CHƯƠNG II:

NGÔN NGỮ LẬP TRÌNH C#

2.1. CÁC THÀNH PHẦN CỦA .NET

- Câu lệnh
- Biến
- Hằng
- . . .

1. CÂU LỆNH (Statement)

- Là một chỉ dẫn lập trình đầy đủ để thực thi một hành động
- Được kết thúc bằng dấu ;
- Kết hợp nhiều câu lệnh sẽ tạo thành phương thức (method)
- Ví dụ:

```
Console.WriteLine("Hello World");
```


1. CÂU LỆNH (Statement)

Comment (chú thích): mô tả code, không thực thi, màu xanh lá cây (mặc định)

- Chú thích trên nhiều dòng:

```
/*
  Project: MyFirstCSharpPrj
  Description: displays a Welcome message
*/
```

- Chú thích trên một dòng

```
//Hiển thị ra màn hình console chuỗi: Welcome to C#
```


2. TỪ KHÓA (Keyword)

- Là những từ được giành riêng cho ngôn ngữ .NET, có màu xanh da trời (mặc định)
- Ví dụ:

```
using, class, namespace, int . . .
```


3. ĐỊNH DANH (Identifier)

- Là tên mà bạn định nghĩa cho lớp, biến, hằng, phương thức, namespace…
- Chỉ sử dụng các ký tự chữ cái, chữ số, gạch nối dưới (_)
- Bắt đầu bằng ký tự chữ cái, gạch nối dưới
- C# phân biệt ký tự hoa thường
- Không được đặt trùng tên với keyword
- Ví dụ: _name, TAX_RATE

Ví dụ

- Viết chương trình giải phương trình bậc 2. Chương trình cho phép user nhập vào các hệ số a, b, c. Tính và hiển thị nghiệm của phương trình.

7

4. BIẾN (Variable)

- Là vùng nhớ được đặt tên, chứa giá trị có thể thay đổi được khi chương trình thực thi
  - Đặt tên biến theo quy tắc của định danh, rõ ràng và gợi nhớ
  - Phải khai báo biến trước khi sử dụng
  - Dùng tên để truy xuất và truy nhập biến

8

4. BIẾN (Variable)

- **Cú pháp khai báo biến**:

  kiểu_dữ_liệu tên_biến [= <giá trị>];
- Ví dụ:

```
string fullName= "Tran Van A";
//hoặc
string fullName;
fullName="Tran Van A";
```

9

5. HẰNG

- Tương tự như biến nhưng giá trị không thay đổi khi chương trình thực thi
- 3 loại hằng:
  - Giá trị hằng (literal)
  - Biểu tượng hằng (symbolic constant)
  - Kiểu liệt kê (enumeration)

10

5. HẰNG

Giá trị hằng

x=100; → 100 là giá trị hằng

Biểu tượng hằng

Gán một tên cho một giá trị hằng. Khai báo hằng bằng cú pháp:

const kiểu_dữ_liệu tên_hằng= giá_trị;

- Ví dụ:

```
const float TAX_RATE = 0.1F;
```

11

6. KIỂU DỮ LIỆU

- C# là ngôn ngữ lập trình mạnh về kiểu dữ liệu, phải khai báo kiểu của mỗi đối tượng khi tạo
- 2 tập hợp kiểu dữ liệu
  - Kiểu xây dựng sẵn (built-in)
  - Kiểu người dùng định nghĩa (user-defined)

12

6. KIỂU DỮ LIỆU

Kiểu dữ liệu xây dựng sẵn

Từ khóa trong C#	Số byte	Kiểu trong .NET	Mô tả
byte	1	Byte	Số nguyên dương từ 0 đến 255
sbyte	1	Sbyte	Số nguyên có dấu từ -128 đến 127
ushort	2	UInt16	Số nguyên không dấu từ 0 đến 65.535

13

6. KIỂU DỮ LIỆU

Kiểu dữ liệu xây dựng sẵn

Từ khóa trong C#	Số byte	Kiểu trong .NET	Mô tả
short	2	Int16	Số nguyên có dấu giá trị từ -32768 đến32767
int	4	Int32	Số nguyên từ -2.147.438.648 đến +2.147.438.647
uint	4	UInt32	Số nguyên không dấu từ 0 đến 4.294.967.295

