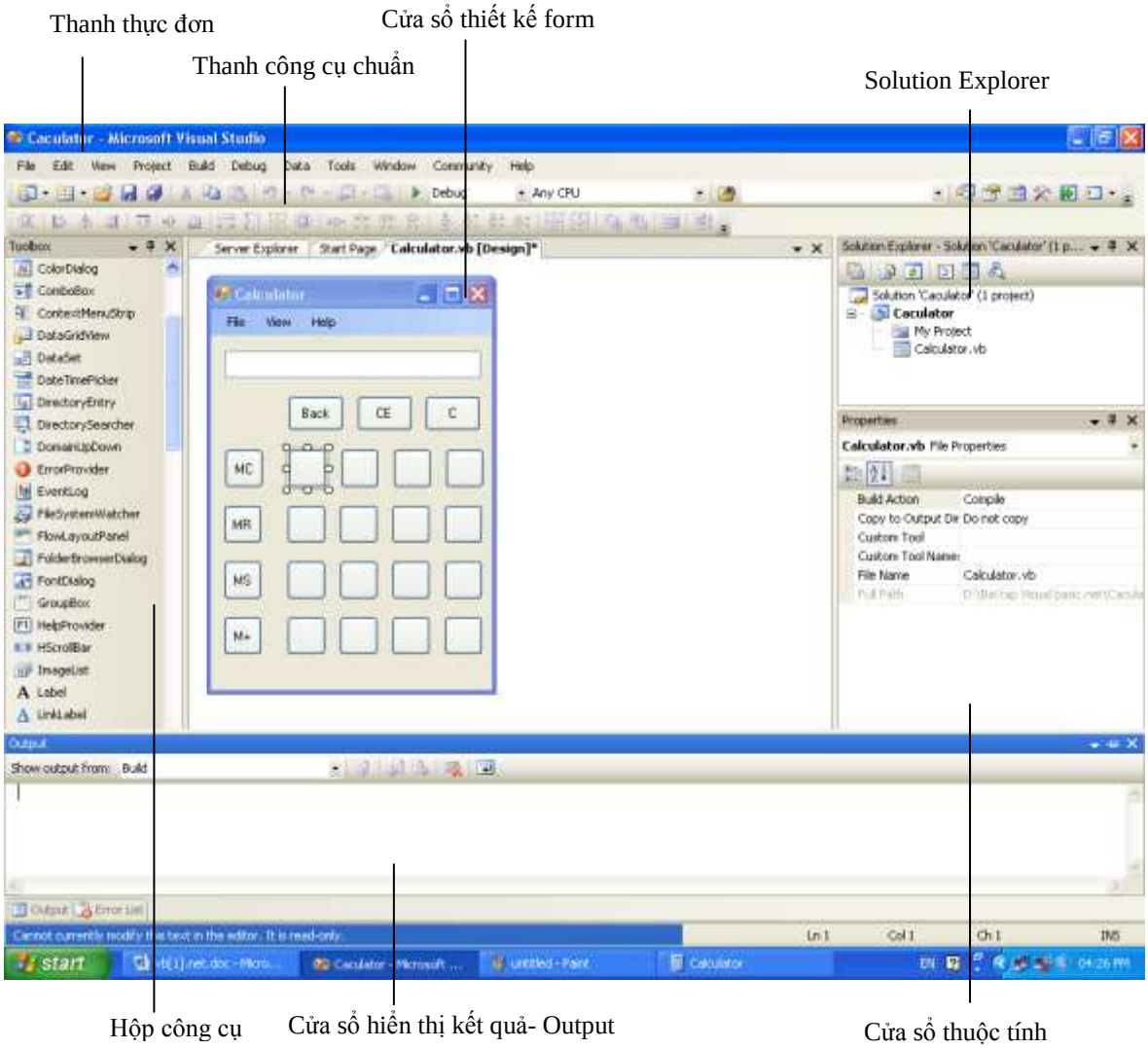


8.5. Dự án (Project) và giải pháp (Solutions)

Trong Visual Studio.NET, các chương trình đang triển khai và phát triển được gọi là dự án (Project) hoặc giải pháp (Solution) bởi chúng chứa rất nhiều file và do nhiều thành phần, đối tượng riêng lẻ hợp lại. Một chương trình Visual Studio.NET bao gồm một file giải pháp và một file dự án hợp lại. File dự án chứa thông tin đặc biệt liên quan đến một tác vụ lập trình đơn lẻ. File giải pháp lại chứa thông tin về một hay nhiều dự án.

