

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

LẬP TRÌNH GIAO DIỆN

Chương 2. Ngôn ngữ lập trình C#



1

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

CHUẨN ĐẦU RA

- Phân biệt và so sánh được các đặc điểm của ngôn ngữ C# và ngôn ngữ lập trình đã học (C++)
- Thao tác thành thạo trong môi trường Visual Studio.NET để xây dựng được ứng dụng bằng ngôn ngữ C#
- Sử dụng được cú pháp và ngôn ngữ C# trong lập trình
- Vận dụng được kỹ thuật xử lý ngoại lệ để phát hiện và xử lý lỗi chương trình

ThS. Nguyễn Thị Mai Trang

2

NỘI DUNG

1. Giới thiệu ngôn ngữ lập trình C#
2. Các bước xây dựng một ứng dụng bằng C#
3. Biến, hằng, toán tử trong C#
4. Các kiểu dữ liệu trong C#
5. Các cấu trúc điều khiển
6. Mảng
7. Chuỗi
8. Xử lý ngoại lệ

ThS. Nguyễn Thị Mai Trang

3

3

2.1 Giới thiệu ngôn ngữ lập trình C#

- Là một ngôn ngữ hiện đại, hướng đối tượng.
- Có nguồn gốc từ các ngôn ngữ C, C++
- Dùng để xây dựng nhiều loại ứng dụng: web, dịch vụ web, xử lý văn bản, đồ họa, bảng tính,...
- Hỗ trợ:
 - Garbage collection
 - Exception handling → xử lý ngoại lệ
 - Lambda expressions
 - Language Integrated Query (LINQ)
 - Asynchronous operations
 - ...

ThS. Nguyễn Thị Mai Trang

4

4

Chương 2. Ngôn ngữ lập trình C#

2.2 Các bước xây dựng một ứng dụng bằng C#

5

- Tạo project trong VS.Net, chọn ngôn ngữ C#
- Cấu trúc một chương trình C#
- Thiết kế giao diện
- Viết code
- Biên dịch, thực thi

ThS. Nguyễn Thị Mai Trang

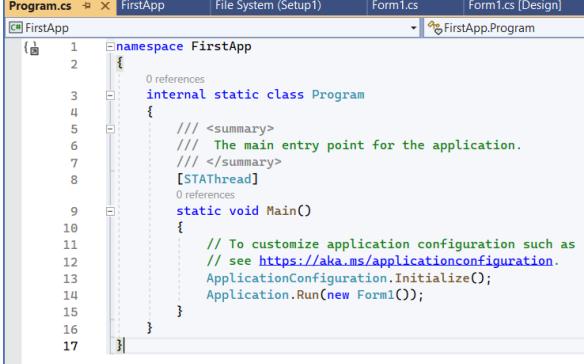
6

6

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

Các bước xây dựng một ứng dụng bằng C#

- Ứng dụng Windows Form với ngôn ngữ C# có duy nhất một hàm Main, nằm trong file program.cs



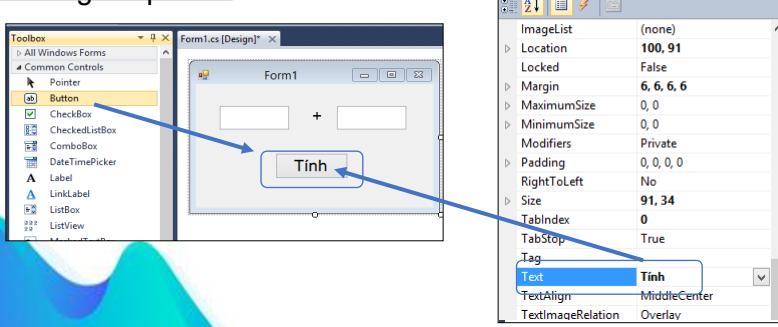
ThS. Nguyễn Thị Mai Trang 7

7

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

Các bước xây dựng một ứng dụng bằng C#

- Thiết kế giao diện:
- Mỗi project có một Form
- Chọn control trên thanh ToolBox kéo thả vào Form
- Thay đổi thuộc tính đối tượng từ bảng Properties



