



Chương 1

Giới thiệu C#

1

Nội dung

1. Tình hình trước khi MS.NET ra đời
2. Microsoft .NET
3. Biên dịch và MSIL
4. Ngôn ngữ C#
5. Câu hỏi

2/46

1. Tình hình trước khi MS.NET ra đời

- Java được Sun viết ra, đã có sức mạnh đáng kể, nó hướng tới việc chạy trên nhiều hệ điều hành khác nhau, độc lập với bộ xử lý
- Đặc biệt là Java rất thích hợp cho việc viết các ứng dụng trên Internet
- Tuy nhiên, Java lại có hạn chế về mặt tốc độ và trên thực tế vẫn chưa thịnh hành

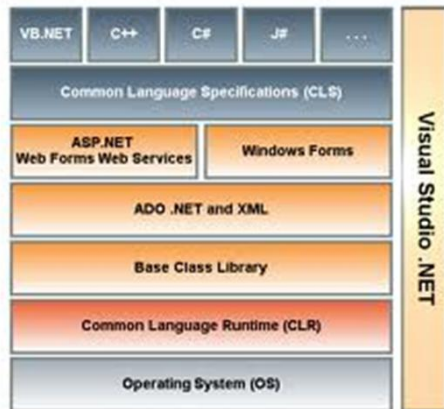
3/46

1. Tình hình trước khi MS.NET ra đời

- Để lập trình trên Web, người ta vẫn dùng PHP, một ngôn ngữ tốc độ nhanh
- Trong giới phát triển ứng dụng trên Windows ta có thể viết ứng dụng bằng Visual C++, Delphi hay Visual Basic
- Do Visual Basic không thích hợp cho việc viết các ứng dụng Web Server nên Microsoft tạo ra **ASP** (Active Server Page)
- Bản thân của ASP vẫn còn một số khuyết điểm quan trọng, nhất là khi chạy trên **IIS** (Internet Information Server) với Windows NT 4, ASP không đáng tin cậy lắm

4/46

2. Microsoft .NET



5/46

2. Microsoft .NET

- Đầu năm 1998, sau khi hoàn tất phiên bản Version 4 của Internet Information Server (IIS), các đội ngũ lập trình ở Microsoft nhận thấy họ còn rất nhiều sáng kiến để kiện toàn IIS.
- Họ bắt đầu xây dựng một kiến trúc mới trên nền tảng ý tưởng đó và đặt tên là Next Generation Windows Services (NGWS)
- Thật sự Microsoft đã đặt cược vào .NET vì theo thông tin của công ty, đã tập trung 80% sức mạnh của Microsoft để nghiên cứu và triển khai .NET

6/46

Microsoft .NET

- **.NET** là một môi trường quản lý, phát triển và thực thi các mã ngôn ngữ .NET
- **.NET** cung cấp các khả năng :
 - Về cấp phát và thu hồi bộ nhớ
 - Quản lý cấp quyền,
 - Cung cấp và quản lý các nguồn tài nguyên
- Gồm 2 phần chính:
 - **Common Language Runtime (CLR)**
 - **Framework class library**

7/46

Common Language Runtime (CLR)

- CLR thực hiện quản lý :
 - Bộ nhớ
 - Quản lý thực thi tiểu trình, thực thi mã nguồn
 - Xác nhận mã nguồn an toàn
 - Biên dịch và các dịch vụ hệ thống khác
- CLR thúc đẩy :
 - Việc mã nguồn thực hiện việc truy cập được bảo mật
 - Mã nguồn được thực thi mạnh mẽ hơn bằng việc thực thi mã nguồn chính xác và sự xác nhận mã nguồn

8/46



.NET framework class library

- Framework là một tập hợp hướng đối tượng của các kiểu dữ liệu được dùng lại
- Nó cho phép phát triển những ứng dụng từ :
 - Ứng dụng truyền thống command-line (Console)
 - Ứng dụng có giao diện đồ họa (GUI)
 - Những ứng dụng mới nhất được cung cấp bởi ASP.NET (Web Form và dịch vụ XML Web)
 - Dịch vụ windows