8.6. Các công cụ của Visual Studio.NET

Công cụ trong Visual Studio.NET rất phong phú. Sau đây là màn hình làm việc của bộ Visual Studio.NET:



8.6.1 Thanh tiêu đề (Title Bar)

Thanh tiêu đề thể hiện tên của chương trình (project) mặc định là Project1

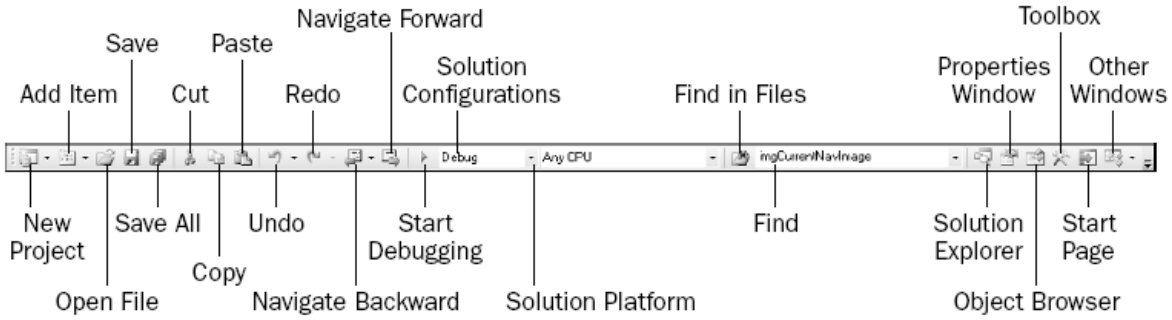
8.6.2 Thanh thực đơn (Menu Bar)

Thanh thực đơn (Menu Bar) xuất hiện ngay dưới thanh tiêu đề, chứa các lệnh đơn để thao tác với Visual Studio. Mỗi lệnh đơn chứa một nhóm lệnh để thực hiện các thao tác trên cùng đối tượng (ví dụ lệnh đơn File chứa các lệnh để thao tác với tệp tin). Thanh thực đơn bao gồm các lệnh đơn sau:

File Edit View Project Build Debug Data Tools Window Community Help

8.6.3 Thanh công cụ (Tool Bar):

Thanh công cụ là thanh chứa các lệnh được thể hiện dưới dạng biểu tượng giúp người sử dụng có thể thi hành một lệnh nhanh hơn. Thanh công cụ bao gồm các nút:



8.6.4 Hộp công cụ (ToolBox)

Hộp công cụ chứa các đối tượng điều khiển (Control) sẽ được đặt vào biểu mẫu (Form) khi thiết kế giao diện của chương trình. Nó được nhóm thành các nhóm công cụ. Để chọn một công cụ trong nhóm, trước tiên ta chọn nhóm công cụ bằng cách Click chuột vào dấu + ở đầu nhóm. Khi đó một danh sách các công cụ trong nhóm sẽ mở ra để ta chọn.

8.7. Bộ thiết kế Windows Forms Designer

Visual Studio.NET có một bộ thiết kế form trực quan, bạn sẽ làm quen dần dần bởi vì thường trong các dự án phát triển có thể có rất nhiều Form.

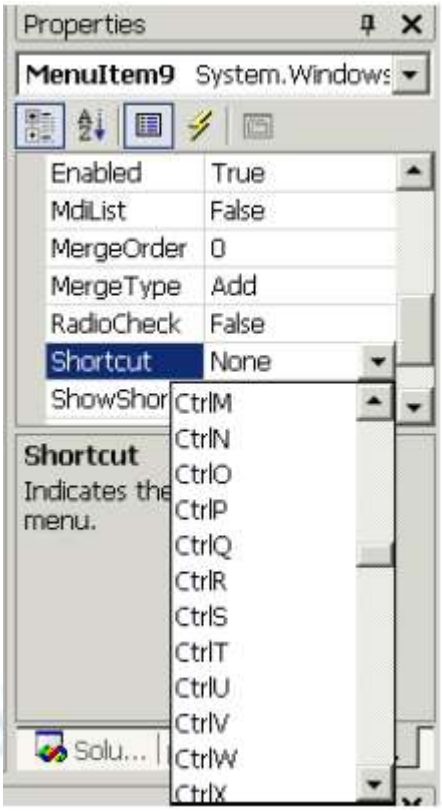
Để hiển thị bộ thiết kế Windows Forms Designer ta hiển thị cửa sổ Solution Explorer bằng cách chọn View | Solution Explorer. Cửa sổ này chứa toàn bộ các phần tử có sử dụng trong dự án. Click chuột vào biểu tượng View Designer trong solution để hiển thị form thiết kế ở dạng trực quan.

8.8. Chạy một chương trình Visual Basic

Click chuột vào nút start màu xanh trên thanh công cụ để chạy chương trình (bạn cũng có thể ấn phím F5).

8.9. Cửa sổ thuộc tính Properties

Cho phép thay đổi thông số của đối tượng thiết kế form sau này.



8.10. Thoát khỏi Visual Studio.NET

- Lưu lại những gì chúng ta làm bằng cách chọn File | Save all.
- Chọn File | Exit để thoát khỏi Visual Studio.NET.