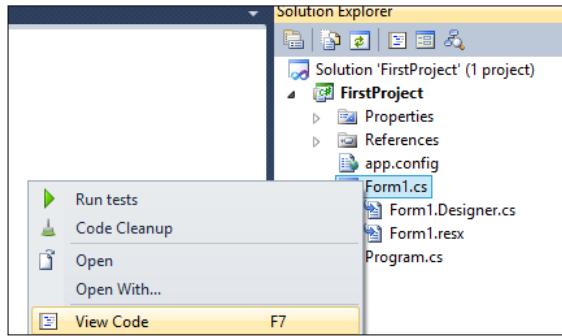
ThS. Nguyễn Thị Mai Trang 8

8

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

Các bước xây dựng một ứng dụng bằng C#

- Viết code:
 - Chuyển sang phần **code-behind** (file.cs)



ThS. Nguyễn Thị Mai Trang 9

9

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

Các bước xây dựng một ứng dụng bằng C# (tt)

```

using System.Windows.Forms;
namespace MyApplication
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
        }
        private void MyFunction()
        {
        }
    }
}
  
```

Bổ sung 22

Vùng khai báo các không gian tên (namespace)

Hàm xử lý sự kiện click chuột trên Button có tên là button1

Hàm người dùng tự định nghĩa

ThS. Nguyễn Thị Mai Trang 10

10

TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

Các bước xây dựng một ứng dụng bằng C#

- Thêm các thành phần vào project

ThS. Nguyễn Thị Mai Trang 11

11

TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

Các bước xây dựng một ứng dụng bằng C#

- Biên dịch, thực thi

- Mã nguồn C# (tập tin *.cs) được biên dịch qua MSIL
 - MSIL (Microsoft Intermediate Language): tập tin .exe hoặc .dll
- MSIL được CLR thông dịch qua mã máy

ThS. Nguyễn Thị Mai Trang 12

12

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

LẬP TRÌNH GIAO DIỆN

Chương 2. Ngôn ngữ lập trình C#



2.3 Biến, hằng, toán tử trong C#

13

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

Biến, hằng, toán tử trong C#

- Quy tắc **định danh** trong C#:
 - **Biến:** `lowerCamelCase`
 - `myRectangle`
 - `myPicture`



- **Kiểu dữ liệu do người dùng định nghĩa:** `UpperCamelCase`
 - `Fraction`
 - `Student`

- **Tránh:**
 - `name`
 - `myVariable, MyVariable`

ThS. Nguyễn Thị Mai Trang

14

14

Biến, hằng, toán tử trong C#

- Biến trong C# được khai báo và sử dụng tương tự như C++
- Cú pháp khai báo biến:
 - [MụcTruyCap] KieuDulieu TenBien [= Giatri];
 - MụcTruyCap: public, protected, private (mặc định)

```
int number = 10;
private int sum = 0;
double radius = 5.0;
protected char myChar = 'A';
```

ThS. Nguyễn Thị Mai Trang

15

15

Biến, hằng, toán tử trong C#

- Tầm vực của biến:
 - Biến khai báo trong phương thức
 - Biến khai báo bên trong class
 - Trong một phạm vi, không thể có hai biến cùng tên

JAVA KHI
KHAI BÁO RẤT ĐÓ

```
internal class Program
{
    static void Main(string[] args)
    {
        int j = 0;
        for (int i = 1; i <= 5; i++)
        {
            j = i * 2; //lỗi
        }
    }
}
```

(Handwritten note: JAVA KHI KHAI BÁO RẤT ĐÓ)

```
internal class Program
{
    int j = 0; //biến j thuộc lớp
    static void Main(string[] args)
    {
        for (int i = 1; i <= 5; i++)
        {
            j = i * 2;
        }
    }
}
```

ThS. Nguyễn Thị Mai Trang

16

16

Biến, hằng, toán tử trong C#

- **Hằng:**

- Khai báo sau từ khóa `const`, phải được khởi tạo khi khai báo
- Giá trị của hằng không được thay đổi khi chương trình thực thi.

- Cú pháp:

```
<const> <type> <CONSTNAME> = <value>;
```

- Ví dụ:

```
const double PI = 3.14158;
public const int MAX = 100;
```

ThS. Nguyễn Thị Mai Trang

17

17

Biến, hằng, toán tử trong C#

- **Toán tử**

Category	Operators
arithmetic	-, +, *, /, %, ++, --
logical	&&, , !, ^
binary	&, , ^, ~, <<, >>
comparison	==, !=, >, <, >=, <=
assignment	=, +=, -=, *=, /=, %=, &=, =, ^=, <<=, >>=
string concatenation	+
type conversion	(type), as, is, typeof, sizeof
other	.., new, (), [], ?:, ??

ThS. Nguyễn Thị Mai Trang

18

18

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

LẬP TRÌNH GIAO DIỆN

Chương 2. Ngôn ngữ lập trình C#



2.4 Các kiểu dữ liệu trong C#

19

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

2.4. Các kiểu dữ liệu trong C#

- Kiểu giá trị (value): kiểu số, ký tự, luận lý, kiểu dữ liệu liệt kê, kiểu cấu trúc (struct)...
- Kiểu dữ liệu tham chiếu (reference): các kiểu dữ liệu do người dùng định nghĩa: class, interface, delegate, array

ThS. Nguyễn Thị Mai Trang

20

20

Các kiểu dữ liệu trong C#

- Predefined types

Type	Description	Examples
object	The ultimate base type of all other types	object o = new Stack();
string	String type; a string is a sequence of Unicode characters	string s = "Hello";
sbyte	8-bit signed integral type	sbyte val1 = 12;
short	16-bit signed integral type	short val1 = 12;
int	32-bit signed integral type	int val1 = 12;
long	64-bit signed integral type	long val1 = 12; long val2 = 34L;
byte	8-bit unsigned integral type	byte val1 = 12; byte val2 = 34U;
ushort	16-bit unsigned integral type	ushort val1 = 12; ushort val2 = 34U;

ThS. Nguyễn Thị Mai Trang

21

21

Các kiểu dữ liệu trong C#

- Predefined types

Type	Description	Examples
uint	32-bit unsigned integral type	uint val1 = 12; uint val2 = 34U;
ulong	64-bit unsigned integral type	ulong val1 = 12; ulong val2 = 34U; ulong val3 = 56L; ulong val4 = 78UL;
float	Single-precision floating point type	float value = 1.23F;
double	Double-precision floating point type	double val1 = 1.23 double val2 = 4.56D;
bool	Boolean type; a bool value is either true or false	bool value = true;
char	Character type; a char value is a Unicode character	char value = 'h';
decimal	Precise decimal type with 28 significant digits	decimal value = 1.23M;

ThS. Nguyễn Thị Mai Trang

22

22

Kiểu dữ liệu liệt kê

- Cú pháp:

```
enum TenKieuDulieu
{
    Giatri1 [= hằng], //mặc định = 0
    Giatri2 [= hằng], //mặc định = 0
    ...
    GiatriN [= hằng] //mặc định = N-1
}
```

- Sử dụng:

```
TenKieuDulieu TenBien;
TenBien = TenKieuDulieu.Giatri;
Ví dụ: Keys.A, Keys.B,...
```

public enum Keys	
	{
...	A = 0x41,
...	B = 0x42,
...	C = 0x43,
...	D = 0x44,
...	E = 0x45,
...	F = 0x46,
...	G = 0x47,
...	H = 0x48,
...	I = 0x49,
...	J = 0x4A,
...	K = 0x4B,

ThS. Nguyễn Thị Mai Trang

23

23

Kiểu dữ liệu cấu trúc (struct)

- struct là kiểu dữ liệu giá trị
- Lưu trữ trên stack trong bộ nhớ
- Lưu trữ các dữ liệu tương đối nhỏ.
- Cú pháp khai báo:


```
struct <TenCauTruc>
{
    <Khai báo các thành phần>
}
```

ThS. Nguyễn Thị Mai Trang

24

24

Kiểu dữ liệu cấu trúc (tt)

- Bao gồm:

- các trường (field) → ?, ?, ?
- thuộc tính (properties)
- phương thức
- phương thức khởi tạo

- Không được gán giá trị khi khai báo

- Không thể ghi đè phương thức khởi tạo không tham số (ngầm định).

- Nếu gọi phương thức khởi tạo với cấp phát new, các thuộc tính chưa khởi trị sẽ được tự động gán trị 0, false hoặc null.

ThS. Nguyễn Thị Mai Trang

25

25

Kiểu dữ liệu cấu trúc

```
struct Thoigian
{
    int gio, phut, giay;
    public Thoigian(int g, int p, int gi)
    {
        gio = g;
        phut = p;
        giay = gi;
    }
    public int Gio
    {
        get { return gio; }
        set { gio = value; }
    }
}
```

```
public int Giay
{
    get { return giay; }
    set { giay = value; }
}
public string HienthiThoigian()
{
    Chuanhoa();
    return String.Format("{0:0} : {1:00} : {2:00}", gio, phut, giay);
}
public void Chuanhoa()
{
    int t = giay / 60;
    giay = giay >= 60 ? giay % 60 : giay;
    phut = phut + t;
    t = phut / 60;
    phut = phut >= 60 ? phut % 60 : phut;
    gio = gio + t;
    gio = gio >= 24 ? gio % 24 : gio ;
}
```

ThS. Nguyễn Thị Mai Trang

26

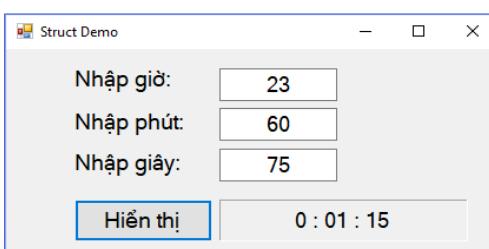
26

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

Kiểu dữ liệu cấu trúc

- Sử dụng kiểu cấu trúc trong Form:

```
private void btHienthi_Click(object sender, EventArgs e)
{
    Thoigian t = new Thoigian (int.Parse(txtGio.Text),
                               int.Parse(txtPhut.Text),
                               int.Parse(txtGiay.Text));
    lbThoigian.Text = t.HienthiThoigian();
}
```



ThS. Nguyễn Thị Mai Trang 27

27

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

LẬP TRÌNH GIAO DIỆN



Chương 2. Ngôn ngữ lập trình C#

2.5 Các cấu trúc điều khiển

28

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

Câu trúc lựa chọn

- Câu trúc if đơn
 - Cú pháp:

```
if (dieukien)
    Khoi_lenh
```

Biểu thức luận lý

```
if (diem >= 60)
    Console.WriteLine("Đậu");
```

 - Lệnh trong Khoi_lenh chỉ thực hiện khi dieukien có giá trị true
- Câu trúc if..else
 - Cú pháp:

```
if (dieukien)
    Khoi_lenh_1
else
    Khoi_lenh_2
```

 - dieukien = true → lệnh trong Khoi_lenh_1
 - Ngược lại, → lệnh trong Khoi_lenh_2.

```
if (diem >= 60)
    Console.WriteLine("Đậu");
else
    Console.WriteLine("Rớt");
```

ThS. Nguyễn Thị Mai Trang

29

29

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

Câu trúc lựa chọn

- if .. else lồng nhau

```
if (diem >= 8)
    Console.WriteLine("Giỏi");
else if (diem >= 6.5)
    Console.WriteLine("Khá");
else if (diem >= 5)
    Console.WriteLine("Trung bình");
else
    Console.WriteLine("Yếu");
```

- Biểu thức điều kiện:
 - dieukien = true → giatri1
 - dieukien = false → giatri2

```
dieukien ? giatri1 : giatri2;
```

```
Console.WriteLine(diem >= 5 ? "Đậu" : "Rớt");
```

ThS. Nguyễn Thị Mai Trang

30

30

Câu trúc lựa chọn

- Câu trúc switch

integer number,
char, string, enum

```
switch (Bien_kiemtra)
{
    case <giatri_1>:
        //code thực hiện nếu Bien_kiemtra = giatri_1
        break;
    case <giatri_2>:
        //code thực hiện nếu Bien_kiemtra = giatri_2
        break;
    ...
    case <giatri_n>:
        //code thực hiện nếu Bien_kiemtra = giatri_n
        break;
    [default]:
        //code thực hiện nếu Bien_kiemtra khác các giá trị trên
        break;
}
```

ThS. Nguyễn Thị Mai Trang

31

31

Câu trúc lựa chọn

- Câu trúc switch: ví dụ

```
public double Tinhtoan (double a, double b, char pheptoan)
{
    double ketqua = 0;
    switch (pheptoan)
    {
        case '+': ketqua = a + b; break;
        case '-': ketqua = a - b; break;
        case '*': ketqua = a * b; break;
        case '/':
            if (b != 0) ketqua = a / b;
            break;
    }
    return ketqua;
}
```

ThS. Nguyễn Thị Mai Trang

32

32

Câu trúc lặp

Câu trúc lặp for
 for (bien_khoi_tao; dieukien; buoc_lap)
 Khoi_lenh

Câu trúc lặp while:
 while (dieukien)
 {
 Khoi_lenh
 }

 dieukien được kiểm tra trước
 - true: thực hiện lệnh trong Khoi_lenh
 - false: thoát khỏi vòng lặp

Câu trúc lặp do .. while
 do
 {
 Khoi_lenh
 }while (dieukien);

Thực hiện lệnh trong Khoi_lenh
 Kiểm tra dieukien
 - true: thực hiện bước lặp tiếp theo
 - false: thoát khỏi vòng lặp

ThS. Nguyễn Thị Mai Trang

33

33

Câu trúc lặp

- Ví dụ: tính tổng các số nguyên từ 1 đến n

```
int tong = 0;
for(int i = 0; i < n; i++)
{
    tong += i;
}
```

```
int tong = 0, i = 1;
while(i <= 10)
{
    tong += i;
}
```

```
int tong = 0, i = 1;
do
{
    tong += i;
} while (i <= 10);
```

ThS. Nguyễn Thị Mai Trang

34

34

Cấu trúc lặp

- break:

- Trong switch: thoát khỏi cấu trúc switch.
- Trong vòng lặp: thoát khỏi vòng lặp trực tiếp chừa nó.

- continue: quay về đầu vòng lặp thực hiện bước lặp kế tiếp

- return: thoát khỏi hàm

```
int tong = 0;
for(int i = 1;i <= 5; i++)
{
    if (n % i == 0) break;
    tong += i;
}
//tong = 1
```

```
int tong = 0;
for(int i = 1;i <= 5; i++)
{
    if (n % i == 0) continue;
    tong += i;
}
//tong = 9
```

ThS. Nguyễn Thị Mai Trang

35

35

LẬP TRÌNH GIAO DIỆN

Chương 2. Ngôn ngữ lập trình C#



2.6 Mảng

36

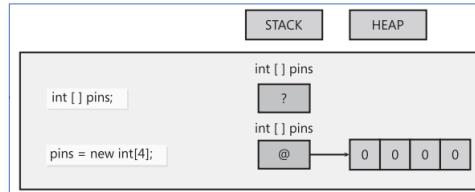
2.6.1 Class Array

- Là lớp cơ sở cho các kiểu tập hợp khác trong .NET

- Kiểu tham chiếu

- Các cách khai báo:

```
int[] pins;  
pins = new int [4];
```



```
int[] pins = new int[4]{ 9, 3, 7, 2 };
```

ThS. Nguyễn Thị Mai Trang

37

37

2.6.2 Thao tác với mảng

- Xác định số phần tử mảng: arrayName.Length

- Sắp xếp mảng: **Array.Sort()**

```
int[] arrayInt = { 5, 7, 3, 8, 2 };  
int size = arrayInt.Length; // 5  
Array.Sort(arrayInt); // 2, 3, 5, 7, 8
```

- Duyệt mảng dựa vào chỉ số như C++:

```
int [] arrInt = { 8, 6, 5, 3, 6, 4 };  
int sum = 0;  
for(int i = 0; i < arrInt.Length;i++)  
{  
    sum += arrInt[i];  
}
```

ThS. Nguyễn Thị Mai Trang

38

38

Thao tác với mảng

- Duyệt mảng dùng foreach:

```
int [] arrInt = { 8, 6, 5, 3, 6, 4 };
int sum = 0;
foreach (int number in arrInt)
{
    sum += number;
}
```

- Duyệt mảng dùng **Array.ForEach**

```
int [] arrInt = { 8, 6, 5, 3, 6, 4 };
int sum = 0;
Array.ForEach(arrInt, number => sum += number);
```

ThS. Nguyễn Thị Mai Trang

39

39

2.6.3 Truyền mảng cho phương thức

- Mảng luôn được truyền bằng tham chiếu.

