

PHÁT TRIỂN ỨNG DỤNG CSDL 1

Tháng 9/2014

LẬP TRÌNH ỨNG DỤNG VỚI NGÔN NGỮ C#

**Môi trường, cấu trúc chương trình, kiểu dữ liệu,
các cấu trúc điều khiển**

Tóm tắt nội dung bài thực hành:

Hướng dẫn về môi trường lập trình, cấu trúc chương trình, sử dụng các kiểu dữ liệu cơ bản, các cấu trúc điều khiển của ngôn ngữ lập trình c#.

Bộ môn **Hệ thống thông tin**

Khoa Công nghệ thông tin

ĐH Khoa học tự nhiên TP HCM



MỤC LỤC

1	Mục tiêu.....	1
2	Công cụ lập trình	1
3	Khởi tạo project.....	1
4	Cấu trúc chương trình:	2
5	Các kiểu dữ liệu cơ bản.....	3
5.1	Kiểu dữ liệu cơ sở	3
5.2	Khai báo biến:	3
5.3	Chuyển đổi kiểu dữ liệu sang chuỗi	4
5.4	Chuyển đổi kiểu dữ liệu chuỗi sang kiểu int , long.....	4
5.5	Khai báo hằng số:	4
6	Thao thác với màn hình console	4
6.1	Nhập liệu với màn hình console.....	4
6.2	Xuất ra màn hình console	4
7	Cú pháp rẽ nhánh với If.....	5
8	Cú pháp rẽ nhánh với switch – case	6
9	Cú pháp lặp với While:.....	6
10	Cấu trúc lặp với do ... While	7
11	Cấu trúc lặp với for.....	8
12	Break.....	8
13	Continue	9
14	Các Toán Tử.....	9
14.1	Toán tử số học	9
14.2	Toán tử so sánh.....	10



14.3	Toán tử luận lý	11
15	Các hàm toán học phổ biến.....	11
16	Mảng 1 chiều:.....	11
16.1	Cú pháp khai báo:.....	11
16.2	Cấp vùng nhớ:.....	12
16.3	Khởi tạo:.....	13
16.4	Nhập mảng.....	13
16.5	Xuất mảng.....	13
17	Mảng hai chiều	14
17.1	Khai báo:	14
17.2	Cấp phát vùng nhớ.....	14
17.3	Nhập mảng:.....	15
17.4	Xuất mảng.....	15

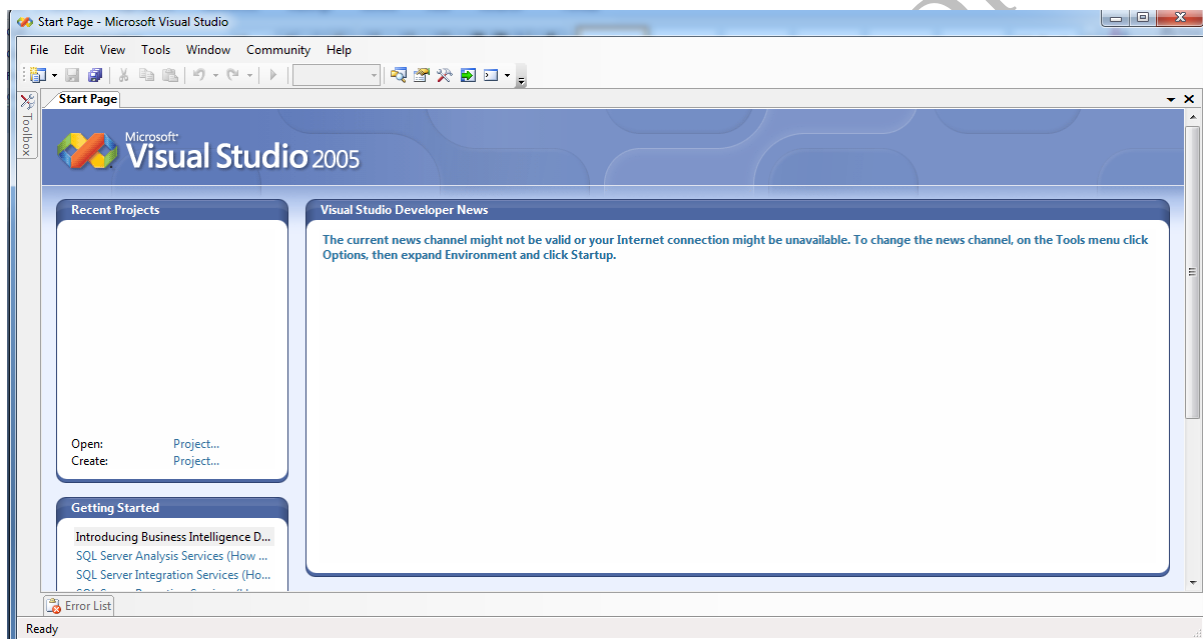
1 Mục tiêu

Sau khi hoàn thành bài tập này sinh viên có thể:

- Tạo được 1 project với môi trường lập trình ứng dụng với C#
- Sử dụng được các cấu trúc dữ liệu cơ bản trong C#
- Sử dụng được các cấu trúc điều khiển với C#

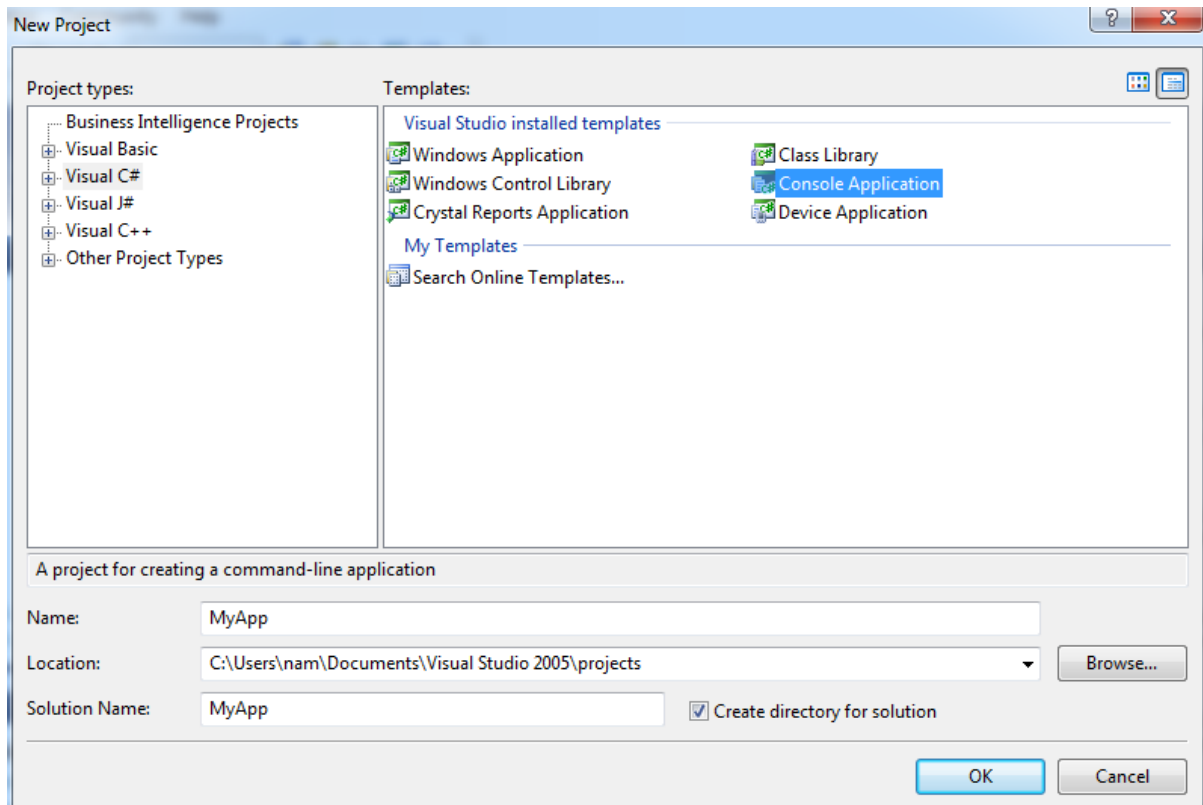
2 Công cụ lập trình

Visual studio



3 Khởi tạo project

File → New Project



Chọn ngôn ngữ lập trình(C#, Visual Basic C++...), loại ứng dụng(Window Application: ứng dụng có giao diện đồ họa, Console Application: ứng dụng với giao diện console...), thiết lập Project name, location, folder → OK

4 Cấu trúc chương trình:

Chương trình C# gồm một hoặc nhiều lớp(class) trong đó phải có 1 lớp chứa hàm main. Hàm main là một phương thức tĩnh của 1 lớp trong chương trình và là điểm bắt đầu của chương trình:

```
static void Main(string[] args)
{
    String s = "Hello World";
    Console.WriteLine(s);
}
```

5 Các kiểu dữ liệu cơ bản

5.1 Kiểu dữ liệu cơ sở

Kiểu	Số byte	Mô tả
byte	1	Số nguyên dương không dấu từ 0-255
char	2	Ký tự Unicode
bool	1	Giá trị logic true/ false
sbyte	1	Số nguyên có dấu (từ -128 đến 127)
short	2	Số nguyên có dấu giá trị từ -32768 đến 32767.
ushort	2	Số nguyên không dấu 0 – 65.535
Int	4	Số nguyên có dấu -2.147.483.647 và 2.147.483.647
uint	4	Số nguyên không dấu 0 – 4.294.967.295
float	4	Kiểu dấu chấm động, giá trị xấp xỉ từ 3,4E-38 đến 3,4E+38, với 7 chữ số có nghĩa..
double	8	Kiểu dấu chấm động có độ chính xác gấp đôi, giá trị xấp xỉ từ 1,7E-308 đến 1,7E+308, với 15,16 chữ số có nghĩa
decimal	8	Có độ chính xác đến 28 con số và giá trị thập phân, được dùng trong tính toán tài chính, kiểu này đòi hỏi phải có hậu tố “m” hay “M” theo sau giá trị
Long	8	Kiểu số nguyên có dấu có giá trị trong khoảng : - 9.223.370.036.854.775.808 đến 9.223.372.036.854.775.807
ulong	8	Số nguyên không dấu từ 0 đến 0xffffffffffffffff

5.2 Khai báo biến:

```
KieuDuLieu TenBien;  
KieuDuLieu TenBien = GiaTri;
```

5.3 Chuyển đổi kiểu dữ liệu sang chuỗi

```
String str= TenBien.ToString();
```

Ví dụ:

```
int a = 1;  
String str = a.ToString();
```

5.4 Chuyển đổi kiểu dữ liệu chuỗi sang kiểu int , long....

```
int int.Parse(string str)  
long long.Parse(string str)  
float float.Parse(string str)  
double double.Parse(string str)  
char[] array =str.ToCharArray();
```

5.5 Khai báo hằng số:

Hằng số là 1 biến mà không được thay đổi giá trị của nó .

```
const KieuDuLieu TENHANGSO = GiaTri;
```

6 Thao tác với màn hình console

6.1 Nhập liệu với màn hình console

```
string str = Console.ReadLine();
```

6.2 Xuất ra màn hình console

Sử dụng `Console.Write (...)` hay `Console.WriteLine (...)`

```
Console.Write("Hello World");  
  
String s = "Hello World";  
  
Console.WriteLine (s);
```


Định dạng chuỗi xuất với cấu trúc:

```
String.Format("chuỗi định dạng", tham số 1, ..., tham số n);
```

Với

```
String s = "Hello World";  
int n = 100;  
double m = 20.8;  
string str = string.Format("n ={0}, m= {1} , s={2}", n,  
m,s);  
Console.WriteLine(str);
```

7 Cú pháp rẽ nhánh với If

Cú pháp:

```
if (biểu thức điều kiện)  
{  
    ...  
}  
else  
{  
    ...  
}
```

Ví dụ:

```
Console.Write("d=");  
String str = Console.ReadLine();  
double d = double.Parse(str);  
if (d > 0)  
{  
    Console.WriteLine("{0}> 0",d);  
}  
else  
{  
    Console.WriteLine("{0} <= 0",d);  
}
```

8 Cú pháp rẽ nhánh với switch – case

Cú pháp:

```
switch (giá trị){  
    case giá trị 1:  
        ...  
        break;  
        ...  
    case giá trị N:  
        ...  
        break;  
    default:  
        ...  
        break;  
}
```

Ví dụ:

```
char grade = 'B';  
switch (grade)  
{  
    case 'A':  
        Console.WriteLine("Excellent!");  
        break;  
    case 'B':  
    case 'C':  
        Console.WriteLine("Well done");  
        break;  
    case 'D':  
        Console.WriteLine("You passed");  
        break;  
    case 'F':  
        Console.WriteLine("Better try  
again");  
        break;  
    default:  
        Console.WriteLine("Invalid grade");  
        break;  
}
```