Chương 2. Cơ sở lập trình trong Visual Basic.NET

2.1. Kiểu dữ liệu cơ sở

Chương trình sử dụng các biến để lưu trữ các giá trị. Mỗi biến phải có một kiểu dữ liệu xác định. Một kiểu của biến sẽ xác định các giá trị mà biến đó có thể nhận. Visual Basic cung cấp các kiểu dữ liệu cơ bản sau:

Tên kiểu	Phạm vi giá trị	Kích thước (byte)
Boolean	True, False	2
Byte	Các số nguyên trong đoạn từ 0 đến 255	1
Char	Các ký tự Unicode 16 bits	2
Decimal	Các số nguyên trong đoạn từ - 2^{95} đến $2^{95} – 1$	16
Double	Các số thực dấu phẩy động 64 bits	8
Integer	Các số nguyên trong đoạn từ - 2^{31} đến $2^{31} – 1$	4
Long	Các số nguyên trong đoạn từ - 2^{63} đến $2^{63} – 1$	8
Short	Các số nguyên trong đoạn từ - 2^{31} đến $2^{31} – 1$	4
Single	Các số thực dấu phẩy động 32 bits	4

Một thay đổi lớn cho Data Type của VB.NET, là những biến dùng Data Type cơ bản như Integer, Single, Boolean,.v.v.. đều là những biến kiểu đối tượng (Objects). Chúng đều được xuất phát từ lớp căn bản nhất tên Object trong VB.NET.

2.2. Khai báo biến

2.2.1. Khai báo nhiều biến

Biến được sử dụng để lưu trữ và đọc thông tin trong quá trình thực hiện chương trình. Để khai báo biến ta dùng từ khoá Dim và chỉ ra tên biến cùng với kiểu dữ liệu của biến. Trong VB 6.0 ta có thể khai báo nhiều biến trên cùng một hàng như:

```
Dim i, j, k As Integer
```

Kết quả là chỉ có k là Integer, còn i và j là Variant. Trong VB.NET thì cả ba i, j và k đều là Integer.

Ta có vừa khai báo biến vừa khởi gán giá trị cho biến.

Ví dụ :

Dim X As Integer = 12

2.2.1. Phạm vi hoạt động của biến

Tuỳ theo vị trí khai báo biến và các từ khoá quy định phạm vi mà ta sử dụng biến sẽ có phạm vi hoạt động khác nhau.

VB.NET cho phép khai báo biến trong một khối lệnh IF ..THEN...END IF, DO...LOOP, WHILE...END WHILE, FOR...NEXT,..v.v..vì vậy, những biến này chỉ có phạm vi hoạt động ở trong khối lệnh đó. Khi chương trình thực hiện xong khối lệnh đó thì các biến cũng bị hủy

Ngoài ra, trong VB.NET ta có thể quyết định giới hạn việc truy cập một biến bằng cách dùng các từ khoá sau:

Loại truy cập	Thí dụ	Chú thích
Public	Public Class ClassForEverybody	Cho phép ở đâu cũng dùng nó được. Ta chỉ có thể dùng Public ở mức độ Module, Namespace hay File. Tức là ta không thể dùng Public trong một Sub/Function.
Protected	Protected Class ClassForMyHeirs	Cho phép các classes con, cháu được dùng. Ta chỉ có thể dùng Protected ở mức độ Class.
Friend	Friend StringForThisProject As String	Cho phép code trong cùng một Project được dùng.
Private	Private NumberForMeOnly As Integer	Cho phép code trong cùng module, class, hay structure được dùng. Lưu ý là **Dim** coi như tương đương với **Private**, do đó ta nên dùng Private cho dễ đọc.

2.3. Khai báo hằng (Constants)

Hằng là đại lượng mà giá trị không thay đổi trong suốt quá trình thực hiện chương trình. Khi khai báo hằng trong VB.NET ta phải khai rõ kiểu dữ liệu của nó là String, Integer, hay Boolean ..v.v.:

Public Const myConstantString As String = "happy"

Public Const maxStudent As Integer = 30

2.4. Toán tử

2.4.1. Toán tử số học

Là các toán tử có các toán hạng dạng số và kết quả thực thu được cũng có dạng số, bao gồm

Toán tử	Giải thích
+	Phép cộng
-	Phép trừ
*	Phép nhân
/	Phép chia
\	Phép chia lấy phần nguyên
^	Phép luỹ thừa
Mod	Phép chia lấy phần dư

2.4.1. Toán tử so sánh

Là các toán tử có kết quả thu được sau khi thực hiện là giá trị logic (True hoặc False), bao gồm:

Toán tử	Giải thích
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng
=	Bằng
<>	Khác
Like	So khớp

2.4.2. Toán tử gán

Toán tử gán dùng để gán giá trị cho một biến, một phần tử mảng. Toán tử gán được ký hiệu bằng dấu bằng (=).

Ví dụ: