

LẬP TRÌNH GIAO DIỆN

Nguyễn Thị Mai Trang

Nguyễn Thị Mai Trang

1

1

Chương 1

Tổng quan về lập trình giao diện với Visual Studio.Net

2

Mục tiêu

- Qua chương này, sinh viên có khả năng:
 - Phân biệt được các tầng và chức năng mỗi tầng trong quá trình thiết kế xây dựng phần mềm, đặc biệt là tầng giao diện.
 - Phân biệt được các loại giao diện ứng dụng để có thể lựa chọn loại giao diện phù hợp với yêu cầu sử dụng của phần mềm.
 - Biết và sử dụng được công nghệ VisualStudio.Net của Microsoft và C#2010
 - Làm quen và xây dựng được một ứng dụng đơn giản trong môi trường Visual Studio.Net

Nguyễn Thị Mai Trang

3

3

NỘI DUNG

- Tổng quan về Lập trình giao diện
- Giới thiệu công nghệ .NET của Microsoft

Nguyễn Thị Mai Trang

4

4

1.1 Tổng quan về lập trình giao diện

- Khái niệm về mô hình đa tầng, đa lớp trong việc phát triển phần mềm
- Lớp giao diện
- Các quy tắc cơ bản của giao diện phần mềm
- Thiết kế chương trình đa lớp

Nguyễn Thị Mai Trang

5

5

1.1.1 Khái niệm về mô hình đa tầng, đa lớp trong việc phát triển phần mềm

- Mô hình ba tầng (3-tier)
 - “là một kiến trúc kiểu client/server mà trong đó giao diện người dùng (UI - user interface), các quy tắc xử lý (BR - Business Rule hay BL - Business Logic), và việc lưu trữ dữ liệu được phát triển như những module độc lập, và hầu hết là được duy trì trên các nền tảng độc lập, và mô hình 3 tầng được coi là một kiến trúc phần mềm và là một mẫu thiết kế.”

Nguyễn Thị Mai Trang

6

6

Khái niệm về mô hình đa tầng, đa lớp

- Mô hình ba tầng bao gồm:
 - Presentation tier: gồm các thành phần xử lý giao diện - Graphic User Interface (GUI)
 - Business tier: gồm các thành phần xử lý nghiệp vụ như
 - Business Logic Layer (BLL)
 - Data Access Layer (DAL)
 - Data Transfer Object (DTO)
 - Data tier: nơi lưu trữ dữ liệu, là các hệ quản trị CSDL như MS SQL Server, Oracle, SQLite, MS Access, XML files, text files,...

Nguyễn Thị Mai Trang

7

7

Khái niệm về mô hình đa tầng, đa lớp

- Mô hình đa tầng phân chia ứng dụng theo mức vật lý
- Ưu điểm của mô hình đa tầng:
 - Dễ dàng mở rộng, thay đổi quy mô của hệ thống
- Nhược điểm:
 - Truyền dữ liệu giữa các tầng chậm vì phải truyền giữa các tiến trình khác nhau
 - Phát triển ứng dụng khá phức tạp
 - → áp dụng trong các hệ thống lớn

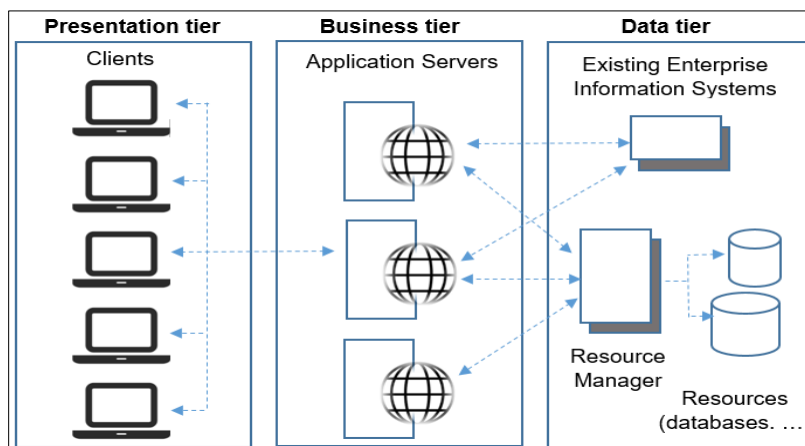
Nguyễn Thị Mai Trang

8

8

Khái niệm về mô hình đa tầng, đa lớp

• Mô hình ba tầng



Nguyễn Thị Mai Trang

9

9

Khái niệm về mô hình đa tầng, đa lớp

• Mô hình ba lớp (3-layer)