9 Cú pháp lặp với While:

Cú pháp:

```
while (điều kiện lặp) {  
    ...  
}
```

Ví dụ:

```
Console.Write("n=");  
String str = Console.ReadLine();  
int n = int.Parse(str);  
int s = 0;  
int i = 1;  
while (i <= n)  
{  
    s = s + i;  
    i++;  
}  
Console.WriteLine("S=" + s.ToString());
```

10 Cấu trúc lặp với do ... While

Cú pháp:

```
do{  
    ...  
}while (điều kiện lặp);
```

Ví dụ:

```
int i = 0;  
Console.Write("Cac so nho hon 5: ");  
do  
{  
    Console.Write("{0} ", i);  
    i++;  
} while ( i < 5 ) ;
```

11 Cấu trúc lặp với for

Cú pháp:

```
for (khởi tạo biến; điều kiện lặp; hành động sau mỗi lần lặp)
{
    ...
}
```

Ví dụ:

```
Console.Write("Cac so chan nho hon 5: ");
for (int i = 0; i <= 5 ; i+=2)
{
    Console.WriteLine(i.ToString() + " ");
}

// Kết quả: Cac so chan nho hon 5 : 0 2 4
```

12 Break

Được sử dụng trong for, while, do-while với mục đích thoát khỏi vòng lặp

Ví dụ:

```
for (int i = 0; i < 5; i++)
{
    if (i == 3)
        break;
    Console.Write("{0} ", i);
}

// Kết quả: 0 1 2
```

13 Continue

Được sử dụng trong for, while, do-while với mục đích bỏ qua bước lặp hiện tại, quay trở lại vòng lặp để thực hiện tiếp.

Ví dụ:

```
for (int i = 0; i < 5; i++)
{
    if (i == 3)
        continue;
    Console.Write("{0} ", i);
}

// Kết quả: 0 1 2 4
```

14 Các Toán Tử

14.1 Toán tử số học

Là dạng phép tính giản đơn cộng, trừ, nhân, chia trong số học. Ngoài ra còn có phép chia lấy dư (%).

Toán tử số học cơ bản

Toán tử	Ý nghĩa	Ví dụ
+	Toán tử cộng	a = a + 2;
-	Toán tử trừ	a = a - 2;
*	Toán tử nhân	a *= 2;
/	Toán tử chia	a /= 2;
%	Toán tử lấy phần dư sau khi chia.	

Toán tử số học viết tắt

Toán tử	Ý nghĩa	Ví dụ
=	Toán tử gán	<code>a = a + 2;</code>
+=	Toán tử cộng thêm giá trị toán hạng bên phải	<code>a += 2;</code>
-=	Toán tử trừ đi giá trị toán hạng bên phải	<code>a -= 2;</code>
*=	Toán tử nhân thêm giá trị toán hạng bên phải	<code>a *= 2;</code>
/=	Toán tử chia đi giá trị toán hạng bên phải	<code>a /= 2;</code>
%=	Toán tử lấy phần dư sau khi chia hết toán hạng bên phải.	

14.2 Toán tử so sánh

Toán tử	Ý nghĩa	Ví dụ		
==	So sánh bằng.	<code>(2==2)</code>	→	True
		<code>(2==3)</code>	→	False
!=	Không bằng	<code>(2!=3)</code>	→	True
		<code>(2!=2)</code>	→	False
>	Lớn hơn	<code>(3>2)</code>	→	True
		<code>(2>4)</code>	→	False
>=	Lớn hơn hay bằng	<code>(2>=2)</code>	→	True
		<code>(2>=3)</code>	→	False
<	Nhỏ hơn	<code>(2<3)</code>	→	True
		<code>(2<1)</code>	→	False
<=	Nhỏ hơn hay bằng	<code>(2<=2)</code>	→	True
		<code>(2<=0)</code>	→	False

14.3 Toán tử luận lý

Toán tử	Ý nghĩa	Ví dụ
&&	Toán tử và	(2==2 && 3<13) → True
		(2==1 && 3>=3) → False
	Toán tử hoặc	(2==2 3!=3) → True
		(2!=2 2>3) → False
!	Phủ định	(2>3) → False → !(2>1) → True
		(2>1) → True → !(2>1) → False

