

C#.NET FOR BASE

Giảng viên: Bùi Quang Đăng

Contents

1

Introduction .NET Framework

2

Data type

3

Operators

4

Structured programming

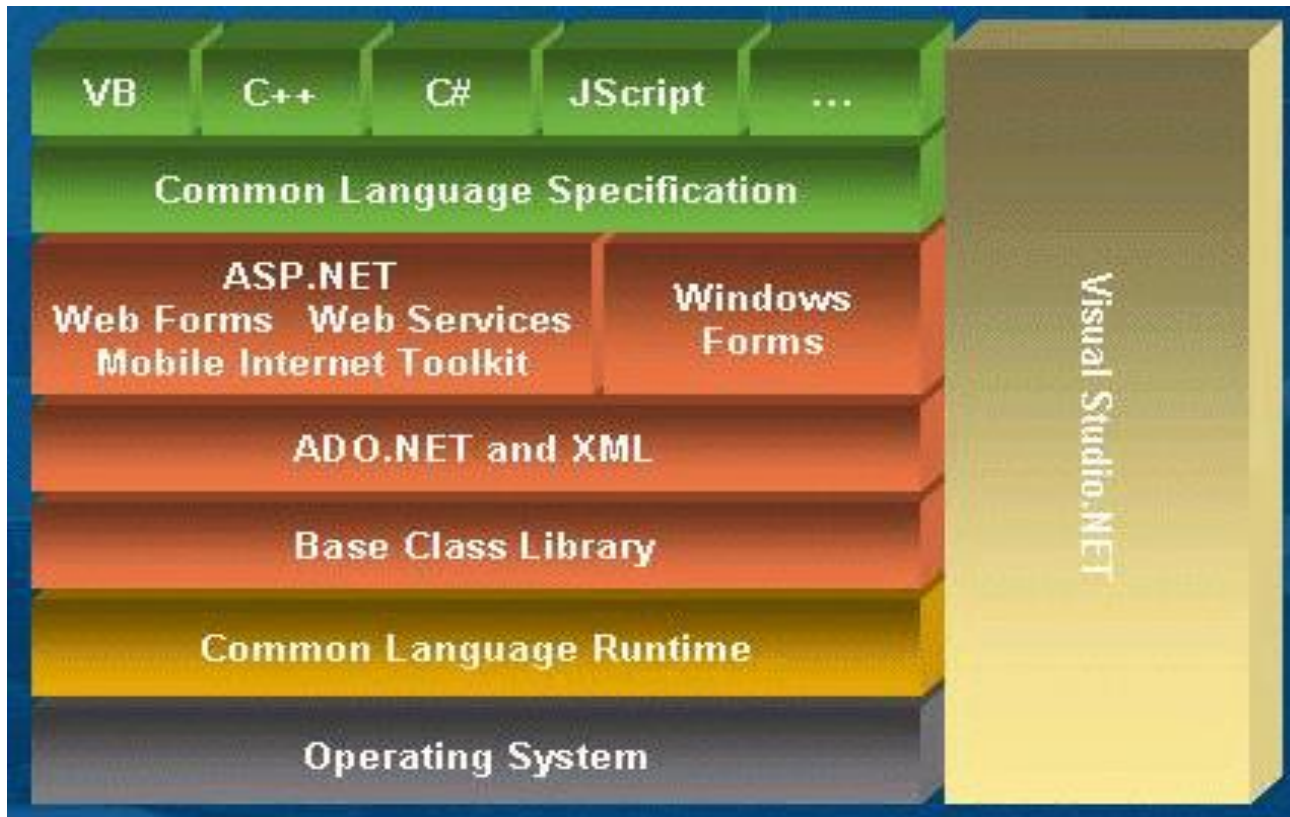


C#.NET for Base

Introduction .NET Framework

C#.NET for Base

❖ Introduction .NET Framework

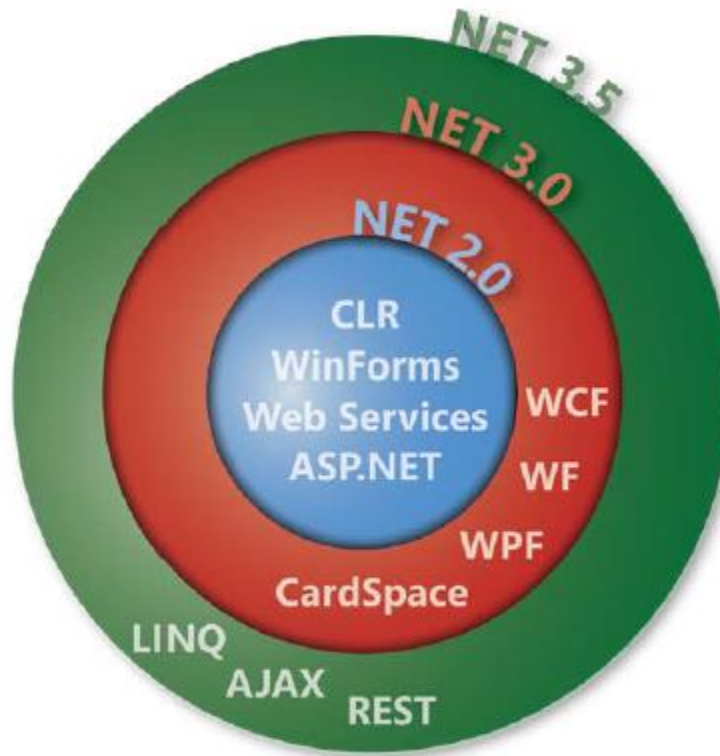


.Net Framework

là nền tảng để
chạy ứng dụng
viết bằng ngôn
ngữ **.NET**

C#.NET for Base

❖ Introduction .NET Framework



Additive versions of the .NET Framework

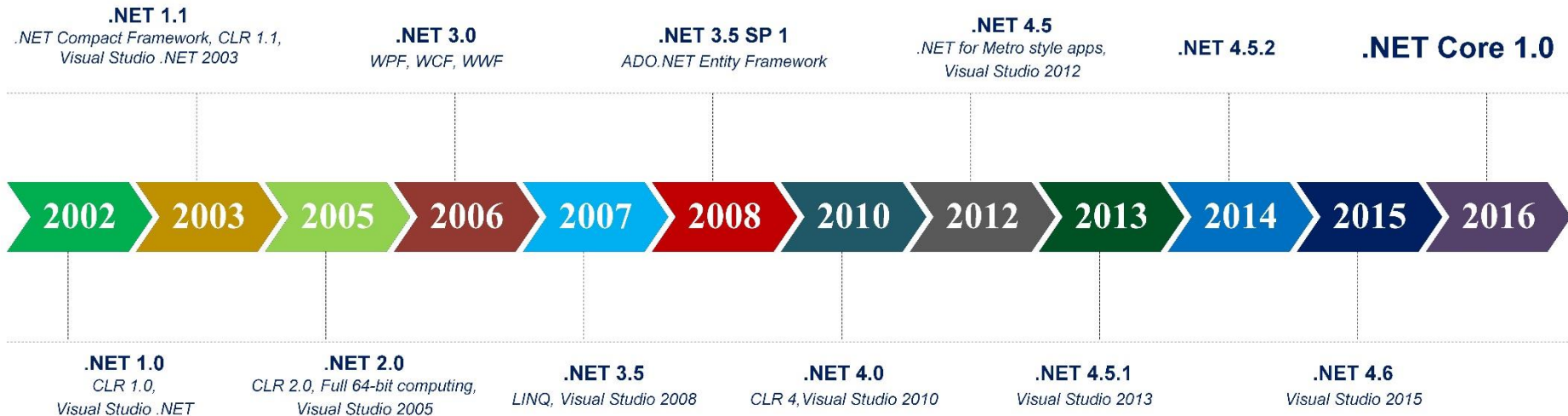
.Net Framework

có nhiều version
khác nhau

C#.NET for Base

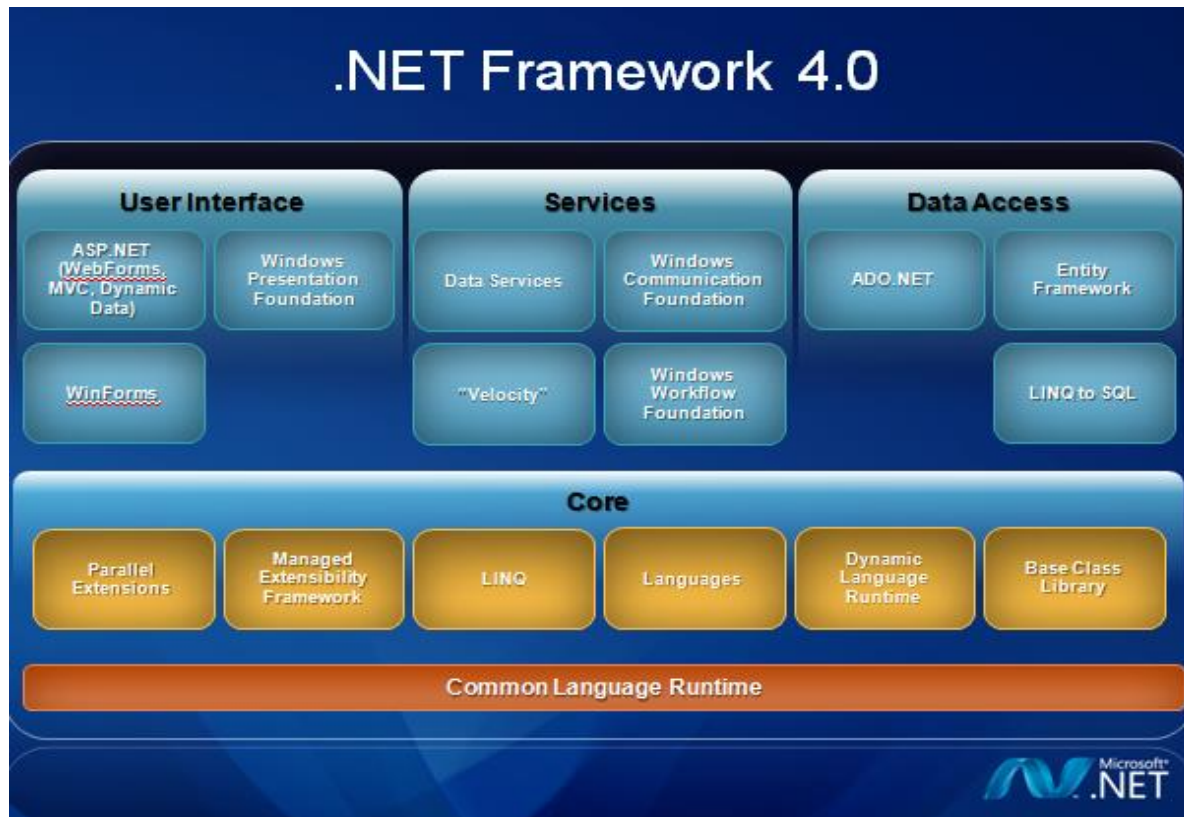
❖ Introduction .NET Framework

The release history of .NET Framework



C#.NET for Base

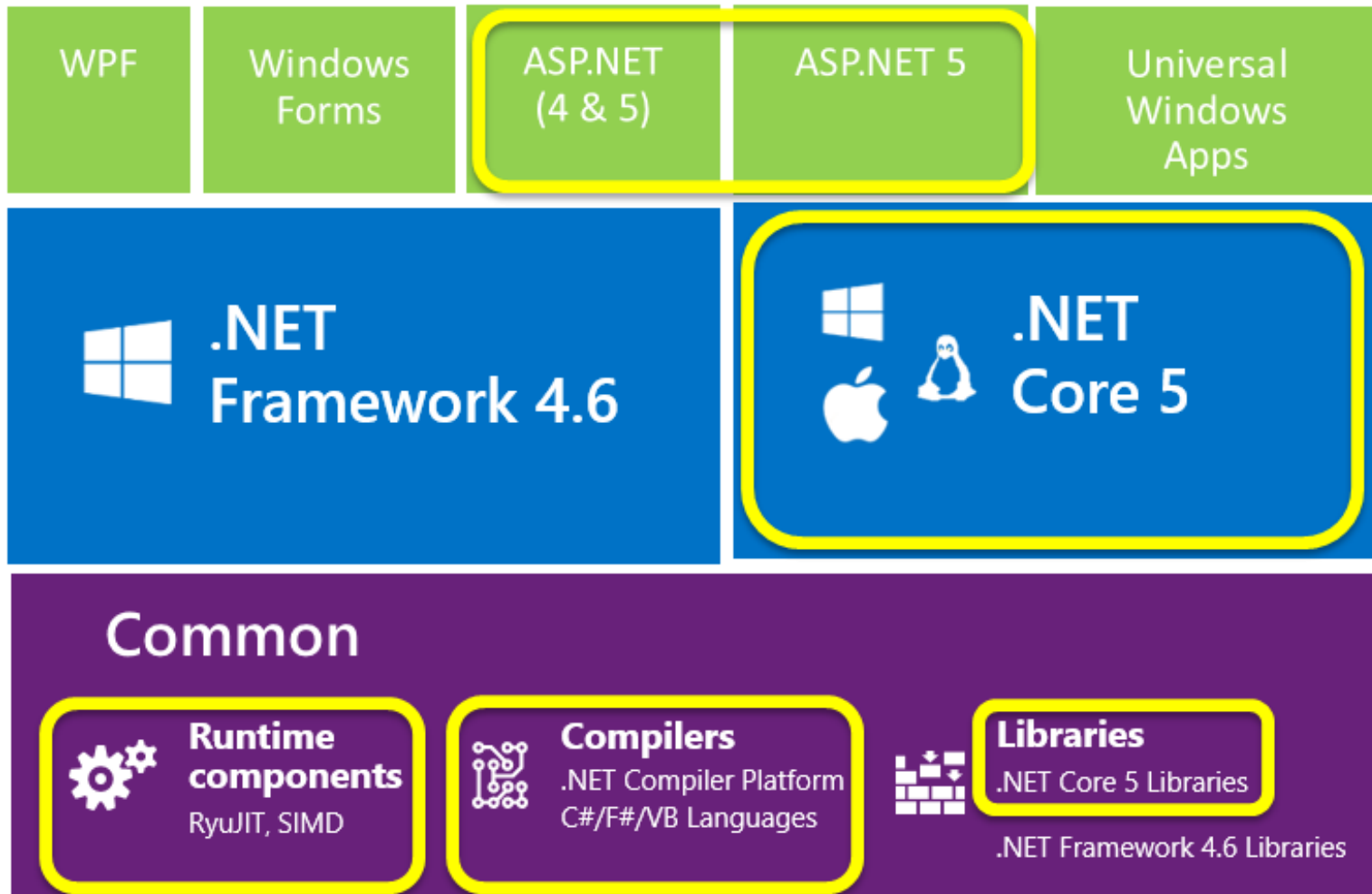
❖ Introduction .NET Framework



Microsoft
Programming the
Microsoft
ADO.NET
Entity Framework

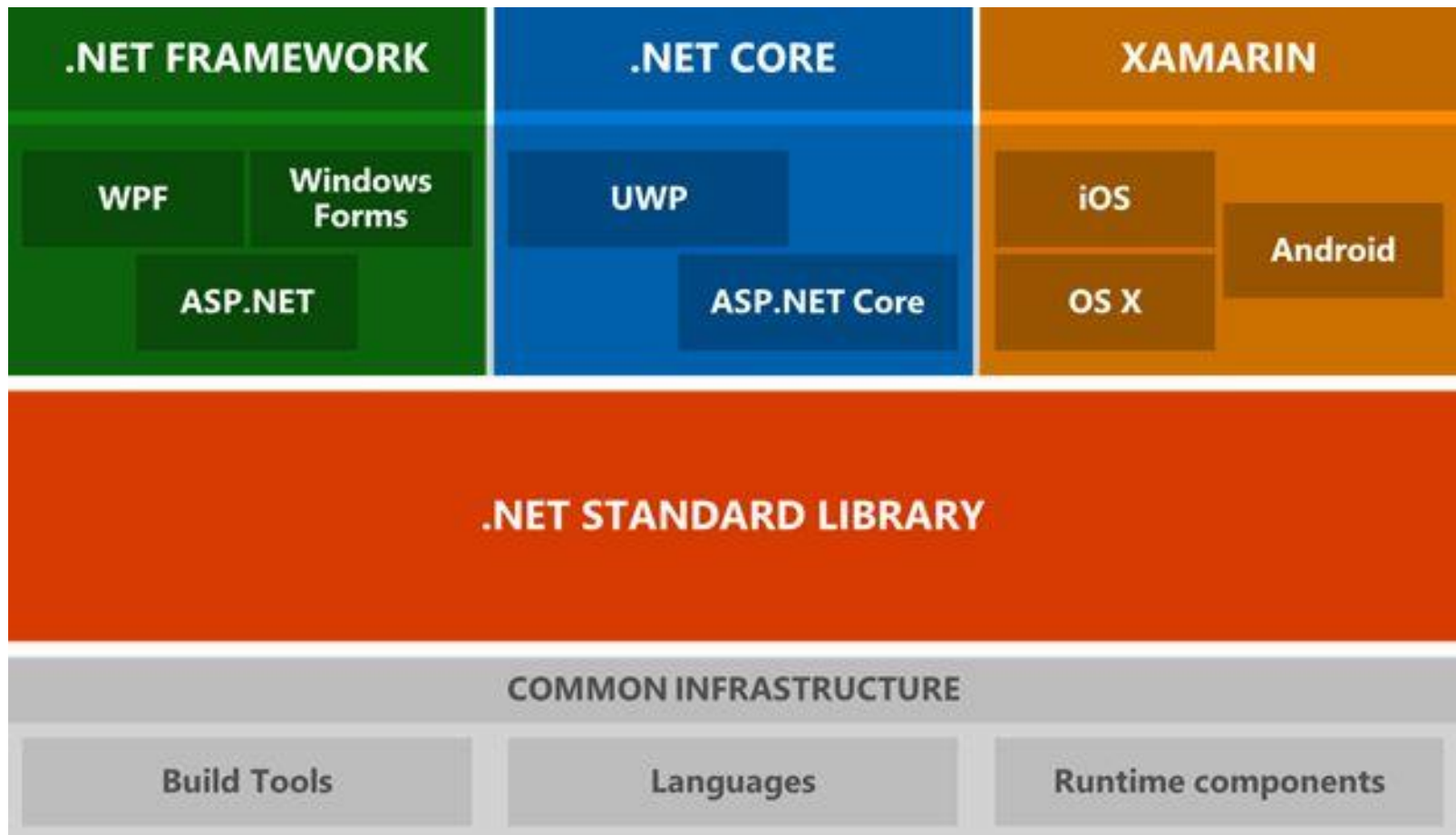
C#.NET for Base

❖ Introduction .NET Framework