- Phân chia ứng dụng về mặt logic dựa vào chức năng.
- Các lớp khác nhau khi được thực thi có thể nằm trong cùng một vùng bộ nhớ của một tiến trình.
- Mô hình ba lớp bao gồm:
 - **Presentation Layer:** Thành phần giao diện, là các form của chương trình tương tác với người sử dụng.
 - **Business Layer:** Lớp xử lý các nghiệp vụ của chương trình như xử lý tính toán, xử lý ngoại lệ, an toàn dữ liệu,...
 - **Data Access Layer:** Lớp giao tiếp với các hệ quản trị cơ sở dữ liệu.

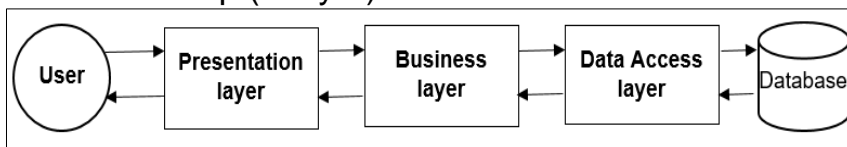
Nguyễn Thị Mai Trang

10

10

Khái niệm về mô hình đa tầng, đa lớp

- Mô hình ba lớp (3-layer)



- Điểm khác nhau cơ bản giữa mô hình đa tầng và mô hình đa lớp:
 - Mô hình ba tầng có tính vật lý, có quy mô lớn hơn
 - Mô hình ba lớp có tính logic, trong một tầng có thể có nhiều lớp.

Nguyễn Thị Mai Trang

11

11

Khái niệm về mô hình đa tầng, đa lớp

- Lợi ích của mô hình đa lớp:
 - Phát triển phần mềm: chuyên nghiệp hóa, có thể chia cho nhiều nhóm được đào tạo nhiều kỹ năng khác nhau, như thiết kế mỹ thuật, lập trình, tổ chức cơ sở dữ liệu,...
 - Bảo trì phần mềm: các thành phần của một hệ thống dễ được thay đổi, không ảnh hưởng hoặc chỉ ảnh hưởng đến lớp ngay gần kề của nó, không phải viết lại cả chương trình.
 - Mở rộng phần mềm: thêm các chức năng mới cho từng lớp dễ dàng hơn.

Nguyễn Thị Mai Trang

12

12

1.1.2 Lớp giao diện (Presentation layer)

- Chứa các thực thể phần mềm làm nhiệm vụ hiển thị.
- Cung cấp các đối tượng có thể tương tác với người sử dụng, mang tính trực quan cao như:
 - Hình ảnh
 - biểu tượng
 - ký hiệu,...
- → mang lại sự tiện lợi, đơn giản và hiệu quả.

Nguyễn Thị Mai Trang

13

13

Lớp giao diện (tt)

- Một số ví dụ về lớp giao diện:
 - Ứng dụng giao diện dòng lệnh.
 - Ứng dụng windows form: các form, menu, thanh công cụ, các nút lệnh, các ô nhập liệu,...
 - Ứng dụng web: các form đăng ký, đăng nhập, lựa chọn mua hàng,...
 - Ứng dụng trên thiết bị di động: các biểu tượng, hình ảnh,...

Nguyễn Thị Mai Trang

14

14

1.1.3 Các quy tắc cơ bản của giao diện phần mềm

- Tầm quan trọng của giao diện phần mềm:
 - Giao diện là bộ mặt của phần mềm → cần rõ ràng, sáng sủa, thân thiện.
 - Giao diện là nơi cung cấp chức năng cho người sử dụng → cần có đầy đủ các chức năng, dễ sử dụng và sử dụng an toàn.
 - Giao diện là nơi cung cấp các trợ giúp → cần có đầy đủ các hướng dẫn sử dụng, kịp thời và hiệu quả.
 - Phần mềm là công cụ sử dụng hàng ngày → cần có khả năng tùy biến và linh hoạt để tránh nhàm chán

Nguyễn Thị Mai Trang

15

15

Các quy tắc cơ bản của giao diện phần mềm (tt)

- Các yêu cầu của giao diện phần mềm:
 - Tính thân thiện: không gây cảm giác lúng túng, mất tự tin khi sử dụng → các thành phần phải được trình bày rõ ràng và sắp xếp có thứ tự theo quy trình nghiệp vụ.
 - Tính dễ sử dụng: danh mục, thanh công cụ, biểu tượng, văn bản minh họa → gần gũi với nhận thức của người sử dụng
 - Tính tin cậy: xác suất hoạt động không thất bại của phần mềm trong thời gian và môi trường xác định.