14

6. KIỂU DỮ LIỆU

Kiểu dữ liệu xây dựng sẵn

Từ khóa trong C#	Số byte	Kiểu trong .NET	Mô tả
ulong	8	UInt64	Số nguyên không dấu từ 0 đến +18.446.744.073.709.551.615
long	8	Int64	Số nguyên từ -9.223.372.036.854.775.808 đến +9.223.372.036.854.775.807

15

6. KIỂU DỮ LIỆU

Kiểu dữ liệu xây dựng sẵn

Từ khóa trong C#	Số byte	Kiểu trong .NET	Mô tả
float	4	Single	Số thực với độ chính xác tới 7 chữ số phần thập phân
double	8	Double	Số thực với độ chính xác tới 14 chữ số phần thập phân
decimal	16	Decimal	Số thực với độ chính xác lên tới 28 chữ số phần thập phân

16

6. KIỂU DỮ LIỆU

Kiểu dữ liệu xây dựng sẵn

Từ khóa trong C#	Số byte	Kiểu trong .NET	Mô tả
bool	1	Boolean	Biểu diễn giá trị true hoặc false
char	2	Char	Biểu diễn 1 ký tự Unicode
string		String	Chuỗi các ký tự, mỗi ký tự 2 byte
object		Object	Kiểu dữ liệu cơ bản của tất cả các kiểu khác. Chứa được tất cả các kiểu dữ liệu được kế thừa từ nó

6. KIỂU DỮ LIỆU

Chú ý:

- Kiểu số nguyên:
  - Kiểu số nguyên mặc định là int
  - Giá trị mặc định: 0
  - Hậu tố: uint (U), long (L), ulong (UL/LU)
- Kiểu số có phần thập phân:
  - Kiểu dấu phẩy động mặc định là double
  - Giá trị mặc định: 0.0
  - Hậu tố: float (0.0F), double (0.0D), decimal (0.0M)

18

6. KIỂU DỮ LIỆU

Chuyển kiểu dữ liệu (ép kiểu)

- là chuyển đổi từ một kiểu dữ liệu này sang một kiểu dữ liệu khác.
- 2 cách:
  - Chuyển kiểu ngầm định (implicit type-cast)
  - Chuyển kiểu tường minh (explicit type-cast)

6. KIỂU DỮ LIỆU

Chuyển kiểu ngầm định

- Được thực hiện một cách tự động, an toàn
- Ép từ kiểu (nguồn) có vùng giá trị **nhỏ hơn** so với vùng giá trị mà kiểu (đích) có thể chứa

```
int i = 59;
double x = i;
```

- Ép từ lớp dẫn xuất sang lớp cơ sở

```
string s = "Hello";
object o = s;
```


6. KIỂU DỮ LIỆU

Chuyển kiểu tường minh

Người sử dụng dùng toán tử | các phương thức định nghĩa trước để chuyển đổi kiểu.

- Ép từ kiểu (nguồn) có vùng giá trị **lớn hơn** so với vùng giá trị mà kiểu đích có thể chứa.

```
double x = 74.86;
int i = (int)x; // i = 74
```

- Ép từ lớp cơ sở qua lớp dẫn xuất

```
string s = "Hello";
object o = s;
string s2 = (string)o;
```


6. Kiểu dữ liệu

Các cách để thực hiện chuyển kiểu tường minh

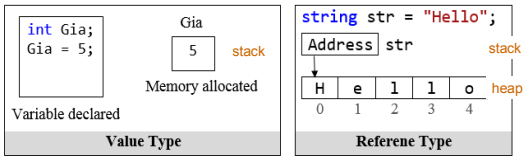
Cách sử dụng	Diễn giải
KiểuDL.ToString() **KiểuDL.Parse()**	Chuyển đổi giữa kiểu string và kiểu cơ sở. Phát sinh ngoại lệ nếu không chuyển được
KiểuDL.TryParse()	Chuyển đổi giữa kiểu string và kiểu cơ sở. Trả về false nếu không chuyển được
(KiểuDL)	Chuyển đổi giữa các kiểu mà toán tử chuyển kiểu định nghĩa
System.Convert	Chuyển đổi giữa các kiểu theo phương thức sử dụng