9/46

Một số tính năng của Microsoft .NET

- Một mô hình lập trình cho phép :
 - Phát triển xây dựng các ứng dụng dịch vụ web
 - Ứng dụng client với Extensible Markup Language (XML).
- Cung cấp các server phục vụ bao gồm: Windows 2000, SQL Server, và BizTalk Server, tất cả đều tích hợp, hoạt động, và quản lý các dịch vụ XML Web và các ứng dụng
- Các phần mềm client như Windows XP và Windows CE giúp người phát triển phân phối sâu và thuyết phục người dùng kinh nghiệm thông qua các dòng thiết bị
- Nhiều công cụ hỗ trợ như Visual Studio .NET, để phát triển các dịch vụ Web XML, ứng dụng trên nền Windows hay nền web một cách dễ dàng và hiệu quả

10/46

Phát triển ứng dụng Client

- Kiểu truyền thống
- ActiveX control

11/46

Kiểu truyền thống

- Đây là những kiểu ứng dụng hiển thị những cửa sổ hay những form trên desktop
- Cho phép người dùng thực hiện một thao tác hay nhiệm vụ nào đó như :
 - Như xử lý văn bản, xử lý bảng tính, những ứng dụng trong lĩnh vực thương mại như công cụ nhập liệu, công cụ tạo báo cáo...
 - Thường sử dụng những cửa sổ, menu, toolbar, button hay các thành phần GUI khác
 - Chúng thường truy cập các tài nguyên cục bộ như là các tập tin hệ thống, các thiết bị ngoại vi như máy in.

12/46

ActiveX control

- Được nhúng vào các trang web trên Internet
- Hiện nay nó được thay thế bởi các Windows Form control
- Windows Form control có sự truy cập giới hạn đến máy của người sử dụng

13/46

3. Biên dịch và MSIL

- Trong .NET Framework, chương trình không được biên dịch vào các tập tin thực thi mà thay vào đó chúng được biên dịch vào những tập tin trung gian gọi là Microsoft Intermediate Language (MSIL)
- Khi ứng dụng của bạn thực thi, mã MSIL là "just in time" biên dịch sang mã máy bởi JITTER (just in time compiler)
- Như vậy khi bạn viết các lớp trên .Net bằng bất kỳ ngôn ngữ nào khi bạn biên dịch sang Assembly bạn đều có thể sử dụng Assembly đó cho các ngôn ngữ khác

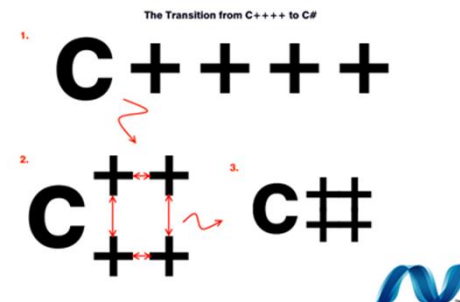
14/46

4. Ngôn ngữ C#

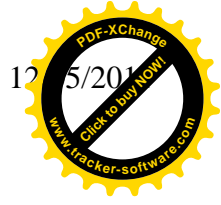
- Giới thiệu
- C# và các ngôn ngữ lập trình khác (sinh viên tự nghiên cứu)
- Tại sao là C#
- Các bước chuẩn bị cho chương trình
- Chương trình C# đơn giản
- Phát triển chương trình minh hoạ

15/46

Giới thiệu



16/46



Giới thiệu

- Ngôn ngữ C# khá đơn giản:
 - Chỉ khoảng 80 từ khóa
 - Hơn mười mấy kiểu dữ liệu được xây dựng sẵn
- Ngôn ngữ C# có ý nghĩa cao :
 - Bao gồm tất cả những hỗ trợ cho cấu trúc
 - Thành phần component
 - Lập trình hướng đối tượng
 - Được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java
- Ngôn ngữ C# được phát triển bởi đội ngũ kỹ sư của Microsoft, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth

17/46

Cách thức làm việc

- C# làm việc xoay quanh khái niệm Assembly
- Assembly là tập hợp mã đã được biên dịch sang .NET
- Tiến trình thực thi bởi một chương trình C# :
 - Khi chương trình được thực thi, CLR sẽ xác nhận đến các Assembly manifest và quyền hạn chạy của chương trình trên hệ thống
 - Hệ bảo vệ hệ thống:
 - Không cho phép chương trình chạy, chương trình sẽ không chạy
 - Nếu được phép, CLR sẽ thực thi mã lệnh. Bit đầu tiên của code được nạp vào bộ nhớ và được biên dịch thành mã nhị phân từ IL bởi JIT. Sau khi được biên dịch, mã được thực thi và chứa trong bộ nhớ

18/46

Tại sao là C#?

- C# đơn giản
- C# hiện đại
- C# hướng đối tượng
- C# mạnh mẽ và mềm dẻo
- C# ít từ khóa
- C# hướng module
- C# sẽ phổ biến

19/46

C# đơn giản

- C# loại bỏ một vài sự phức tạp và rối rắm của những ngôn ngữ như Java và c++ [macro, template, đa kế thừa, và lớp cơ sở ảo (virtual base class)]
- Ngôn ngữ C# đơn giản vì nó dựa trên nền tảng C và C++

20/46

C# hiện đại

- Con trỏ được tích hợp vào ngôn ngữ C++. Chúng cũng là nguyên nhân gây ra những rắc rối của ngôn ngữ này
- Ngôn ngữ hiện đại phải có những đặc tính :
 - Xử lý ngoại lệ
 - Thu gom bộ nhớ tự động
 - Những kiểu dữ liệu mở rộng
 - Bảo mật mã nguồn

21/46

C# hướng đối tượng

- Đóng gói (encapsulation)
- Sự kế thừa (inheritance)
- Đa hình (polymorphism)

22/46

C# mạnh mẽ và mềm dẻo

- Với ngôn ngữ C# chúng ta chỉ bị giới hạn ở :
 - Chính bởi bản thân người dùng
 - Trí tưởng tượng của chúng ta
- C# được sử dụng cho nhiều các dự án khác nhau như :
 - Tạo ra ứng dụng xử lý văn bản
 - Ứng dụng đồ họa
 - Bảng tính
 - Trình biên dịch cho các ngôn ngữ khác

23/46

C# ít từ khóa

abstract	default	foreach	object	sizeof	unsafe
as	delegate	goto	operator	stackalloc	ushort
base	do	if	out	static	using
bool	double	implicit	override	string	virtual
break	else	in	params	struct	volatile
byte	enum	int	private	switch	void
case	event	interface	protected	this	while
catch	explicit	internal	public	throw	
char	extern	is	readonly	true	
checked	false	lock	ref	try	
class	finally	long	return	typeof	
const	fixed	namespace	sbyte	uint	
continue	float	new	sealed	ulong	
decimal	for	null	short	unchecked	

24/46



C# hướng module

- Mã nguồn C# có thể được viết trong những phần được gọi là những lớp
- Những lớp này chứa các phương thức thành viên của nó
- Những lớp và những phương thức có thể được sử dụng lại trong ứng dụng hay các chương trình khác

25/46

C# sẽ phổ biến

- Microsoft và sự cam kết của .NET
- Nhiều sản phẩm của công ty này đã chuyển đổi và viết lại bằng C#. Bằng cách sử dụng ngôn ngữ này Microsoft đã xác nhận khả năng của C# cần thiết cho những người lập trình

26/46

Chuẩn bị cho chương trình

- Xác định mục tiêu của chương trình.
- Xác định những phương pháp giải quyết vấn đề.
- Tạo một chương trình để giải quyết vấn đề.
- Thực thi chương trình để xem kết quả.