Nguyễn Thị Mai Trang

16

16

1.1.4 Thiết kế chương trình đa lớp

- Xác định các yêu cầu, chức năng của ứng dụng.
- Phân loại và nhóm các chức năng của ứng dụng.
- Xác định các thành phần của ứng dụng.
- Thiết kế cấu trúc dữ liệu, cơ sở dữ liệu.
- Xây dựng các lớp truy xuất dữ liệu.
- Xây dựng lớp giao diện tương tác với người sử dụng và xử lý các chức năng

Nguyễn Thị Mai Trang

17

17

1.2. Giới thiệu công nghệ .NET của Microsoft

- Giới thiệu Microsoft Visual Studio.NET
- .NET Framework và Common Language Runtime
- Các ngôn ngữ trong Visual Studio.NET
- Các loại project trong Visual Studio.Net
- Tạo project mới trong Visual Studio
- Các thành phần trong một project
- Trang Start Page
- Minh họa xây dựng ứng dụng với Visual C#

Nguyễn Thị Mai Trang

18

18

1.2.1 Giới thiệu Microsoft Visual Studio.NET

- Công nghệ .NET của Microsoft được công bố vào năm 2000
- Cho phép tạo các ứng dụng .Net với nhiều loại ngôn ngữ như C#, Visual Basic, và những ngôn ngữ khác.
- Công nghệ ASP.NET cho phép tạo ra các ứng dụng và dịch vụ Web một cách trực quan.



Nguyễn Thị Mai Trang

19

19

Giới thiệu Microsoft Visual Studio.NET (tt)

- Các phiên bản của Visual Studio.NET
 - Visual Studio 2002 (12/2/2002): .NET Framework 1.0
 - Visual Studio 2003 (24/4/2003): .NET Framework 1.1
 - Visual Studio 2005 (10/2005): .NET Framework 2.0
 - Visual Studio 2008 (11/2007): .NET Framework 3.5
 - Visual Studio 2010 (12/4/2010): .NET Framework 4.0
 - Visual Studio 2012 (15/8/2012): .NET Framework 4.5
 - Visual Studio 2013: .NET Framework 4.5.2, hỗ trợ các công nghệ mới như Windows Phone, Windows Store,...
 - Visual Studio 2015: .NET Framework 4.6, hỗ trợ đa nền tảng, cho phép xây dựng ứng dụng chạy trên hệ điều hành Linux, Mac OS, lập trình ứng dụng cho Android, iOS.
 - Visual Studio 2017: .NET Framework 4.7
 - Visual Studio 2019: NET Core 3.0
 - Visual Studio 2022: 64 bit

Nguyễn Thị Mai Trang

20

20

Giới thiệu Microsoft Visual Studio.NET (tt)

- Mỗi phiên bản Visual Studio.NET tương thích với một version .NET Framework.
- Ví dụ, một chương trình được viết với Visual Studio 2015 chỉ có thể chạy được trên hệ điều hành có cài .NET Framework 4.6.
- Các phiên bản Visual Studio .NET của Microsoft liên tục được cập nhật và phát triển.

Nguyễn Thị Mai Trang

21

21

1.2.2 .NET Framework và Common Language Runtime

- .NET Framework là một thành phần quan trọng trong công nghệ .NET, là nền tảng cho phép phát triển các loại ứng dụng:
 - Desktop: Windows Form Application, ứng dụng văn phòng (office/sharepoint), Windows Presentation Foundation (WPF).
 - Windows Store: các ứng dụng dành cho thiết bị cảm ứng.
 - Web: các ứng dụng web (ASP.NET), dịch vụ web (Web Service), Silverlight.
 - Windows Communication Foundation(WCF) services: các ứng dụng dạng dịch vụ chia sẻ dữ liệu qua mạng, internet...

