

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI – BÀI GIẢNG ĐIỆN TỬ

LẬP TRÌNH .NET

Bài 1. Tổng quan về Microsoft .NET và ngôn ngữ C# (tiếp)

Căn bản về ngôn ngữ C#

Website: <http://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

MỤC TIÊU BÀI HỌC

- Bài học cung cấp các kiến thức tổng quan về Microsoft .NET và ngôn ngữ C#, đồng thời trang bị các kiến thức và kỹ năng sử dụng hằng, biến, kiểu dữ liệu và các cấu trúc điều khiển cơ bản trong C#.
- Sau khi học xong bài này sinh viên có khả năng:
 - Giải thích được cấu trúc của Microsoft .NET.
 - Trình bày được các đặc điểm của ngôn ngữ C#
 - Sử dụng được hằng, biến, kiểu dữ liệu và các cấu trúc điều khiển cơ bản trong C# để giải một số bài tập lập trình đơn giản.

Website: <http://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

NỘI DUNG BÀI HỌC

- Câu lệnh (Statement)
- Định danh (Identifier)
- Từ khóa (Keyword)
- Biến
- Hằng
- Kiểu dữ liệu
- Biểu thức
- Các cấu trúc rẽ nhánh
- Các cấu trúc lặp

Website: <http://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CÂU LỆNH (STATEMENT)

- Là hành động chúng ta yêu cầu chương trình thực hiện
- Đó có thể là khai báo biến, gán giá trị, gọi phương thức, xử lý điều kiện...
- Câu lệnh được kết thúc bằng dấu chấm phẩy (;)
- Ví dụ:

```
Console.WriteLine("Hello world!");
```

Website: <http://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

KHOẢNG LỆNH (CODE BLOCK)

- Một chuỗi câu lệnh có thể được nhóm lại với nhau tạo thành khối lệnh
- Một khối lệnh được đặt chung trong cặp dấu {}
- Các khối lệnh có thể lồng nhau
- Ví dụ:

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

Website: <http://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CHÚ THÍCH (COMMENTS)

- Chú thích (Comment): là một hoặc nhiều dòng văn bản, được chèn vào mã nguồn chương trình, nhằm làm cho mã nguồn trở nên dễ hiểu hơn với người đọc.
- Chú thích được bỏ qua bởi trình biên dịch.
- Có 3 cách viết chú thích:
 - Chú thích trên 1 dòng
 - Chú thích trên nhiều dòng
 - Chú thích tài liệu

Website: <http://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CHÚ THÍCH (COMMENTS)

- Chú thích trên một dòng:
`//Hiển thị ra màn hình console chuỗi: Welcome to CSharp`
- Chú thích trên nhiều dòng:

```
/*
Project: MyFirstCSharpPrj
Description: displays a Welcome message
*/
```
- Chú thích tài liệu: là chú thích có dạng đặc biệt, có thể được sử dụng để tạo ra tài liệu XML từ các chú thích đó. Chú thích dạng này thường đi kèm thêm các thẻ XML.

```
///Chú thích dùng để tạo tài liệu XML
/*
Đây cũng là chú thích để tạo tài liệu XML
*/
```

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

ĐỊNH DANH (IDENTIFIER)

- Là tên mà ta dùng định nghĩa cho lớp, biến, phương thức, hằng, 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`

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

TỪ KHÓA (KEYWORD)

- Là từ khóa được giành riêng cho ngôn ngữ .NET
- Có màu xanh da trời trong môi trường Visual Studio (mặc định)
- Ví dụ:
`using, class, namespace, int ...`

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

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 identifier, rõ ràng và gọi nhớ
 - Phải khai báo biến trước khi sử dụng
 - Dùng tên để truy xuất và truy nhập biến