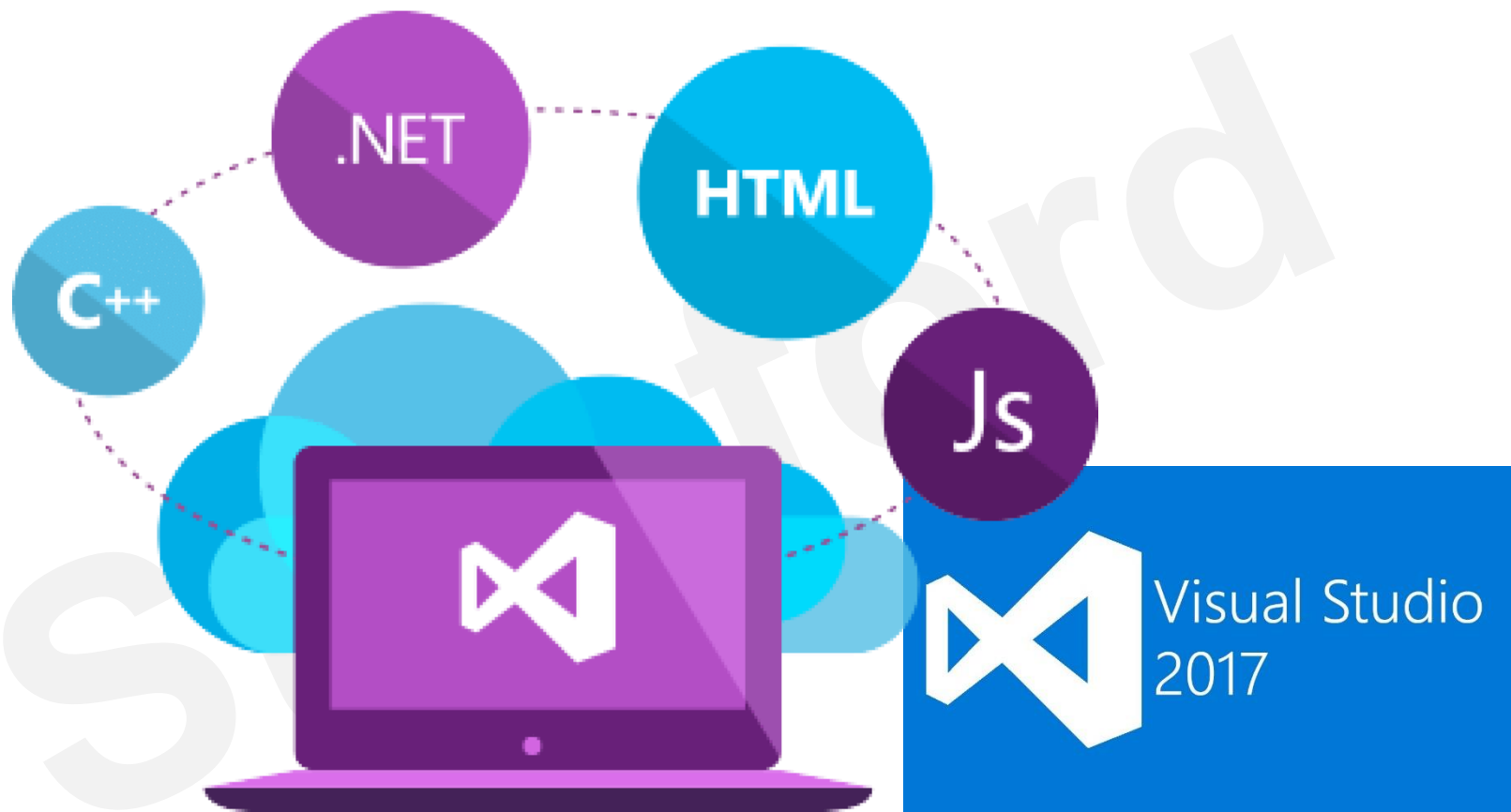
C#.NET for Base

❖ Introduction .NET Framework



C#.NET for Base

❖ Introduction .NET Framework





C#.NET for Base

Data types

C#.NET for Base

❖ Data types

Kiểu	Kích thước (byte)	Kiểu .Net	Mô tả - giá trị
byte	1	Byte	Không dấu (0..255)
char	1	Char	Mã ký tự Unicode
bool	1	Boolean	true hay false
sbyte	1	Sbyte	Có dấu (-128 .. 127)
short	2	Int16	Có dấu (-32768 .. 32767)
ushort	2	UInt16	Không dấu (0 .. 65535)
int	4	Int32	Có dấu (-2147483647 .. 2147483647)
uint	4	UInt32	Không dấu (0 .. 4294967295)
float	4	Single	Số thực ($\approx \pm 1.5 \cdot 10^{-45} \dots \approx \pm 3.4 \cdot 10^{38}$)
double	8	Double	Số thực ($\approx \pm 5.0 \cdot 10^{-324} \dots \approx \pm 1.7 \cdot 10^{308}$)
decimal	8	Decimal	số có dấu chấm tĩnh với 28 ký số và dấu chấm
long	8	Int64	Số nguyên có dấu (- 9223372036854775808 .. 9223372036854775807)
ulong	8	UInt64	Số nguyên không dấu (0 .. 0xffffffffffffffff)