- Giá trị các phần tử mảng có thể bị thay đổi
- Tham chiếu của biến mảng giữ nguyên

```
static void Main(string[] args)
{
    int[] arrInt = {1, 2, 3};
    Console.Write("Mang khai tao: ");
    foreach (int n in arrInt)
        Console.WriteLine("{0},", n);
    DoubleArray(arrInt);
    Console.WriteLine("\nMang sau khi goi DoubleArray: ");
    foreach (int n in arrInt)
        Console.WriteLine("{0},", n);
    Console.ReadLine();
}

static void DoubleArray(int[] arr)
{
    for (int i = 0; i < arr.Length; i++)
        arr[i] = arr[i] * 2;
    arr = new int[] { 10, 11, 12 };
}
```

Mang khai tao: 1 2 3
Mang sau khi goi DoubleArray: 2 4 6
Kết quả

ThS. Nguyễn Thị Mai Trang

40

40



TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

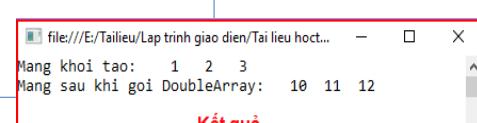
Truyền mảng cho phương thức (tt)

- Truyền mảng với tham chiếu (ref): tham chiếu của biến mảng có thể bị thay đổi.

```
static void Main(string[] args)
{
    int[] arrInt = {1, 2, 3};
    Console.Write("Mang khoi tao: ");
    foreach (int n in arrInt)
        Console.WriteLine("{0},", n);
    DoubleArray(ref arrInt);
    Console.WriteLine("\nMang sau khi goi DoubleArray:");
}
foreach (int n in arrInt)
    Console.WriteLine("{0},", n);
Console.ReadLine();
}
```

```
static void DoubleArray(ref int[] arr)
{
    for (int i = 0; i < arr.Length; i++)
        arr[i] = arr[i] * 2;
    arr = new int[] { 10, 11, 12 };
}
```

↑ thay đổi



```
file:///E/Tailieu/Lap trinh giao dien/Tai lieu hoc...
Mang khoi tao: 1 2 3
Mang sau khi goi DoubleArray: 10 11 12

Kết quả
```

ThS. Nguyễn Thị Mai Trang

41

41

42

Mảng nhiều chiều (tt)

- Khai báo mảng hai chiều:

- Cú pháp: `type[,] array_name;`

- Ví dụ:

```
int [,] arrInt = new int [2,3];
int [,] arrInt = {{1, 2},{3,4}};
int [,] arrInt = new int [,] {{1, 2, 3},
                            {4, 5, 6},
                            {7, 8, 9}};
```

- Duyệt mảng hai chiều:

- `GetLength(n)`: truy xuất số phần tử của mỗi chiều.

- Sử dụng vòng lặp để duyệt qua các hàng và cột

ThS. Nguyễn Thị Mai Trang

43

43

Mảng nhiều chiều (tt)

- Ví dụ

```
int[,] arrInt = { { 1, 2, 3 },
                  { 4, 5, 6 },
                  { 7, 8, 9 } };
int sum = 0;
for (int i = 0; i < arrInt.GetLength(0); i++)
    for (int j = 0; j < arrInt.GetLength(1); j++)
        sum += arrInt[i, j];
```

ThS. Nguyễn Thị Mai Trang

44

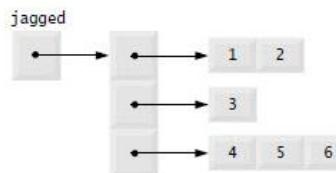
44

Mảng nhiều chiều (tt)

•Zagged array:

- Là mảng hai chiều, trong đó mỗi hàng là một mảng
- Số phần tử trong mỗi hàng không bằng nhau.

```
int[][] jagged = { new int[] { 1, 2 },
                   new int[] { 3 },
                   new int[] { 4, 5, 6 } };
```



ThS. Nguyễn Thị Mai Trang

45

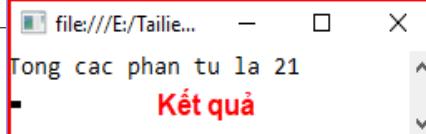
45

Mảng nhiều chiều (tt)