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

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";
```

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

HẲNG

- Tương tự như biến nhưng **giá trị không thay đổi** khi chương trình thực thi
- Cú pháp khai báo hằng
`const kiểu_dữ_liệu tên_hằng = giá trị;`
- Ví dụ:
`const float TAX_RATE = 0.1F;`

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

KIỀU DỮ LIỆU

- C# có một số kiểu dữ liệu dựng sẵn (built-in) gọi là kiểu dữ liệu cơ bản
- Có 2 loại:
 - Value Type** (kiểu giá trị): chứa trực tiếp giá trị trên địa chỉ của biến đang giữ
 - Reference Type** (kiểu tham chiếu): chứa địa chỉ mà địa chỉ này đang chứa giá trị của biến

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

KIỀU DỮ LIỆU

- Kiểu dữ liệu giá trị và tham chiếu (xem hình vẽ)
- 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

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

KIỀU DỮ LIỆU

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
short	2	Int16	Số nguyên không dấu từ -32.768 đến +32.7767

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

KIỀU DỮ LIỆU

Từ khóa trong C#	Số byte	Kiểu trong .NET	Mô tả
byte	1	Byte	Số nguyên dương từ 0 đến 255
int	4	Int32	Số nguyên từ -2.147.438.648 đến +2.147.438.647

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

KIỀU DỮ LIỆU

Từ khóa trong C#	Số byte	Kiểu trong .NET	Mô tả
ushort	2	UInt16	Số nguyên không dấu từ 0 đến 65.535
int	4	Int32	Số nguyên từ -2.147.438.648 đến +2.147.438.647

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

KIỀU DỮ LIỆU

Từ khóa trong C#	Số byte	Kiểu trong .NET	Mô tả
uint	4	UInt32	Số nguyên không dấu từ 0 đến 4.294.967.295
long	8	Int64	Số nguyên từ -9.223.372.036.854.775.808 đến +9.223.372.036.854.775.807

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

KIỂU DỮ LIỆU

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
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

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved.

LẬP TRÌNH .NET

KIỂU DỮ LIỆU

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

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved.

LẬP TRÌNH .NET

KIỂU DỮ LIỆU

Từ khóa trong C#	Số byte	Kiểu trong .NET	Mô tả
char	2	Char	Biểu diễn 1 ký tự Unicode
bool	1	Boolean	Biểu diễn giá trị true hoặc false
string		String	Chuỗi các ký tự, mỗi ký tự 2 byte

Có nhiều kiểu dữ liệu xây dựng sẵn khác trong C#, hãy tự tìm hiểu thêm về chúng!

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved.

LẬP TRÌNH .NET

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)

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved.

LẬP TRÌNH .NET

CHUYỂN ĐỔI KIỂU DỮ LIỆU

- Sử dụng phương thức được định nghĩa sẵn trong mọi cấu trúc dữ liệu

Phương thức	Mô tả
ToString([format])	Chuyển đổi giá trị sang chuỗi tương ứng. Nếu bỏ qua việc định dạng chuỗi thì chuỗi kết quả sẽ không được định dạng
Parse(string)	Phương thức tĩnh chuyển đổi một chuỗi thành kiểu dữ liệu tương ứng

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved.

LẬP TRÌNH .NET

CHUYỂN ĐỔI KIỂU DỮ LIỆU

- Sử dụng phương thức tĩnh của lớp **Convert**

Phương thức	Mô tả
ToDecimal(value)	Chuyển đổi giá trị sang kiểu decimal
ToDouble(value)	Chuyển đổi giá trị sang kiểu double
ToInt32(value)	Chuyển đổi giá trị sang kiểu int
ToBool(value)	Chuyển đổi giá trị sang kiểu bool
ToString(value)	Chuyển đổi giá trị sang kiểu string
ToChar(value)	Chuyển đổi giá trị sang kiểu char

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved.

LẬP TRÌNH .NET

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

Website: <http://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

BIỂU THỨC

Toán tử

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

Website: <http://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

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

Website: <http://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

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 1

Website: <http://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CÂU TRÚC IF

Câu lệnh if đơn giản

- Cú pháp


```
if (biểu thức điều kiện)
{
    //Khối lệnh A
}
```
- Ý nghĩa:
 Nếu **biểu thức điều kiện** đúng thì thực hiện **Khối lệnh A**

Website: <http://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CÂU TRÚC IF

Câu lệnh if ...else

Cú pháp

```
if (biểu thức điều kiện)
{
    //Khối lệnh A
}
else
{
    //Khối lệnh B
}
```

Ý nghĩa:

Nếu **biểu thức điều kiện** đúng thì thực hiện **Khối lệnh A**
 Ngược lại, **biểu thức điều kiện** sai thì thực hiện **Khối lệnh B**

Website: <http://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CẤU TRÚC IF

Câu lệnh if ...else if

Cú pháp

```
if (biểu thức điều kiện 1)
{
    //Khối lệnh 1
}
else if (biểu thức điều kiện 2)
{
    //Khối lệnh 2
}
.....
else
{
    //Khối lệnh n
}
```

Ý nghĩa:

Nếu *biểu thức điều kiện 1* đúng thì thực hiện *Khối lệnh 1*

Ngược lại, *biểu thức điều kiện 2* đúng thì thực hiện *Khối lệnh 2*

.....

Ngược lại tất cả các điều kiện trên thì thực hiện *Khối lệnh n*

Website: <https://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CẤU TRÚC IF

Chú ý:

- Nếu khối lệnh chỉ có 1 câu lệnh thì không cần dùng cặp ngoặc nhọn { }
- Nếu có một chuỗi các điều kiện xử lý liên tục, có thể sử dụng nhiều if lồng nhau.

Website: <https://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CẤU TRÚC SWITCH

Cú pháp

```
switch(biểu thức điều khiển)
{
    case giá trị 1:
        //Tập lệnh 1
        break;
    case giá trị 2:
        //Tập lệnh 2
        break;
    ...
    default:
        //Tập lệnh n
        break;
}
```

Ý nghĩa

- case:** liệt kê các trường hợp cần xét.
- giá trị i:** là các giá trị hằng cần so sánh với biểu thức điều khiển.
- Nếu biểu thức điều khiển bằng giá trị *i* thì **tập lệnh i** được thực hiện.
- Nếu các case không thỏa thì thực hiện tập lệnh **default** (nếu có).

Website: <https://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CẤU TRÚC SWITCH

Chú ý:

- Kết thúc phát biểu case phải có break.
- Chỉ sử dụng switch cho các kiểu dữ liệu nguyên thủy như: int, string, char, bool... (biến thuộc kiểu dữ liệu hữu hạn, đếm được) những kiểu khác kể cả float, double nên sử dụng câu lệnh if.
- Các giá trị phải là một hằng số như: 10, "Hà Nội"... Nếu cần tính toán dữ liệu so sánh khi runtime thì nên dùng câu lệnh if.

Website: <https://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CẤU TRÚC SWITCH

Chú ý:

- Các giá trị của các phát biểu case phải khác nhau, không có 2 case có cùng giá trị
- Nếu muốn nhiều case cùng thực hiện một tập lệnh thì viết như sau:

```
...
case giá trị 1:
case giá trị 2:
...
//Tập lệnh chung
```

Website: <https://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CẤU TRÚC WHILE

Thực thi vòng lặp nhiều lần nhưng không biết trước sẽ lặp bao nhiêu lần trước khi vòng lặp thực hiện

Cú pháp

```
while (biểu thức logic)
{
    //Khối lệnh
}
```

Ý nghĩa:

Kiểm tra *biểu thức logic*, nếu thỏa mãn thì thực hiện *khối lệnh*. Tiếp tục thực hiện các lệnh cho đến khi *biểu thức logic* là false

Website: <https://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CẤU TRÚC WHILE

```

...
int n = 1;
while (n < 6)
{
    Console.WriteLine("Current value of n is {0}", n);
    n++;
}
/*
Output:
Current value of n is 1
Current value of n is 2
Current value of n is 3
Current value of n is 4
Current value of n is 5
*/

```

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CẤU TRÚC DO ... WHILE

Cú pháp

```

do
{
    //khởi_lệnh;
    ...
} while (biểu_thức_logic);

```

Ý nghĩa:

Khởi_lệnh trong vòng lặp sẽ được thực thi trước, sau đó kiểm tra **biểu_thức_logic**, nếu thỏa mãn thì khởi_lệnh tiếp tục được thực thi, nếu biểu thức logic không thỏa mãn thì vòng lặp sẽ dừng lại

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CẤU TRÚC DO ... WHILE

```

...
int x = 0;
do
{
    Console.WriteLine(x);
    x++;
} while (x < 5);
/*
Output:
0
1
2
3
4
*/

```

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CẤU TRÚC FOR

Sử dụng khi đã biết trước số lần lặp.

Cú pháp