27/46

Phát triển chương trình minh họa

- Cách 1:
 - Soạn chương trình: Sử dụng chương trình soạn thảo văn bản bất kỳ như Notepad
 - Biên dịch chương trình: Sử dụng câu lệnh
- Cách 2: Sử dụng môi trường phát triển tích hợp (IDE) Visual Studio .NET

28/46

Cách 1

- Sử dụng Notepad soạn thảo chương trình
- Lưu tập tin xuống đĩa và tập tin này có phần mở rộng là *.cs
- Biên dịch tập tin nguồn vừa tạo ra :
`csc.exe [/out: <file thực thi>] <file nguồn>`
- Ví dụ:
`csc.exe /out: d:\chaomung.exe d:\chaomung.cs`

29/46

Cách 2

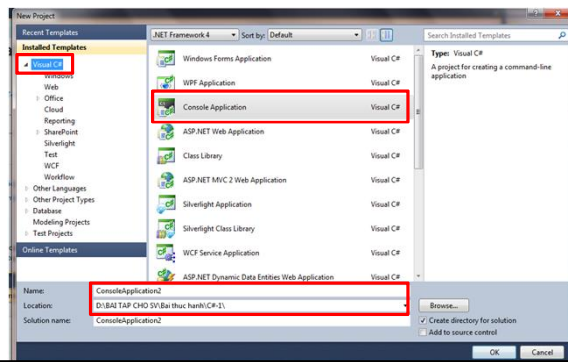
Sử dụng Visual Studio .NET :

- Soạn thảo chương trình
- **Biên dịch**
- Thực thi

30/46

Soạn thảo chương trình

- Chọn mục Visual Studio .NET trong menu Start
- File ⇒ New ⇒ Project



31/46

Soạn thảo chương trình

- Lớp, đối tượng và kiểu dữ liệu (type)
- Phương thức
- Chú thích
- Ứng dụng Console
- Namespace
- Toán tử '.'
- Từ khóa using
- Từ khóa static
- Phân biệt chữ thường và chữ hoa

32/46



Lớp, đối tượng và kiểu dữ liệu (type)

- Lớp : tập các đối tượng cùng hành vi
- Đối tượng : thể hiện riêng của lớp
- Kiểu :
 - Là một thứ được xem như trừu tượng. Nó có thể là một bảng dữ liệu, hay một nút lệnh trong một cửa sổ
 - Như một dạng vừa có thuộc tính chung (properties) và các hành vi ứng xử (behavior) của nó
- Các kiểu khác nhau : **kiểu lớp**; kiểu liệt kê; cấu trúc; kiểu ủy quyền (delegates)

33/46

Lớp

```
class Program
```

```
{
```

Các thuộc tính
Các hành động

```
}
```

34/46

Phương thức

```
class Program
{
    static void Main(string[] args)
    {
        System.Console.WriteLine("Chao mung ban den voi C#");
        System.Console.ReadLine();
    }
}
```

Main() :

- Hàm đặc biệt; Nó được gọi đầu tiên; Không mô tả hành động nào của lớp
- Trả về **void** hoặc **int**

35/46

Chú thích

- Chú thích: Giải thích; nhắc nhở
- Có 2 cách ghi chú thích:
 - // chỉ có tác dụng trên dòng này
 - /* có tác dụng cho nhiều dòng */

36/46



Ứng dụng Console

- Ứng dụng này giao tiếp với người dùng thông qua bàn phím
- Không có giao diện người dùng (GUI)

Các ứng dụng nâng cao trên Windows hay Web thì ta mới dùng các các giao diện đồ họa

