

Nhập môn DotNet.

Bài 1

Yêu cầu

- Biết được sự ra đời, quá trình phát triển, tương lai cũng như cấu trúc của bộ sản phẩm .Net
- Nắm được các khái niệm cơ bản trong C# như: cấu trúc chương trình, từ khóa, các toán tử...
- Biết cách sử dụng một số kiểu dữ liệu cơ bản: mảng, chuỗi ký tự, liệt kê, struct.

Giới thiệu Microsoft.Net

- Cùng với sự phát triển liên tục của CNTT nhất là phần mềm, hệ điều hành, các môi trường phát triển phần mềm → các ứng dụng liên tục được ra đời, tuy nhiên phát triển chưa được thống nhất.
- Java ra đời đã có sức mạnh đáng kể, đặc biệt với các ứng dụng trên Internet.

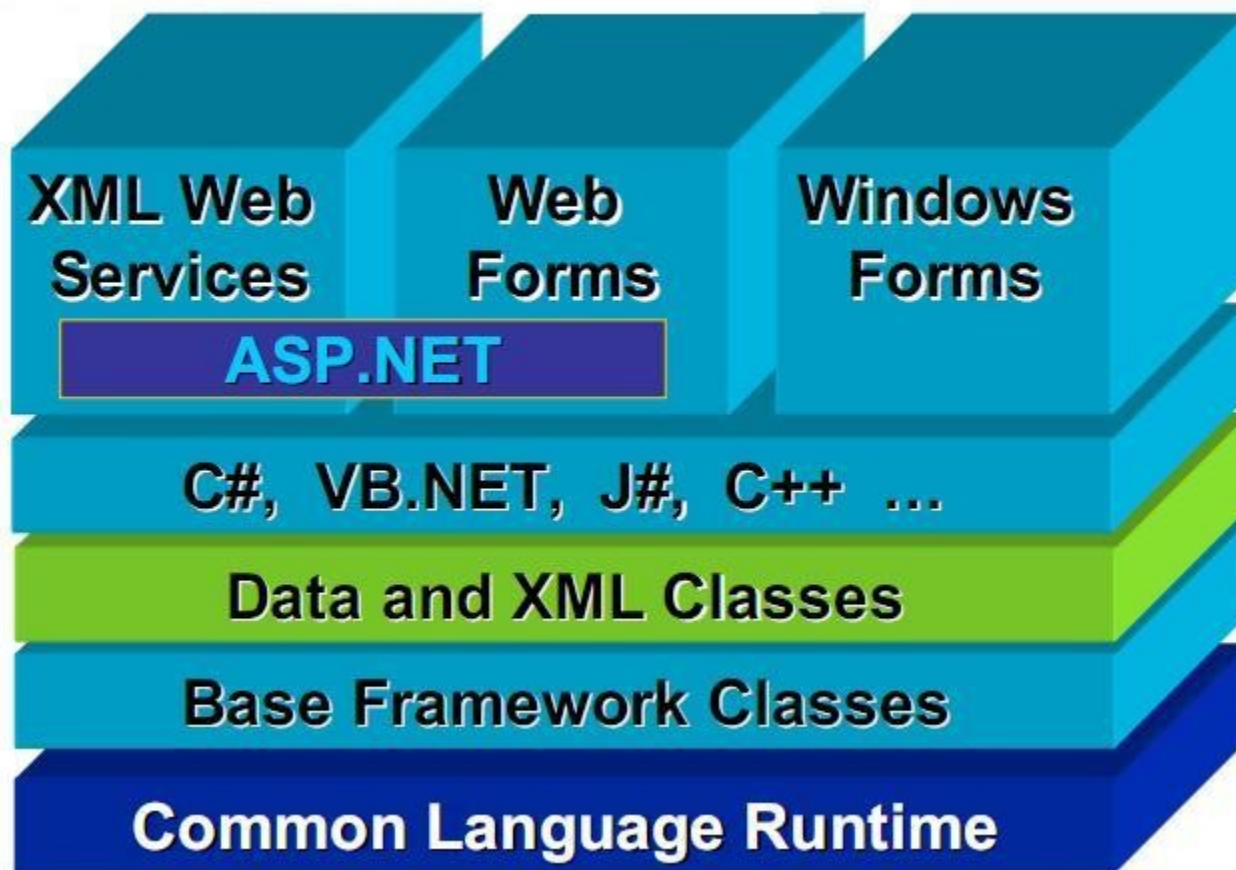
Giới thiệu Microsoft.Net

- Microsoft đã dùng ASP để làm giảm ảnh hưởng của JAVA. Ngoài ra khi làm việc trên môi trường web còn một số ngôn ngữ như CGI-Perl, PHP
- Lập trình ứng dụng cũng có nhiều công cụ: Visual C++, Delphi, Visual Basic...
- Tháng 7/2000 Microsoft phát hành phiên bản beta của .Net

Cấu trúc Microsoft.Net

- Microsoft.Net bao gồm hai phần chính: Framework và Integrated Development Environment (IDE).
- Thành phần Framework là tinh hoa, nền tảng của .Net, còn IDE chỉ là môi trường để triển khai các ứng dụng. Trong .Net toàn bộ các ngôn ngữ C#, Visual C++, Visual Basic.Net đều dùng cùng một IDE.

Microsoft.Net Framework



Microsoft.Net Framework

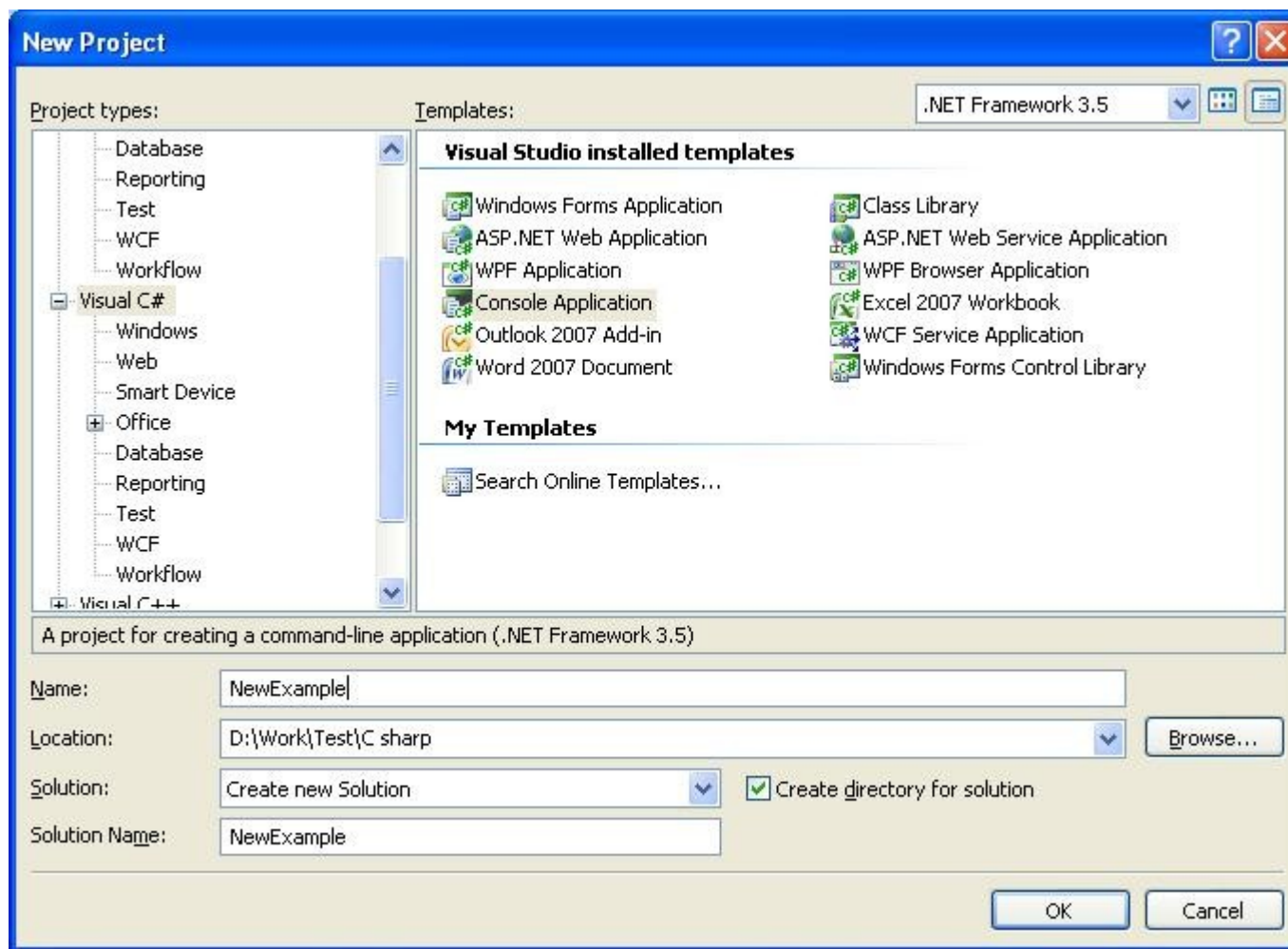
- Thành phần quan trọng nhất của .Net Framework là CLR-Common Language Runtime, cung cấp môi trường cho ứng dụng thực thi.
- .Net hỗ trợ tích hợp ngôn ngữ, có thể kế thừa các lớp, bắt các biệt lệ, đa hình thông qua nhiều ngôn ngữ. .Net framework sử dụng đặc tả Common Type System-CTS (hệ thống kiểu chung) và Common Language Specification-CLS (đặc tả ngôn ngữ chung).

Ngôn ngữ trung gian MSIL

- Với .Net chương trình không biên dịch thành tập tin thực thi mà biên dịch thành ngôn ngữ trung gian MSIL-Microsoft Intermediate Language. Sau đó chúng được CLR thực thi. Các tập tin MSIL biên dịch từ C# đồng nhất với các tập tin MSIL biên dịch từ ngôn ngữ .Net khác.
- Khi chạy chương trình IL được biên dịch (hay thông dịch) một lần nữa bằng trình Just In Time-JIT → mã máy.

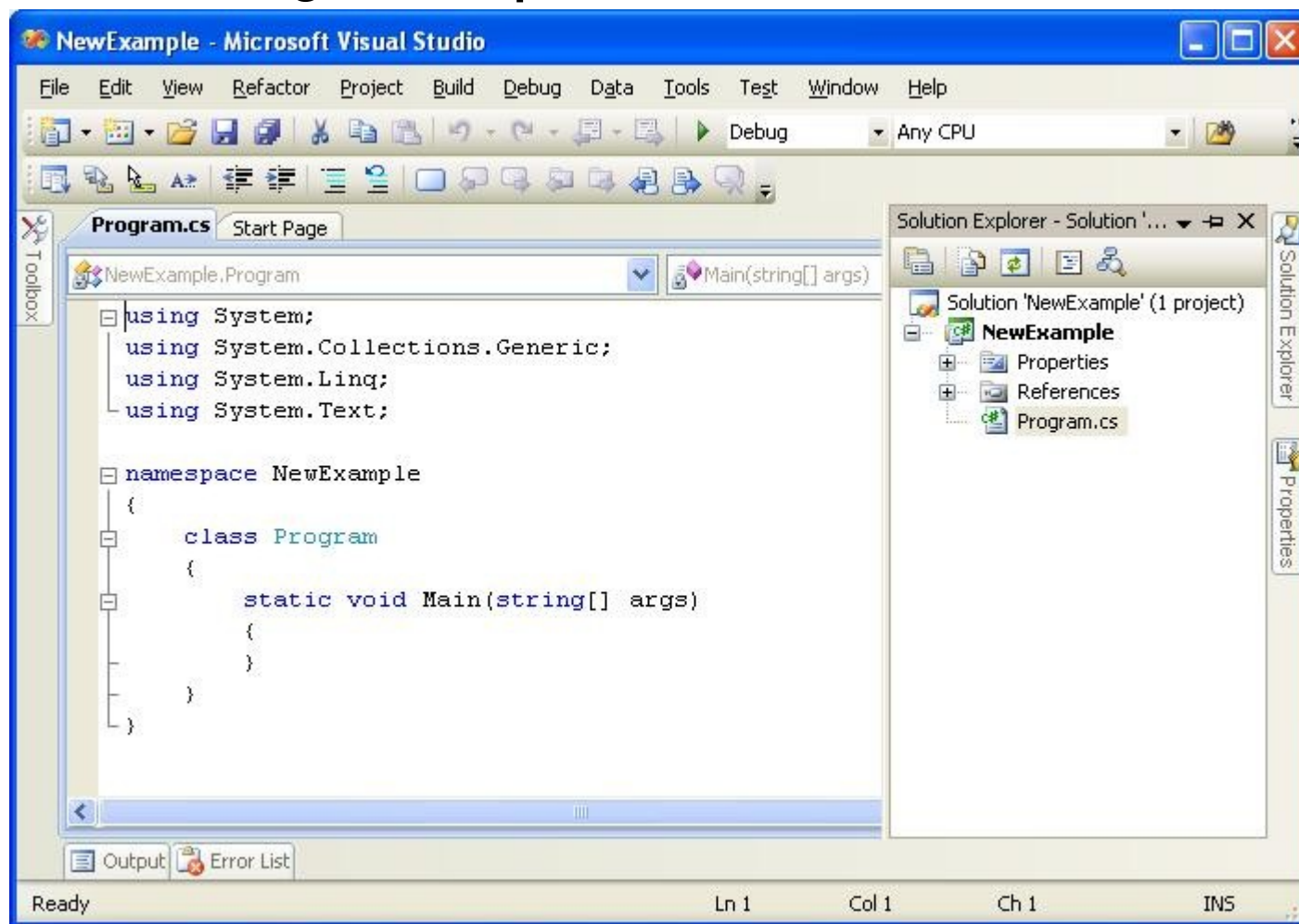
Visual Studio.Net

Tạo một Project mới



Visual Studio.Net

Môi trường làm việc



Ngôn ngữ C#

C# là một ngôn ngữ rất đơn giản, được kế thừa có chọn lọc, phát triển từ C++ và Java, với khoảng 80 từ khóa và hơn 10 kiểu dữ liệu dựng sẵn. C# hỗ trợ lập trình có cấu trúc, lập trình hướng đối tượng và hướng thành phần (component oriented).

Cấu trúc chương trình C#

```
using System;
namespace example
{
class Program
    {
        static void Main(string[] args)
        {
            //In ra man hinh
            Console.WriteLine("Lap trinh C#");
            Console.ReadLine();
        }
    }
}
```

Cấu trúc chương trình C#

- Lớp, đối tượng: Bản chất của lập trình HĐT là tạo ra các kiểu mới. Giống như các ngôn ngữ HĐT khác một kiểu trong C# cũng định nghĩa bằng từ khóa class (gọi là lớp). Thể hiện của một lớp gọi là đối tượng.
- Hành vi của lớp được gọi là các phương thức thành viên
- Ghi chú: trong C# sử dụng hai kiểu ghi chú quen thuộc là “//” và “/* ...*/”

namespace (không gian tên)

Console là một lớp trong cả ngàn lớp của bộ thư viện .Net.

Một vùng tên có thể có nhiều lớp và vùng tên con khác. Nếu vùng tên A nằm trong vùng tên B, ta nói vùng tên A là vùng tên con của vùng tên B, khi đó các lớp trong vùng tên A được ghi như sau: B.A.ten_lop_trong_vung_ten_A