```

for (khởi_tạo_biến_lặp; điều_kiện_lặp; cập_nhật_biến_lặp)
{
    //khởi_lệnh;
}

```

Trong đó:

- khởi_tạo_biến_lặp**: khai báo và gán giá trị ban đầu cho một biến đếm.
- điều_kiện_lặp**: xác định điều kiện dừng của vòng lặp.
- cập_nhật_biến_lặp**: tăng hoặc giảm giá trị của biến đếm sau mỗi lần lặp.

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CẤU TRÚC FOR

```

...
for (int i = 1; i <= 5; i++)
{
    Console.WriteLine(i);
}
/*
Output:
1
2
3
4
5
*/

```

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CẤU TRÚC FOREACH

Cú pháp

```

foreach (biến_cục_bộ in tập_hợp)
{
    //khởi_lệnh;
}

```

Ý nghĩa:

Lệnh foreach thực hiện duyệt qua từng phần tử trong **tập_hợp** để thực hiện các lệnh trong **khởi_lệnh**.

- biến_cục_bộ**: biến chạy để duyệt qua từng phần tử của **tập_hợp**.
- Kiểu dữ liệu của **biến_cục_bộ** phải phù hợp với kiểu dữ liệu của các phần tử trong **tập_hợp**.

Website: <https://hanoi.edu.vn> © 2021 Hanoi University of Industry All rights reserved

LẬP TRÌNH .NET

CẤU TRÚC FOREACH

```

...
string[] values = {"Toi", "yeu", "Ha", "Noi"};
foreach (string s in values)
{
    Console.WriteLine(s);
}
/*
Output:
Toi
yeu
Ha
Noi
*/

```

Website: <http://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved 43

LẬP TRÌNH .NET

SỬ DỤNG BREAK & CONTINUE TRONG VÒNG LẶP

- Lệnh **break** thực hiện việc dừng vòng lặp.
- Khi chương trình đang chạy mà gặp lệnh **break** thì chương trình sẽ lập tức chấm dứt vòng lặp cho dù điều kiện của vòng lặp vẫn cho phép chạy tiếp.
- Trong trường hợp có nhiều vòng lặp lồng nhau, chương trình sẽ chấm dứt vòng lặp gần với **break** nhất.

Website: <http://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved 44

LẬP TRÌNH .NET

SỬ DỤNG BREAK & CONTINUE TRONG VÒNG LẶP

```

//In ra số nguyên lớn nhất trong khoảng từ 1 đến 100 chia hết cho n
//(0<n<=100)

int i, max = 0;
int n = int.Parse(Console.ReadLine());
for (i = 100; i >= 1; i--)
{
    if (i % n == 0)
    {
        max = i;
        break; //thoát khỏi vòng lặp
    }
}
Console.WriteLine("Số lớn nhất chia hết cho {0} là {1}", n, max);

```

Website: <http://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved 45

LẬP TRÌNH .NET

SỬ DỤNG BREAK & CONTINUE TRONG VÒNG LẶP

- Trong khi thực hiện vòng lặp, người lập trình đôi khi cần thực hiện việc bỏ qua một số dòng lệnh để tiếp tục thực hiện việc lặp cho lần tiếp theo.
- Lệnh **continue** thực hiện việc chuyển sang lần lặp tiếp theo và bỏ qua các lệnh nằm trong vòng lặp nhưng nằm phía sau nó.

Website: <http://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved 46

LẬP TRÌNH .NET

SỬ DỤNG BREAK & CONTINUE TRONG VÒNG LẶP

```

//In ra tổng các số nguyên chẵn từ 1 đến 100

int tong = 0;
for (int i = 1; i <= 100; i++)
{
    if (i % 2 != 0)
        continue;
    tong += i;
}
Console.WriteLine("Tổng các số nguyên chẵn từ 1 đến 100 là:" + tong);
/*
Output:
Tổng các số nguyên chẵn từ 1 đến 100 là:2550
*/

```

Website: <http://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved 47

LẬP TRÌNH .NET

CẢM ƠN ĐÃ CHÚ Ý THEO DÕI !

Website: <http://hvu.edu.vn> © 2021 Hanoi University of Industry All rights reserved 48