37/46

Namespace

- .NET cung cấp một thư viện các lớp đồ sộ và thư viện này có tên là FCL (Framework Class Library)
- Mỗi lớp có một tên riêng, vì vậy FCL có hàng ngàn tên
- Từ đó dẫn đến:
 - Người lập trình không thể nào nhớ hết được tên của các lớp trong .NET Framework
 - Tệ hơn nữa là sau này có thể ta tạo lại một lớp trùng với lớp đã có chẳng hạn

38/46

Giải pháp

- Tạo ra một namespace, namespace sẽ hạn chế phạm vi của một tên, làm cho tên này chỉ có ý nghĩa trong vùng đã định nghĩa
- Ngoài ra còn có subnamespace
- Ví dụ:
 - Có một người nói Tùng là một kỹ sư (kỹ sư cầu đường, cơ khí hay phần mềm)
 - C# : Tùng là CauDuong.KySu phân biệt với CoKhi.KySu hay PhanMem.KySu.
 - Namespace trong trường hợp này là CauDuong, CoKhi, PhanMem

39/46

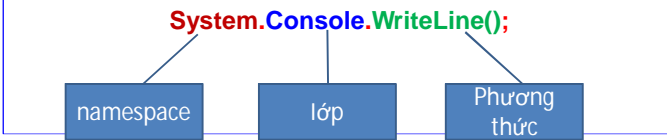
Tạo namespace

```
namespace MyLib
{
    using System;
    public class Tester
    {
        public static int Main()
        {
            for (int i =0; i < 10; i++)
            {
                Console.WriteLine( "i: {0}", i);
            }
            return 0;
        }
    }
}
```

40/46

Toán tử '.'

- Dấu '.' được sử dụng để truy cập đến **phương thức** hay **dữ liệu** trong một **lớp**, và ngăn cách giữa tên lớp đến một **namespace**
- Việc thực hiện này theo hướng từ trên xuống, trong đó mức đầu tiên **namespace**, tiếp theo là **lớp**, và cuối cùng là truy cập đến các **phương thức** hay **thuộc tính** của lớp



41/46

Từ khóa using

using < Tên namespace >

- Để làm cho chương trình gọn hơn, và không cần phải viết từng namespace cho từng đối tượng
- Sau từ khóa này là một namespace hay subnamespace với mô tả đầy đủ trong cấu trúc phân cấp của nó

42/46

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;
```

```
namespace ConsoleApplication1  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            System.Console.WriteLine("Chao mung ban den voi C#");  
            Console.WriteLine("Chuc ban thoai mai voi C#");  
            Console.ReadLine();  
        }  
    }  
}
```

using System.Console

43/46

Phân biệt chữ thường và chữ hoa

- C# là ngôn ngữ phân biệt chữ thường với chữ hoa
- Điều này có nghĩa rằng hai câu lệnh **writeLine** thì khác với **WriteLine** và cũng khác với **WRITELINE**

44/46



Biên dịch – Thực thi

1. Để biên dịch chương trình :
 1. Nhấn **Ctrl-Shift-B**
 2. **Build** ⇒ **Build Solution**
2. Để chạy chương trình :
 1. **Ctrl-F5**
 2. **Debug** ⇒ **Start Without Debugging**
 3. Nút lệnh ► trên thanh toolbar

45/46

Câu hỏi

1. Một chương trình C# có thể chạy trên bất cứ máy nào?
2. Nếu muốn đưa chương trình mà ta viết cho một người bạn thì tập tin nào mà chúng ta cần đưa?
3. Sau khi tạo ra được tập tin thực thi .exe. Có cần thiết giữ lại tập tin nguồn không?
4. Nếu trình biên dịch C# đưa ra một trình soạn thảo, có phải nhất thiết phải sử dụng nó?
5. Có thể không quan tâm đến những cảnh báo khi biên dịch mã nguồn?
6. CLR viết tắt cho từ nào và ý nghĩa của nó?
7. Ngôn ngữ máy là gì? Khi biên dịch mã nguồn C# ra tập tin .exe thì tập tin này là ngôn ngữ gì?

46/46