System là vùng tên chứa nhiều lớp hữu ích cho việc giao tiếp với hệ thống hoặc các lớp công dụng chung như Console, Math, Exception...

Toán tử chấm “.”

Toán tử chấm dùng để truy xuất dữ liệu và phương thức của một lớp, đồng thời cũng để chỉ định tên lớp trong một vùng tên

Ví dụ: `System.Console`

Toán tử chấm cũng dùng để truy xuất các vùng tên con của một vùng tên

Ví dụ: `VungTen.VungTenCon.VungTenConCon`

Biên dịch, thực thi chương trình

Có hai cách biên dịch, thực thi chương trình đó là dùng môi trường phát triển tích hợp IDE (Visual Studio) hoặc viết bằng trình soạn thảo văn bản và dịch bằng dòng lệnh

Sau khi đầy đủ mã nguồn ta tiến hành biên dịch: nhấn “Ctrl+Shift+B” hoặc Build>Build Solution. Nhấn F5 để bắt đầu chạy chương trình.

Kiểu dữ liệu định sẵn

Kiểu	Kích thước(byte)	Kiểu .Net	Giá trị
byte	1	Byte	0..255
char	1	Char	Mã Unicode
bool	1	Boolean	True hoặc False
sbyte	1	Sbyte	-128..127
short	2	Int16	-32768...32767
ushort	2	UInt16	0..65535
int	4	Int32	-2147483647..
float	4	Single	Số thực

Chuyển đổi kiểu định sẵn

Một đối tượng có thể chuyển từ kiểu này sang kiểu kia một cách ngầm định hoặc tường minh

```
short x=3;
```

```
int y;
```

```
y=x;
```

```
x=(short) y;
```

Biến và hằng

Khi dùng một biến trước hết nó phải được khởi tạo

```
int x;
```

```
x=5;
```

```
int y=x;
```

Hằng là một biến nhưng giá trị không thay đổi theo thời gian.

```
const int hs=10;
```

Vào ra dữ liệu

```
int n;
```

```
    Console.Write("Nhập giá trị n:");
```

```
    n = Convert.ToInt32(Console.ReadLine());
```

```
    n = int.Parse(Console.ReadLine());
```