6. KIỂU DỮ LIỆU

- C# chia các kiểu dữ liệu thành 2 loại
  - **Kiểu giá trị (value)**: chứa trực tiếp giá trị trên địa chỉ của biến đang giữ
  - **Kiểu tham chiếu (reference)**: lưu trữ địa chỉ tham chiếu tới vùng nhớ chứa giá trị thật sự của biến

6. KIỂU DỮ LIỆU

Kiểu dữ liệu giá trị và tham chiếu



6. KIỂU DỮ LIỆU

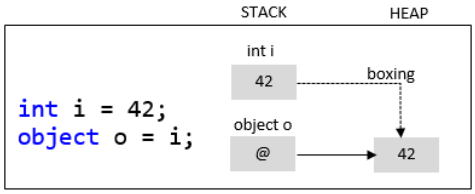
Kiểu dữ liệu giá trị và tham chiếu

- Các kiểu built-in trừ string và object đều là kiểu giá trị
- Các kiểu người dùng định nghĩa trừ struct là kiểu tham chiếu
- Bạn có thể chuyển đổi từ kiểu giá trị sang kiểu tham chiếu và ngược lại qua việc boxing và unboxing

6. KIỂU DỮ LIỆU

Boxing

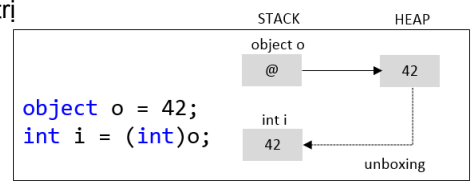
Là quá trình gán giá trị kiểu tham trị về cho đối tượng kiểu object



6. KIỂU DỮ LIỆU

Unboxing

Là quá trình chuyển đổi từ đối tượng object (đã thực hiện boxing) sang kiểu dữ liệu tham trị



7. BIỂU THỨC

- Các câu lệnh thực hiện việc đánh giá một giá trị gọi là biểu thức. Biểu thức bao gồm toán hạng và toán tử hoặc các phép logic
- Tuân theo thứ tự ưu tiên:
  - Trong ngoặc
  - Lũy thừa
  - Số âm
  - Nhân | chia
  - Chia lấy nguyên
  - Chia lấy dư
  - Cộng | trừ

7. BIỂU THỨC

Toán tử

- **Arithmetic** (số học) : **+ - * / %**
- **Assignment** (gán) : **= += -= *= /= %=**
- **Unary** (một ngôi): **++ --**
- **Comparison** (so sánh): kết quả là kiểu boolean sau khi đã so sánh
  < <= = = != > >=
- **Logic:** đánh giá biểu thức và trả về kiểu boolean
  && (và) **||** (hoặc)
  ! (phủ định) **^** (XOR – chỉ sai khi cả hai biểu thức đều đúng)

7. BIỂU THỨC

Toán tử một ngôi

Toán tử	Ý nghĩa	Ví dụ
++	tăng giá trị của toán hạng lên 1	y = ++x → tăng x lên 1 rồi gán giá trị của x cho y y = x++ →gán y bằng giá trị của x rồi tăng x lên 1
- -	giảm giá trị của toán hạng đi 1	y = --x → giảm x rồi gán giá trị của x cho y y=x-- → gán y bằng giá trị của x rồi giảm x đi

7. BIỂU THỨC

Toán tử gán

Toán tử	Ví dụ	Ý nghĩa
=	x = 5	gán giá trị 5 cho biến x
+=	x += y	tương tự: x = x+y
-=	x -= y	tương tự: x = x-y
*=	x *= y	tương tự: x = x*y
/=	x /= y	tương tự: x = x/y
%=	x %= y	tương tự: x = x%y

31