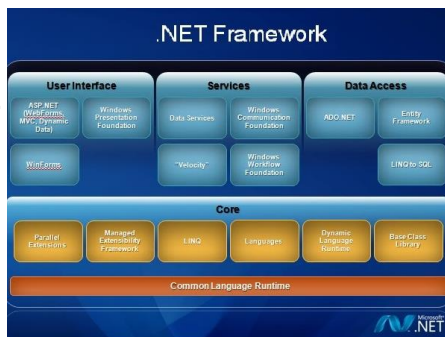
Nguyễn Thị Mai Trang

22

22

.NET Framework và Common Language Runtime (tt)

- Bao gồm hai thành phần chính:
 - Lớp thư viện Framework (Framework Class Library)
 - Bộ thực thi ngôn ngữ chung (CommonLanguageRuntime).



Nguyễn Thị Mai Trang

23

23

Framework Class Library

- Là thư viện chứa các lớp cho phép xây dựng các loại ứng dụng với các ngôn ngữ khác nhau.
 - Base class library: thư viện các lớp cơ bản nhất, ví dụ: String, Integer, Exception, ...
 - ADO.NET và XML: thư viện các lớp dùng để xử lý dữ liệu.
 - ASP.NET: thư viện các lớp dùng để xây dựng ứng dụng web.
 - Web services: thư viện các lớp dùng để xây dựng các dịch vụ được cung cấp qua mạng và internet.
 - Windows Form: thư viện các lớp đối tượng dành cho việc xây dựng các ứng dụng có giao diện đồ họa.

Nguyễn Thị Mai Trang

24

24

Common Language Runtime (CLR)

- Là thành phần kết nối giữa các phần trong .NET Framework với hệ điều hành.
- Các thành phần bên trong CLR:
 - Class loader
 - Just In Time compiler
 - Garbage collector
 - Exception handler
 - COM marshaller
 - Security engine
- Các chức năng của CLR:
 - Bảo vệ bộ nhớ
 - Xử lý lỗi
 - Thu gom rác tự động
 - Kiểm tra tính an toàn, không cho phép các lệnh nguy hiểm được thi hành.

Nguyễn Thị Mai Trang

25

25

Common Language Runtime (CLR)

- Máy tính cài đặt .NET Framework có thể thực thi một chương trình .NET có version phù hợp.
- Khả năng tương tác giữa các ngôn ngữ:
 - Chương trình có thể được viết bằng ngôn ngữ khác nhau (C#, Visual Basic,...)
 - → biên dịch thành MSIL.
 - → .NET Framework trở thành một ngôn ngữ độc lập, không gắn với một ngôn ngữ lập trình cụ thể.

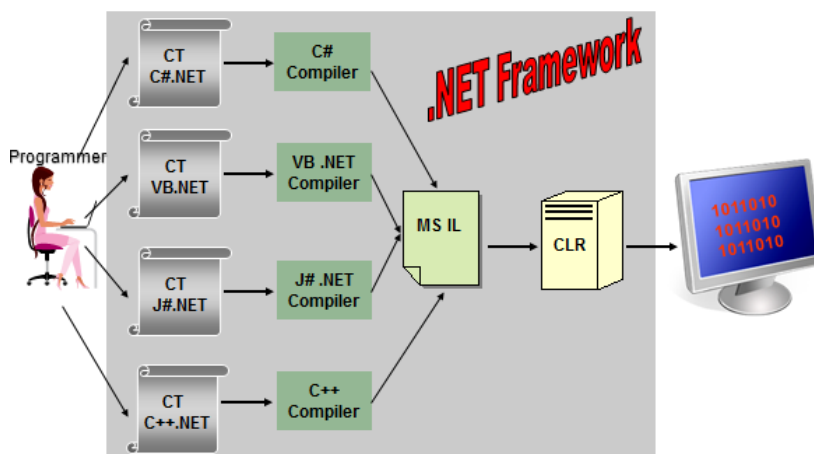
Nguyễn Thị Mai Trang

26

26

Common Language Runtime (CLR)

- Quá trình biên dịch và thực thi một chương trình .NET



Nguyễn Thị Mai Trang

27

27

1.2.3 Các ngôn ngữ trong VS

- Visual C#: ngôn ngữ mới, hỗ trợ lập trình trực quan, hướng sự kiện, cú pháp tương tự C++.
- Visual Basic: được phát triển từ BASIC, một ngôn ngữ lập trình đơn giản, dành cho người mới học về kỹ thuật lập trình cơ bản, hỗ trợ lập trình trực quan, hướng sự kiện, chỉ có trong .NET.
- Visual C++: nền tảng là C++, cho phép phát triển các ứng dụng với thư viện MFC, .NET.
- Visual F#: ngôn ngữ lập trình hàm dành cho nhà phát triển .NET, mới được tích hợp vào Visual Studio.
- Python: ngôn ngữ lập trình dạng script.