Các toán tử

Nhóm toán tử	Toán tử	Ý nghĩa
Toán học	+ - * / %	cộng , trừ, nhân chia, lấy phần dư
Logic	& ^ ! ~ && true false	phép toán logic và thao tác trên bit
Ghép chuỗi	+	ghép nối 2 chuỗi
Tăng, giảm	++, --	tăng / giảm toán hạng lên / xuống 1. Đứng trước hoặc sau toán hạng.
Dịch bit	<< >>	dịch trái, dịch phải
Quan hệ	== != < > <= >=	bằng, khác, nhỏ/lớn hơn, nhỏ/lớn hơn hoặc bằng
Gán	= += -= *= /= %= &= = ^= <<= >>=	phép gán
Chỉ số	[]	cách truy xuất phần tử của mảng
Ép kiểu	()	
Indirection và Address	* -> [] &	dùng cho con trỏ

Các toán tử

$n++$, $++n$ là các toán tử tăng giá trị giá trị của n được tăng thêm 1. Sự khác nhau chỉ thể hiện rõ trong các biểu thức toán học.

Toán tử tam phân:

$\langle \text{Biểu thức đk} \rangle ? \langle \text{biểu thức1} \rangle : \langle \text{biểu thức2} \rangle ;$

Các lệnh rẽ nhánh không điều kiện

Có hai loại câu lệnh rẽ nhánh không điều kiện: lệnh gọi phương thức và sử dụng từ khóa goto, break, continue, return.

Các lệnh rẽ nhánh có điều kiện

Các từ khóa if-else, while, do-while, for, switch-case dùng để điều khiển dòng chảy của chương trình

if(biểu thức logic)

khởi lệnh ;

if(biểu thức logic)

khởi lệnh 1;

else khởi lệnh 2;

Các lệnh rẽ nhánh có điều kiện

switch (biểu thức lựa chọn)

{

case biểu_thức_hằng:

 khởi_lệnh;

 lệnh_nhảy;

[*default*:

 khởi_lệnh;

 lệnh_nhảy;]

}

Lệnh lặp

- **Lệnh goto** có thể dùng để tạo vòng lặp có điều kiện, cụ thể: tạo nhãn và goto đến nhãn đó

- **Lệnh while**

Cú pháp:

while (*biểu_thức_logic*)

khởi_lệnh;

Lệnh lặp

- **Lệnh do-while**

do khối lệnh;

while (biểu_thức_logic)

- **Vòng lặp for**

for ([khởi_tạo_biến_đếm];[biểu_thức];
[tăng_biến_đếm])

khối lệnh;

Các kiểu dữ liệu cơ bản

Trong C# cũng hỗ trợ các kiểu dữ liệu như: mảng, chuỗi, liệt kê, cấu trúc giống như trong C++ và Java

- Kiểu mảng:
 - kiểu[] tên_mảng;
 - kiểu[,] tên_mảng;
 - Kiểu[][] tên_mảng

Các kiểu dữ liệu cơ bản

- Kiểu chuỗi
 - Cú pháp: `string` tên_chuỗi;
 - Tạo chuỗi bằng phương thức `ToString()` của đối tượng.
- Kiểu liệt kê: là tập hợp các tên hằng có giá trị không thay đổi
 - Cú pháp
`enum` <ten_liet_ke>[kieu_co_so]
{danh_sach_cac_thanh_phan_liet_ke};

Các kiểu dữ liệu cơ bản

- Cấu trúc: là kiểu dữ liệu đơn giản, kích thước nhỏ, dùng thay thế lớp, cũng có thể chứa các phương thức, thuộc tính, các trường, các toán tử, các kiểu dữ liệu lồng bên trong.
 - **struct** <tên_cấu_trúc> [:danh_sách_giao_diên]
{
 [thành_viên_cấu_trúc];
}

Các kiểu dữ liệu cơ bản

```
using System;
public struct Location
{
    public Location( int xCoordinate, int yCoordinate)
    {
        xVal = xCoordinate;
        yVal = yCoordinate;
    }
    public int x
    {
        get{return xVal;}
        set{xVal = value;}
    }
    public int y
    {
        get{return yVal;}
        set{yVal = value;}
    }
}
```

Các kiểu dữ liệu cơ bản

```
public override string ToString()
{return (String.Format("{0}, {1}", xVal,
    yVal));}
// thuộc tính private lưu tọa độ x, y
private int xVal;
private int yVal;
}
```