•Duyệt mảng Zagged array:

```
static void Main(string[] args)
{
    int[][] zaggedInt = {new int [] {1, 2},
                          new int [] {3},
                          new int [] {4, 5, 6}};

    int sum = 0;
    foreach (var row in zaggedInt)
        foreach (var number in row)
            sum += number;
    Console.WriteLine("Tong cac phan tu la {0}", sum);
    Console.ReadLine();
}
```



ThS. Nguyễn Thị Mai Trang

46

46

2.6.5 Các lớp tập hợp thông dụng

- ArrayList
- Hashtable, SortedList, Dictionary
- Queue
- Stack
- List
- ...
- Cần khai báo: **using System.Collections;**

ThS. Nguyễn Thị Mai Trang

47

47

ArrayList

- Mảng một chiều, lưu trữ các đối tượng kiểu object, truy xuất thông qua chỉ số.
- Một số thuộc tính: Count, Capacity
- Một số phương thức:
 - Add, Insert, Remove, RemoveAt
 - Contains
 - IndexOf/LastIndexOf
 - Sort

ThS. Nguyễn Thị Mai Trang

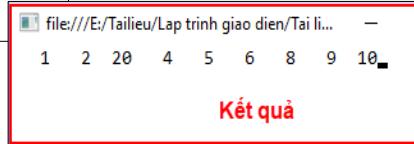
48

48

ArrayList (tt)

- Ví dụ sử dụng ArrayList:

```
static void Main(string[] args)
{
    ArrayList arrInt = new ArrayList();
    for (int i = 1; i <= 10; i++)
        arrInt.Add(i);
    arrInt.Remove(3);
    arrInt.RemoveAt(5);
    arrInt.Insert(2, 20);
    foreach (int number in arrInt)
        Console.WriteLine("{0}, {4}", number);
    Console.ReadLine();
}
```



ThS. Nguyễn Thị Mai Trang

49

49

Các danh sách kiểu từ điển

- Lưu trữ các phần tử là đối tượng của lớp **DictionaryEntry**
- Xác định thông qua hai trường **Key** và **Value**.
- **SortedList**: các phần tử tự động xếp thứ tự theo trường **Key**
- **Hashtable**: tương tự **SortedList**, nhưng không tự động sắp xếp
- **Dictionary**: tương tự hai kiểu trên

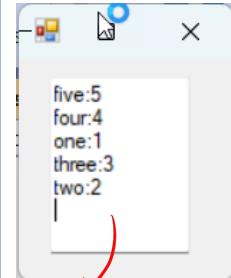
ThS. Nguyễn Thị Mai Trang

50

50

SortedList

```
SortedList list = new SortedList();
list["one"] = 1;
list["two"] = 2;
list["three"] = 3;
list["four"] = 4;
list["five"] = 5;
foreach(DictionaryEntry entry in list)
{
    txtList.AppendText(String.Format("{0}:{1}\r\n",
                                    entry.Key, entry.Value));
}
```



? ? ?

ThS. Nguyễn Thị Mai Trang

51

51

Dictionary

```
Dictionary<string, string> list = new Dictionary<string, string>();
private void Form1_Load(object sender, EventArgs e)
{
    string[] arrEng = { "red", "yellow", "white", "black" };
    string[] arrVn = { "Màu đỏ", "Màu vàng", "Màu trắng", "Màu đen" };
    for (int i = 0; i < arrEng.Length; i++)
        list.Add(arrEng[i], arrVn[i]);
}
private void btTranslate_Click(object sender, EventArgs e)
{
    try {
        txtVn.Text = list[txtEng.Text.ToLower().Trim()];
    }
    catch { }
}
```



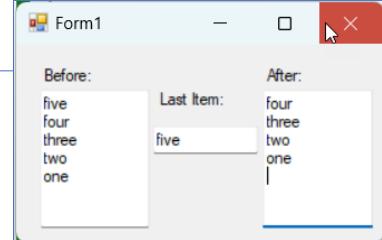
ThS. Nguyễn Thị Mai Trang

52

52

Stack

```
Stack stack = new Stack();
stack.Push("one");
stack.Push("two");
stack.Push("three");
stack.Push("four");
stack.Push("five");
foreach (var number in stack)
    txtList1.AppendText(String.Format("{0}\r\n", number));
txtLastItem.Text = stack.Pop().ToString(); //five
foreach (var number in stack)
    txtList2.AppendText(String.Format("{0}\r\n", number));
```