Khác nhau về kích thước cấp phát vùng nhớ để lưu trữ giá trị

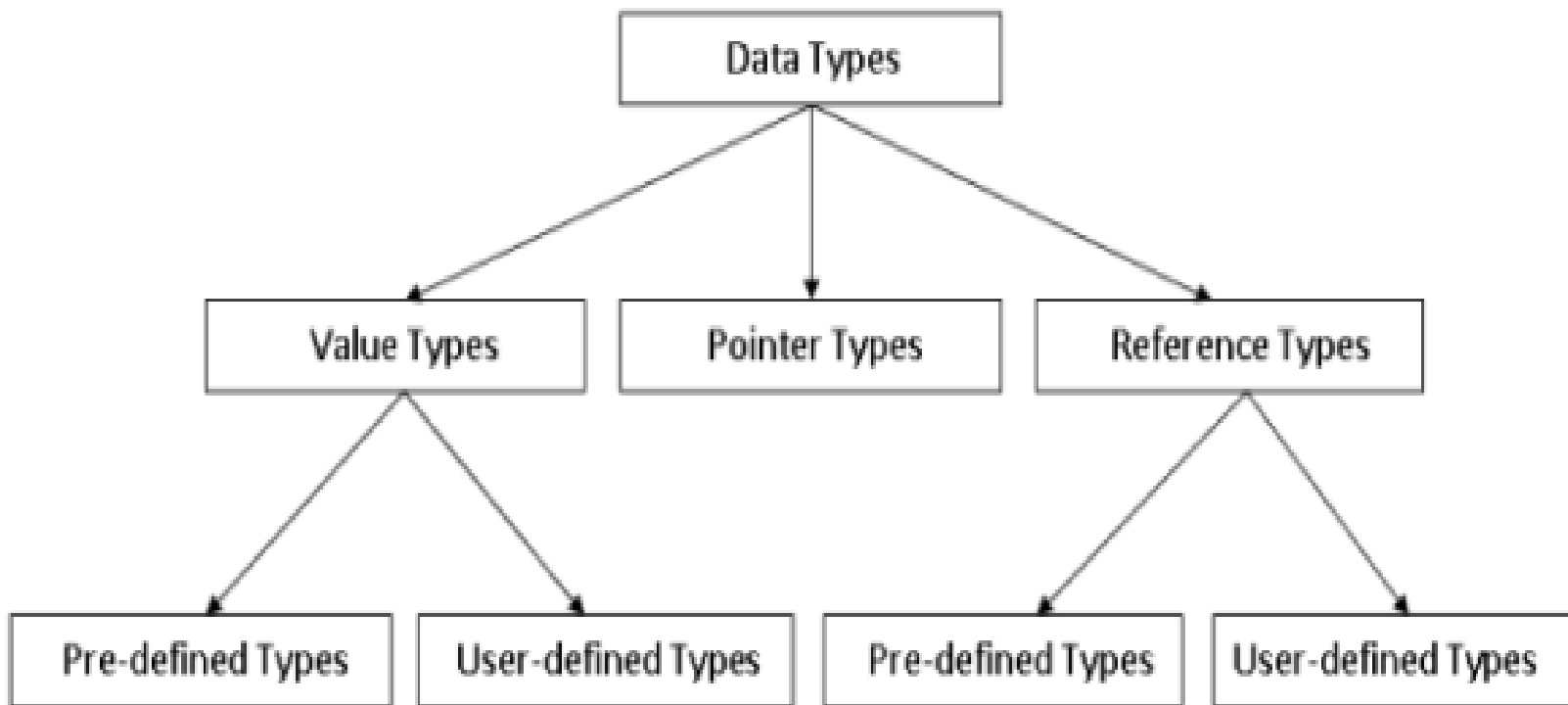
C#.NET for Base

❖ Data types

- Không thể chuyển đổi kiểu dữ liệu có giá trị lớn → nhỏ
- Kiểu boolean có giá trị true, false tương ứng với giá trị là 1 và 0
- Không thể chuyển đổi kiểu Char → Boolean. Và ngược lại
- Chỉ có thể chuyển đổi qua lại giữa kiểu Char với tất cả các kiểu khác. và ngược lại (tham khảo bảng ASCII)
- Kiểu float khi khởi tạo phải đi với ký tự F. Ví dụ: a=1.234F
- Kiểu decimal khi khởi tạo phải đi với ký tự M. Ví dụ: a=1.234M
- Chuyển đổi về kiểu Boolean giá trị khác 0 là True còn = 0 là False