Nguyễn Thị Mai Trang

28

28

1.2.4 Các loại project trong Visual Studio.Net

- Windows Forms Application: ứng dụng dạng giao diện đồ họa.
- Class Library: tạo các thư viện liên kết động dùng trong các ứng dụng khác.
- ASP .NET Web Application / Service / Control Library: các ứng dụng về web, dịch vụ web.
- Console Application: ứng dụng dạng giao diện dòng lệnh.
- Windows Service: các ứng dụng cung cấp dịch vụ chạy trên máy server.
- Windows Presentation (WPF): ứng dụng giao diện đồ họa trên nền tảng XAML.

Nguyễn Thị Mai Trang

29

29

Các loại project trong Visual Studio.Net (tt)

- Windows Communication Foundation Application (WCF): các ứng dụng hướng dịch vụ trên nền Windows.
- Silverlight Application, Silverlight Class Library Others: ứng dụng đa phương tiện chạy trên nền web.
- Excel WorkBook, Word Document: các ứng dụng văn phòng tương tự Microsoft Word, Excel.
- Cloud: các ứng dụng điện toán đám mây.
- Android, IOS, Windows Phone: ứng dụng trên thiết bị di động.

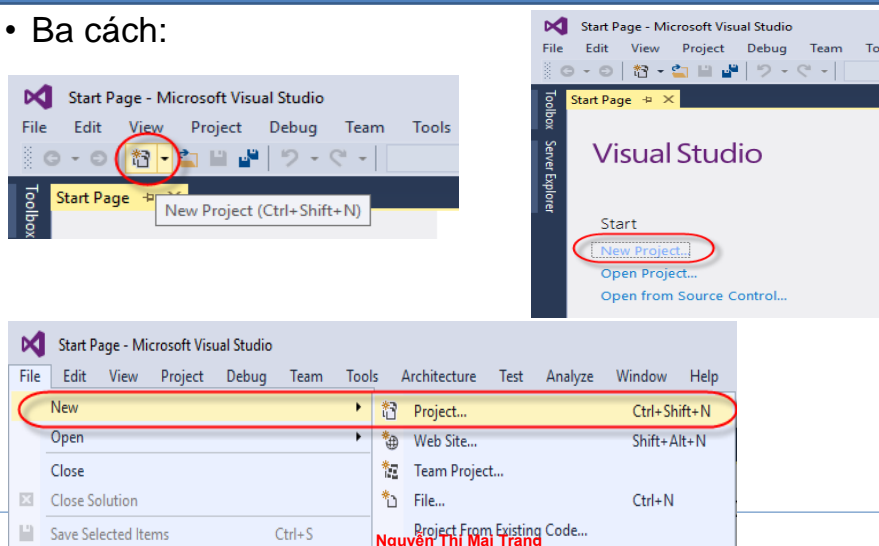
Nguyễn Thị Mai Trang

30

30

1.2.5 Tạo project mới trong Visual Studio

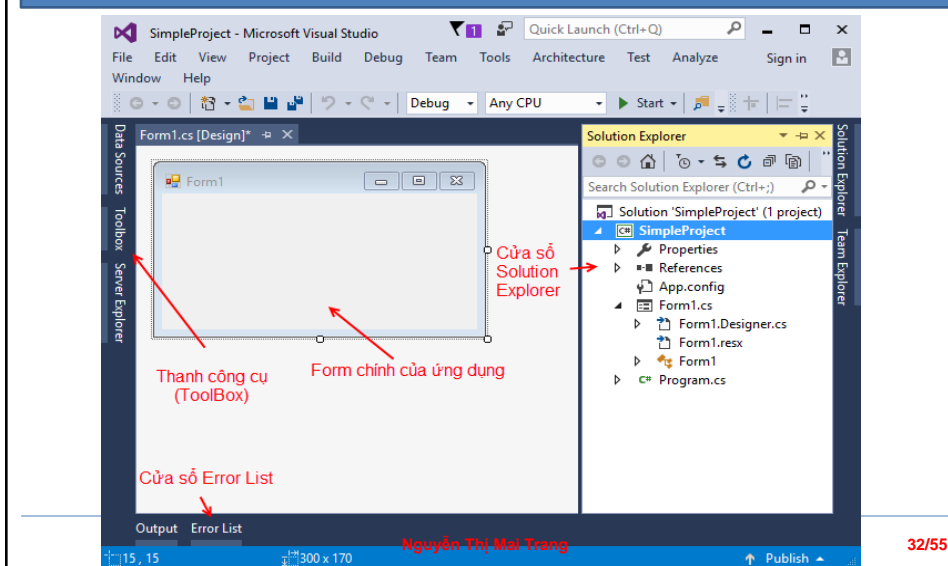
- Ba cách:



31

31

1.2.6 Các thành phần trong một project



32/55

32

Các thành phần trong một project (tt)

- Form1: cửa sổ chính của ứng dụng
- Cửa sổ Solution Explorer: chứa các thành phần có trong project, bao gồm Project References, Forms, Classes, Modules, Folder với các items
- Error List: hiển thị kết quả biên dịch và thông báo lỗi (nếu có)

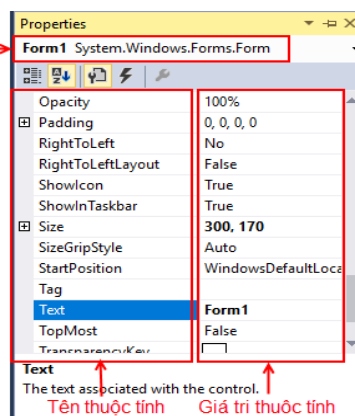
Nguyễn Thị Mai Trang

33

33

Các thành phần trong một project (tt)

- Cửa sổ Properties: cho phép thiết lập các thuộc tính cho Form và controls

 Tên form
hoặc control


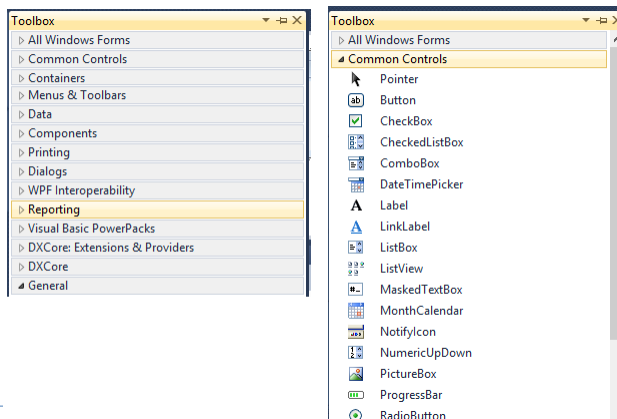
Nguyễn Thị Mai Trang

34

34

Các thành phần trong một project (tt)

- Thanh Toolbox: Chứa các control được tổ chức thành các nhóm chức năng, cho phép thiết kế giao diện bằng cách kéo thả các control lên Form



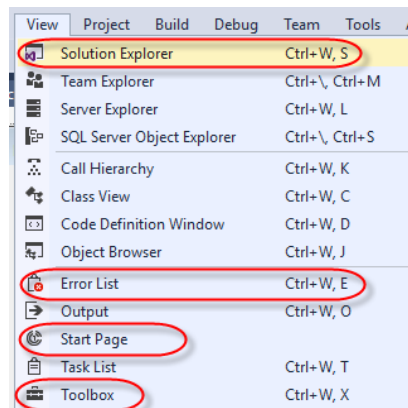
Nguyễn Thị Mai Trang

35

35

Các thành phần trong một project (tt)

- Cửa sổ Server Explorer: Quản lý các kết nối tới cơ sở dữ liệu, hiển thị và quản lý các phần tử của Server trong VS.Net IDE như Event Logs, Message Queues,...
- Hiển thị/ ẩn các thành phần: menu View



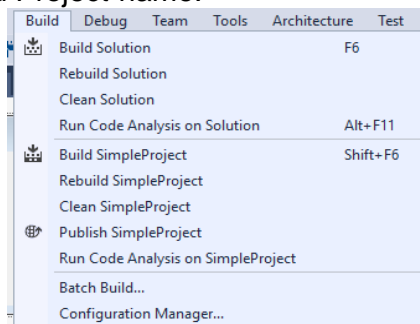
Nguyễn Thị Mai Trang

36

36

Các thành phần trong một project (tt)