15 Các hàm toán học phổ biến

```
double Math.Abs (double d)
double Math.Sqrt (double d)
double Math.Min (double a, double b)
double Math.Max (double a, double b)
double Math.Exp (double e)
double Math.Round (double r)
double Math.Pow (double a, double b)
double Math.PI
double Math.E
```

16 Mảng 1 chiều:

16.1 Cú pháp khai báo:

```
KieuDuLieu[ ] tenBien;
```

Ví dụ:

```
int[] arr1; //Khai báo mảng kiểu int
long[] arr2; //Khai báo mảng kiểu long
float[] arr3; //Khai báo mảng kiểu float
double[] arr4; //Khai báo mảng kiểu double
bool[] arr5; //Khai báo mảng kiểu boolean
String[] arr6; //Khai báo mảng kiểu string
```

16.2 Cấp vùng nhớ:

Cách 1:

```
KieuDuLieu[] tenBien = new KieuDuLieu [n] ;
```

Cách 2:

```
KieuDuLieu[] tenBien;

tenBien = new KieuDuLieu [n];
```

Ví dụ:

```
int[] arr1 = new int[5]; //Khai báo và cấp phát
mảng kiểu int , a.Length = 5
long[] arr2 = new long[5]; // Khai báo và cấp phát
mảng kiểu long , a.Length = 5
float[] arr3 = new float[7]; //Khai báo và cấp
phát mảng kiểu float, a.Length = 7
String[] arr6 = new String[6]; //Khai báo và cấp
phát mảng kiểu string, a.length = 6
```


16.3 Khởi tạo:

Cách 1:

```
int[] arr = new int[5];  
arr[0] = 1;  
arr[1] = 3;  
arr[2] = 5;  
arr[3] = 7;  
arr[4] = 9;
```

Cách 2 :

```
int[] arr = { 1, 3, 5, 7, 9 }; //a.Length=5
```

16.4 Nhập mảng

```
Console.WriteLine("Số phần tử của mảng là ");  
int n = int.Parse(Console.ReadLine());  
int[] a = new int[n]; //a.Length = n  
for (int i = 0; i < a.Length; i++)  
{  
    Console.WriteLine("a[" + i.ToString() + "]=");  
    a[i] = int.Parse(Console.ReadLine());  
}
```

16.5 Xuất mảng

```
Console.WriteLine("Số phần tử của mảng {0}",  
a.Length);  
//Xuất mảng dùng for  
for (int i = 0; i < a.Length; i++) {  
    Console.WriteLine(a[i]);  
}
```

17 Mảng hai chiều

17.1 Khai báo:

```
KieuDuLieu[,] tenBien;
```

Ví dụ:

```
//Khai báo mảng hai chiều kiểu int  
int[,] arr1;  
//Khai báo mảng hai chiều kiểu long  
double[,] arr4;  
//Khai báo mảng hai chiều kiểu boolean  
bool[,] arr5;
```

17.2 Cấp phát vùng nhớ

Cách 1:

```
KieuDuLieu[,] tenBien = new KieuDuLieu [n,m] ;
```

Cách 2:

```
KieuDuLieu[,] tenBien;  
  
tenBien = new KieuDuLieu [n,m];
```

n: số dòng

m: số cột

Ví dụ:

```
/Khai báo và cấp phát mảng hai chiều kiểu int
int[,] arr1 = new int[3, 5];
//soDong = 3 ; soCot = 5
//Khai báo và cấp phát mảng kiểu long
long[,] arr2 = new long[5, 6];
//soDong = 5 ; soCot = 6
//Khai báo và cấp phát mảng hai chiều kiểu float
float[,] arr3 = new float[7, 9];
//soDong = 7 ;soCot = 9
```

17.3 Nhập mảng:

```
int dong, cot;
Console.Write("So dong:");
dong = int.Parse(Console.ReadLine());
Console.Write("So cot:");
cot = int.Parse(Console.ReadLine());
int[,] arr = new int[dong, cot];
//Nhap gia tri cho mang
for (int i = 0; i < dong; i++)
{
    for (int j = 0; j < cot; j++)
    {
        Console.Write("arr[{0}][{1}]:", i, j );
        arr[i, j] = int.Parse(Console.ReadLine());
    }
}
```

17.4 Xuất mảng

```
//Xuat mang da nhap
for (int i = 0; i < dong; i++)
{
    for (int j = 0; j < cot; j++)
    {
        Console.WriteLine("arr[{0}][{1}]={2}", i,
j,arr[i,j]);
    }
}
```