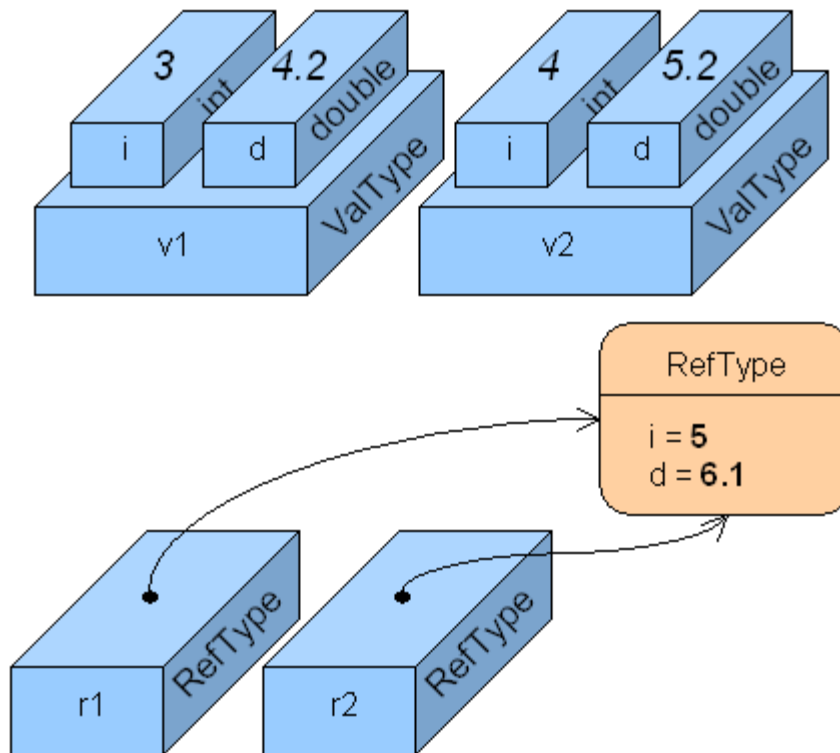
C#.NET for Base

❖ Data types



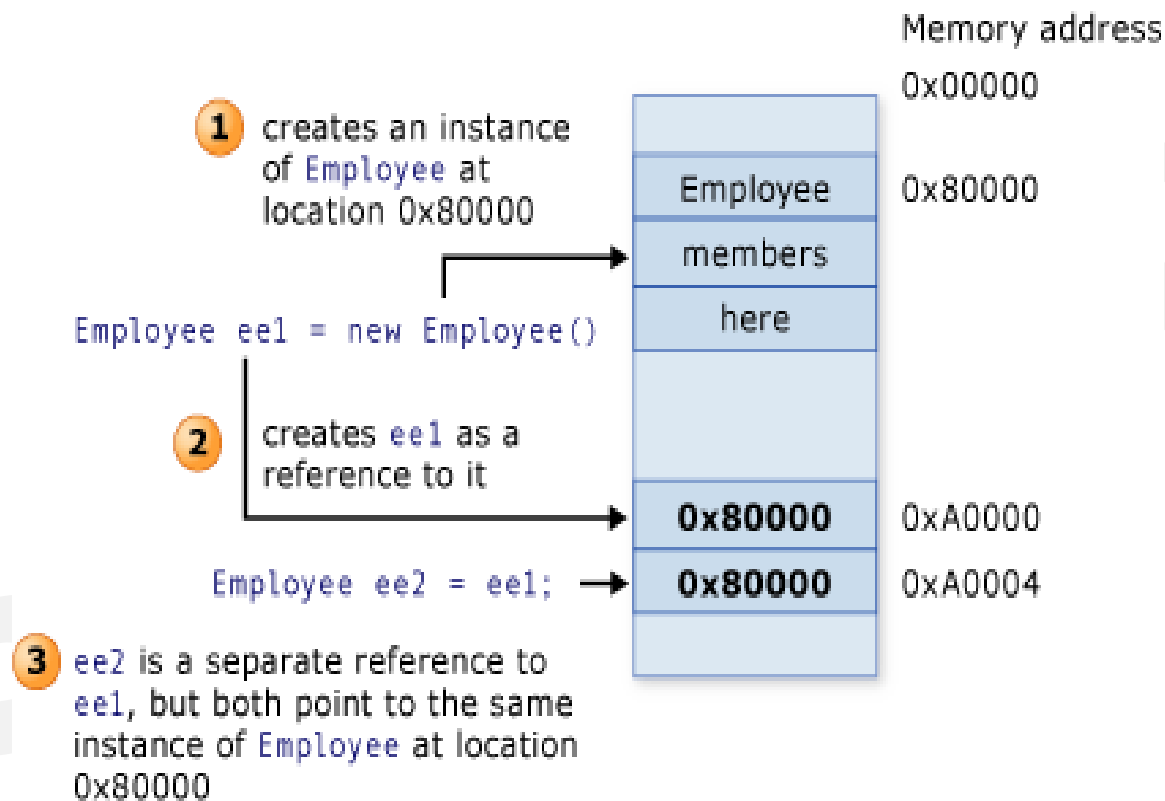
C#.NET for Base

❖ Data types



C#.NET for Base

❖ Data types



C#.NET for Base

❖ Data types

- Một số phương thức chuyển đổi

STT	Phương thức	Mô tả
1	ToBoolean	Chuyển giá trị sang kiểu boolean
2	ToByte	Chuyển sang dạng byte
3	ToChar	Chuyển sang dạng kí tự unicode
4	<b.todatetime< b=""></b.todatetime<>	Chuyển từ dạng integer, string sang dạng ngày tháng trong C#.
5	ToDecimal	Chuyển dữ liệu sang số thập phân
6	ToDouble	Chuyển dữ liệu sang kiểu double
7	ToInt16	Chuyển dữ liệu sang số nguyên 16 bit
8	ToInt32	Chuyển dữ liệu sang số nguyên 32 bit

C#.NET for Base

❖ Data types

- Một số phương thức chuyển đổi

STT	Phương thức	Mô tả
1	ToInt64	Chuyển sang số nguyên 64 bit
2	ToString	Chuyển dữ liệu sang dạng chuỗi
3	ToUInt16	Chuyển sang số nguyên không dấu 16 bit
4	ToUInt32	Chuyển sang số nguyên không dấu 32 bit
5	ToUInt64	Chuyển sang số nguyên không dấu 64 bit

C#.NET for Base

❖ Data types

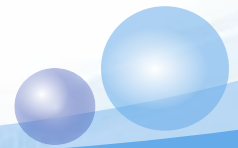
- Một số phương thức chuyển đổi
 - Ví dụ:

```
//Khai báo biến  
string strSo = "68";
```

```
//Chuyển từ dạng chuỗi sang dạng số  
int so_nguyen = Convert.ToInt32(strSo);
```

```
string strNgày = "10/10/1990";
```

```
//Chuyển từ dạng chuỗi sang kiểu ngày tháng  
DateTime dtNgàySinh = Convert.ToDateTime(strNgày);
```

A blue sphere with a white highlight, positioned to the left of the title.

C#.NET for Base

Operators

C#.NET for Base

❖ Operators

Toán tử	Ý nghĩa
<code>+=</code>	Cộng thêm giá trị toán hạng bên phải vào giá trị toán hạng bên trái
<code>-=</code>	Toán hạng bên trái được trừ bớt đi một lượng bằng giá trị của toán hạng bên phải
<code>*=</code>	Toán hạng bên trái được nhân với một lượng bằng giá trị của toán hạng bên phải.
<code>/=</code>	Toán hạng bên trái được chia với một lượng bằng giá trị của toán hạng bên phải.
<code>%=</code>	Toán hạng bên trái được chia lấy dư với một lượng bằng giá trị của toán hạng bên phải.

Toán tử đặc biệt là tăng một (`++`) hoặc giảm một (`--`)

C#.NET for Base

❖ Operators

▪ Toán tử so sánh

Tên toán tử	Kí hiệu	Biểu thức so sánh	Kết quả so sánh
So sánh bằng	==	value1 == 100 value1 == 50	true false
Không bằng	!=	value1 != 100 value2 != 90	false true
Lớn hơn	>	value1 > value2 value2 > value1	true false
Lớn hơn hay bằng	>=	value2 >= 50	true
Nhỏ hơn	<	value1 < value2 value2 < value1	false true
Nhỏ hơn hay bằng	<=	value1 <= value2	false

C#.NET for Base

❖ Operators

▪ Toán tử logic

Tên toán tử	Ký hiệu	Biểu thức logic	Giá trị	Logic
and	&&	$(x == 3) \&\& (y == 7)$	false	Cả hai điều kiện phải đúng
or		$(x == 3) (y == 7)$	true	Chỉ cần một điều kiện đúng
not	!	$! (x == 3)$	true	Biểu thức trong ngoặc phải sai.