- Biên dịch: menu Build
 - Build Solution / Rebuild Solution: biên dịch/biên dịch lại tất cả các project trong ứng dụng
 - Build Project-name /Rebuild Project-name: biên dịch project có tên Project-name



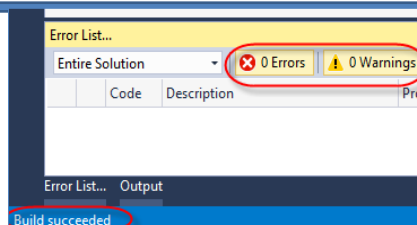
Nguyễn Thị Mai Trang

37

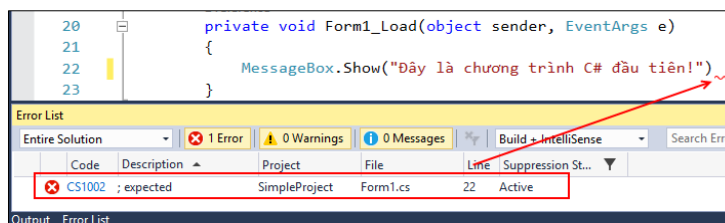
37

Các thành phần trong một project (tt)

- Không có lỗi:



- Có lỗi: double click vào dòng thông báo lỗi để chuyển đến vị trí bị lỗi



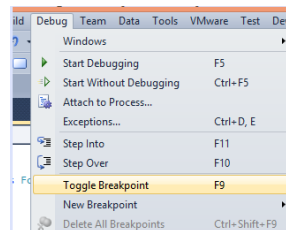
Nguyễn Thị Mai Trang

38

38

Các thành phần trong một project (tt)

- Debug (chạy từng bước):
 - Chạy chương trình theo từng bước để theo dõi giá trị các biến, các đối tượng, được sử dụng rất hiệu quả trong trường hợp chương trình bị lỗi logic, cho kết quả không đúng.
 - Chạy debug: đặt các break point tại vị trí muốn kiểm tra bằng cách chuyển dấu nháy đến mã lệnh, trên menu Debug, chọn Toggle Breakpoint hoặc nhấn phím F9



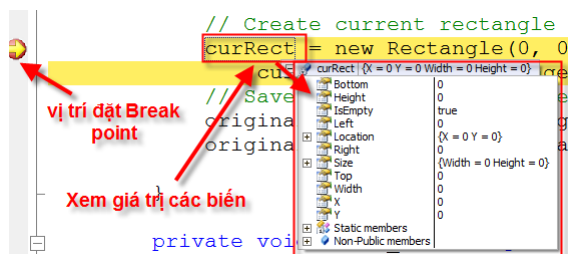
Nguyễn Thị Mai Trang

39

39

Các thành phần trong một project (tt)

- Debug (chạy từng bước):
 - Khi chương trình thực thi, trình biên dịch sẽ dừng lại tại vị trí đặt breakpoint.
 - Để chuyển đến dòng lệnh kế tiếp, nhấn phím F10, để chuyển đến phần nội dung của một hàm, nhấn F11.



Nguyễn Thị Mai Trang

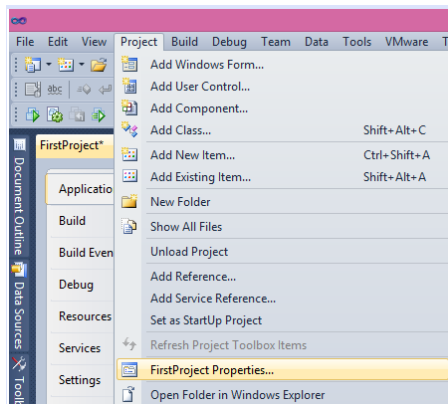
40

40

Các thành phần trong một project (tt)

- Thiết lập các thuộc tính cho Project:
 - Từ menu Project, chọn Project-name Properties

– hoặc click chuột phải trên tên project trong cửa sổ Solution Explorer, chọn Properties