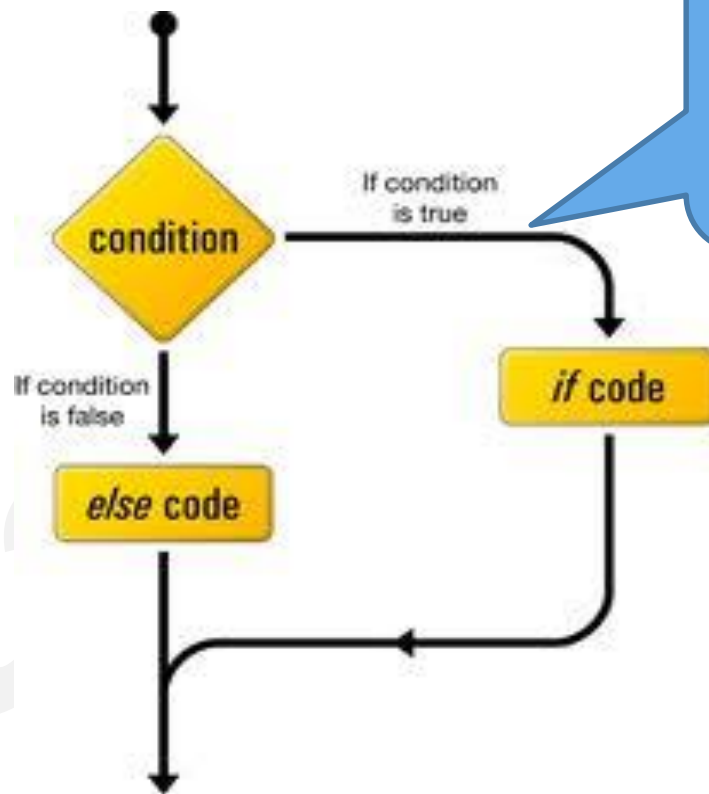
C#.NET for Base

Structured programming

C#.NET for Base

❖ Structured programming

▪ If...Else



Sử dụng để kiểm tra điều kiện nếu đúng sẽ thực hiện một công việc nào đó hoặc ngược lại

C#.NET for Base

❖ Structured programming

▪ If...Else

if (biểu thức điều kiện)

{

<lệnh 1>

<lệnh 2>

....

}

[else

{

<lệnh 1>

<lệnh 2>

...

}]

* Câu lệnh If lồng nhau :

if(biểu thức điều kiện)

{

if(biểu thức điều kiện)

{

<lệnh n>

}

else

{

<lệnh n>

}

}

else

{

<lệnh n>

}

C#.NET for Base

❖ Structured programming

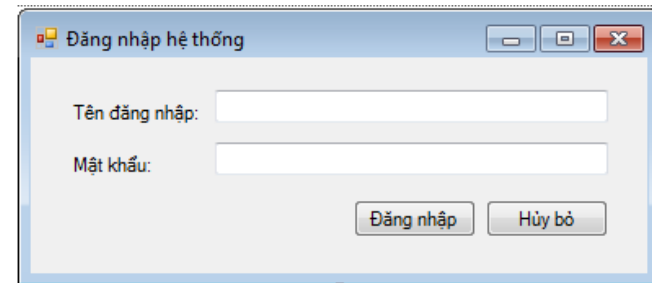
▪ If...Else

- Ví dụ:

```
//Khai báo biến
string username = "";
//Gán giá trị từ textbox trên giao diện
username = txtTenDangNhap.Text.Trim();

//Kiểm tra nếu người dùng không nhập sẽ
//đưa ra thông báo và thoát khỏi chương trình
//không thực hiện công việc tiếp theo nữa
if (username.Length == 0)
{
    //Đưa ra thông báo
    MessageBox.Show("Bạn cần nhập tên đăng nhập", "Thông báo");

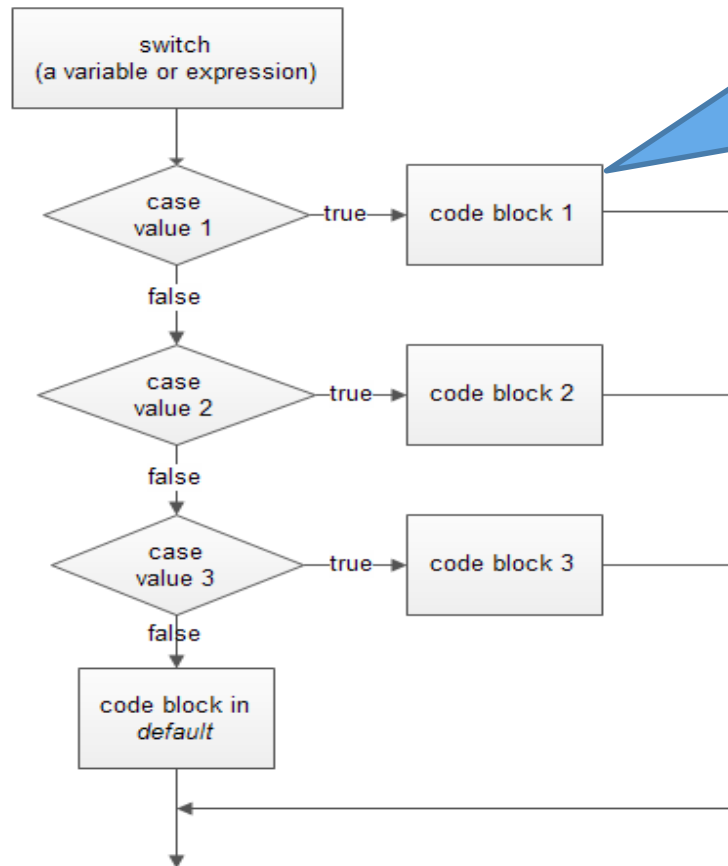
    //Thoát khỏi sự kiện nhấn nút đăng nhập
    //mà không thực hiện công việc tiếp theo nữa
    return;
}
```



C#.NET for Base

❖ Structured programming

▪ Switch...Case



Sử dụng trong trường hợp có nhiều trường hợp cần so sánh

C#.NET for Base

❖ Structured programming

▪ Switch...Case

switch (biểu thức điều kiện)

{

 case <giá trị>:

 <Các câu lệnh thực hiện>

 <lệnh nhảy>

 [default:

 <Các câu lệnh thực hiện mặc định>]

}

C#.NET for Base

❖ Structured programming

▪ Switch...Case

- Ví dụ: Chương trình hiển thị thứ hiện tại trong tuần

```
//Gán ngày trong tuần  
int ngay_trong_tuan = (int)DateTime.Today.DayOfWeek;  
  
//Khai báo biến lưu thứ trong tuần  
string strThu = "";
```

C#.NET for Base

❖ Structured programming

▪ Switch...Case

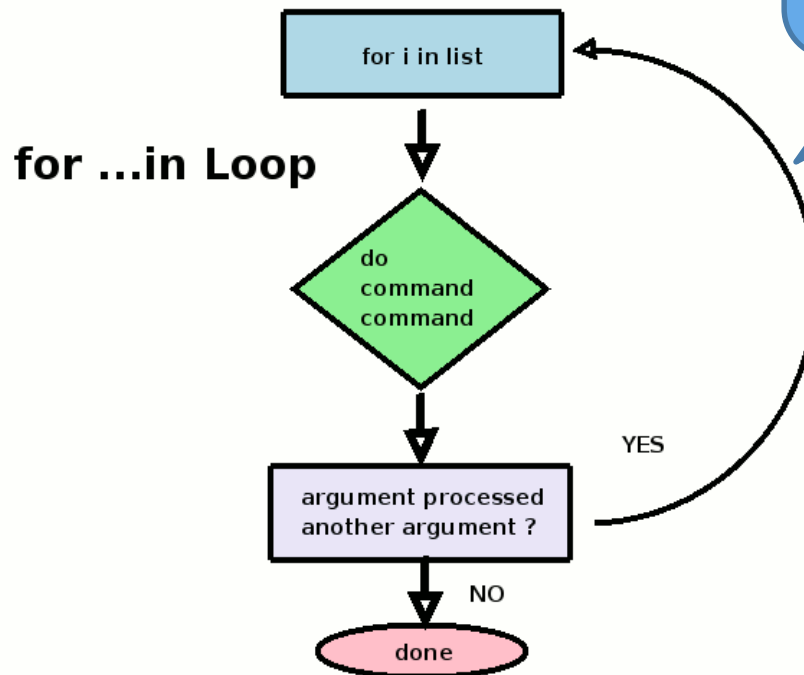
- Ví dụ: Chương trình hiển thị thứ hiện tại trong tuần

```
switch (ngay_trong_tuan)
{
    case 0: strThu = DayOfWeek.Sunday.ToString();
            break;
    case 1: strThu = DayOfWeek.Monday.ToString();
            break;
    case 2: strThu = DayOfWeek.Tuesday.ToString();
            break;
    case 3: strThu = DayOfWeek.Wednesday.ToString();
            break;
    case 4: strThu = DayOfWeek.Thursday.ToString();
            break;
    case 5: strThu = DayOfWeek.Friday.ToString();
            break;
    case 6: strThu = DayOfWeek.Saturday.ToString();
            break;
}
```

C#.NET for Base

❖ Structured programming

▪ Vòng lặp



Xử lý Công việc
giống nhau lặp đi lặp
lại nhiều lần

C#.NET for Base

❖ Structured programming

▪ Vòng lặp

- Sử dụng để thực hiện một số công việc giống nhau lặp đi lặp lại nhiều lần theo một khuôn mẫu.
- Sử dụng các cấu trúc while, do...while, for, foreach trong lập trình C# khi dùng vòng lặp.

C#.NET for Base

❖ Structured programming

- **while:** Thực hiện công việc trong vòng lặp nếu biểu thức điều kiện đúng.

```
while (Biểu thức điều kiện)
{
    <Câu lệnh thực hiện>
}
```

Ví dụ: while(i<5)

```
{
    i++;
    Console.WriteLine("Số thứ tự là: " + i);
}
```

C#.NET for Base

❖ Structured programming

- **Do...while:** Thực hiện công việc trong vòng lặp ít nhất một lần

```
do
{
    <Câu lệnh thực hiện>
}
```

while (Biểu thức điều kiện)

Ví dụ:

```
do { Console.WriteLine("Số thứ tự là:" + i);
}
while (i<0)
```

C#.NET for Base

❖ Structured programming

- **for:** Sử dụng để duyệt một tập phần tử theo các biểu thức điều kiện, bước lặp quy định

for ([phần khởi tạo] ; [biểu thức điều kiện]; [bước lặp])
 <Câu lệnh thực hiện>

C#.NET for Base

❖ Structured programming

▪ for:

- Ví dụ:

```
//Khai báo biến
string strKetQua = "";

//Duyệt vòng lặp for
for (int i = 0; i <= 5; i++)
{
    strKetQua += "Số thứ tự: " + i;
}

//Hiển thị kết quả
lblKetQua.Text = strKetQua;
```

C#.NET for Base

❖ Structured programming

- **foreach:** Duyệt một tập đối tượng từ đầu đến cuối

foreach (<kiểu tập hợp> <tên truy cập thành phần > in < tên tập hợp>)
<Các câu lệnh thực hiện>

Ví dụ:

```
//Khai báo một mảng chuỗi như sau
string[] arrThongTin = new string[3] {"Stanford",
    "Dạy kinh nghiệm lập trình", "Website: www.stanford.com.vn"};

//Duyệt vòng lặp bằng foreach
foreach (string arr in arrThongTin)
{
    strKetQua += arr + "\r\n";
}
```

C#.NET for Base

❖ Các phím tắt hay dùng trong lập trình .NET

- ✓ **Ctrl + J**: Xuất hiện nhanh danh sách tên các đối tượng.
- ✓ **F9**: Đặt điểm dừng chương trình để Debug mã lệnh.
- ✓ **F10**: Duyệt qua từng đoạn mã lệnh khi Debug
- ✓ **F11**: Truy xuất vào bên trong từng đoạn mã lệnh khi Debug
- ✓ **F5**: Để chạy chương trình
- ✓ **Ctrl + Shift + B**: Biên dịch chương trình
- ✓ **Ctrl + Shift + A**: Thêm mới một đối tượng trong danh sách



Thank You !