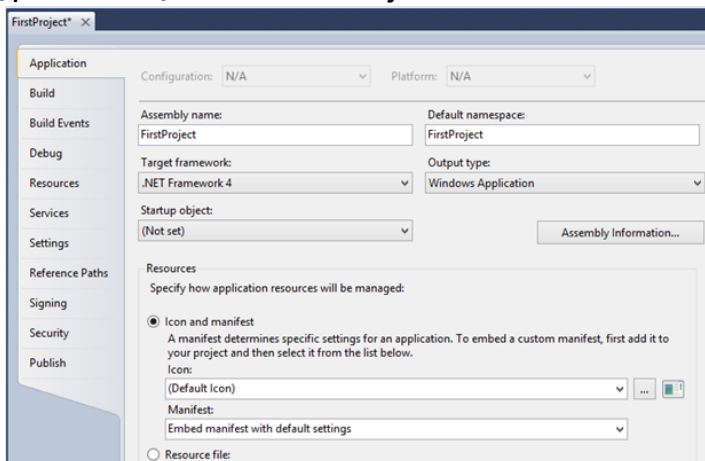
Nguyễn Thị Mai Trang

41

41

Các thành phần trong một project (tt)

- Thiết lập các thuộc tính cho Project



Nguyễn Thị Mai Trang

42

42

Các thành phần trong một project (tt)

- Thiết lập các thuộc tính thông thường
 - Định nghĩa tên Assembly
 - Namespace gốc
 - Đầu ra Project
 - Đối tượng chạy đầu tiên
 - Import các mức namespaces
- Thiết lập các thuộc tính cấu hình
 - Debug
 - Các tùy chọn Build

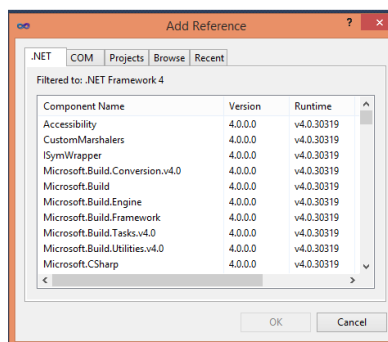
Nguyễn Thị Mai Trang

43

43

Các thành phần trong một project (tt)

- Thiết lập các tham chiếu trong Project:
 - Nhiều Project cần tham chiếu đến các ứng dụng khác hoặc code của các thư viện.
 - Sử dụng hộp thoại Add Reference



Nguyễn Thị Mai Trang

44

44

1.2.7 Trang Start Page

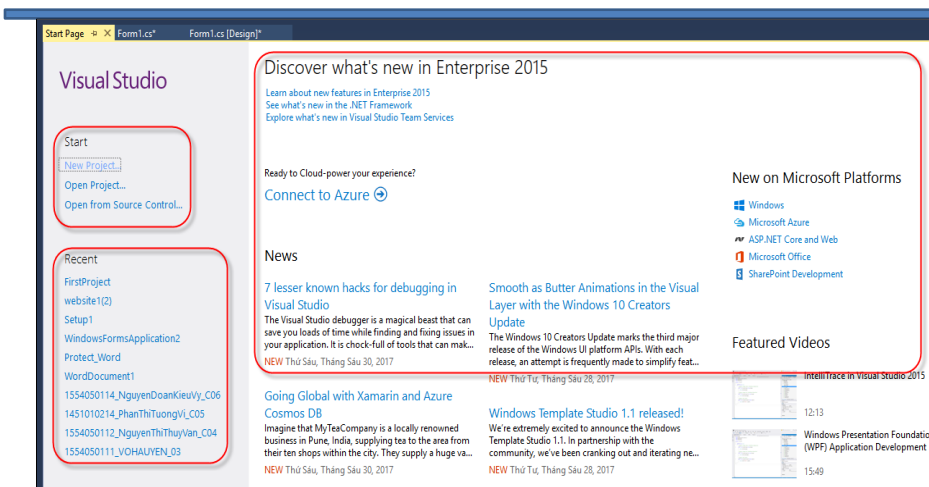
- Chứa danh sách các liên kết đến các nguồn tài nguyên Visual Studio IDE, mỗi liên kết cho phép truy cập đến thông tin có liên quan từ các web site. Bao gồm:
 - Phần Start: chứa các nút cho phép tạo hoặc mở project.
 - Phần Recent: chứa thông tin về các project mới nhất được tạo hoặc sửa đổi giúp mở nhanh các project này.
 - Các liên kết khác: chứa các thông tin giới thiệu và hướng dẫn về cách sử dụng các tài nguyên trong .NET.

Nguyễn Thị Mai Trang

45/55

45

Trang Start Page (tt)



Nguyễn Thị Mai Trang

46/55

46

1.2.8 Minh họa xây dựng ứng dụng với Visual C#



Nguyễn Thị Mai Trang

47