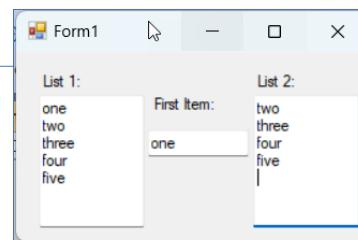
ThS. Nguyễn Thị Mai Trang

53

53

Queue

```
Queue queue = new Queue();
queue.Enqueue("one");
queue.Enqueue("two");
queue.Enqueue("three");
queue.Enqueue("four");
queue.Enqueue("five");
foreach (var number in queue)
    txtList1.AppendText(String.Format("{0}\r\n", number));
txtFirstItem.Text = queue.Dequeue().ToString();
foreach (var number in queue)
    txtList2.AppendText(String.Format("{0}\r\n", number));
```



ThS. Nguyễn Thị Mai Trang

54

54

List

- Danh sách kiểu Generic, cho phép viết code không phụ thuộc vào một kiểu dữ liệu cụ thể

- Lưu trữ danh sách các đối tượng cùng kiểu

- Thao tác tương tự như ArrayList:

 - Add, AddRange, Insert, RemoveAt, Remove,...

- Ví dụ:

```
var arrInt = new List<int>();
var arrInt = new List<int>() {4, 2, 5, 7};
```

ThS. Nguyễn Thị Mai Trang

55

55

List (tt)

- Tạo danh sách chứa các đối tượng:

```
class Circle
{
    double x;
    double y;
    double radius;
    public Circle(double x, double y, double r)
    {
        this.x = x;
        this.y = y;
        radius = r;
    }
}
```

```
var arrCircle = new List<Circle>()
{
    new Circle(100, 50, 30),
    new Circle(150, 60, 50)
};
```

ThS. Nguyễn Thị Mai Trang

56

56

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

LẬP TRÌNH GIAO DIỆN

Chương 2. Ngôn ngữ lập trình C#



2.7 Chuỗi

57

 TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
HO CHI MINH CITY OPEN UNIVERSITY

2.7.1 Lớp String

- Lưu trữ chuỗi các ký tự UTF-16 liên tiếp nhau trong bộ nhớ.
- String constructor:
 - public String(char[] value);
 - public String(char[] value, int startIndex, int length);
 - public String(char c, int count);

```
char[] arrChar = {'b', 'i', 'r', 't', 'h', ' ', 'd', 'a', 'y'};
string string0 = "Welcome to C# programming!";
string string1 = string0; //Welcome to C# programming!
string string2 = new string(arrChar); //birthday
string string3 = new string(arrChar, 6, 3); //day
string string4 = new string('C', 5); //CCCCC
```

ThS. Nguyễn Thị Mai Trang

58

58

Lớp String (tt)

- Các phương thức:
 - CompareTo, Equals
 - EndsWith
 - StartsWith
 - Insert
 - Remove
 - IndexOf/LastIndexOf
 - Split
 - SubString
 - ToCharArray
 - ToLower/ToUpper
 - TrimStart/TrimEnd/Trim

- Các phương thức static:
 - Compare, Equals
 - Concat
 - Copy
 - Format
 - Join

ThS. Nguyễn Thị Mai Trang

59

59

2.7.2 Các thao tác với chuỗi

- So sánh chuỗi

```
bool equal = (s1 == s2);
```

```
bool equal = s1.Equals(s2);
//equal = true nếu s1 = s2
//equal = false nếu s1 != s2
```

```
bool equal = String.Equals(s1, s2);
```

```
int cmp = s1.CompareTo(s2);
//cmp < 0 nếu s1 > s2
//cmp = 0 nếu s1 = s2
//cmp > 0 nếu s1<s2
```

```
int cmp = String.Compare(s1, s2);
```

ThS. Nguyễn Thị Mai Trang

60

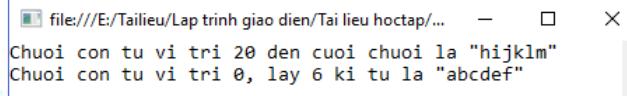
60

Các thao tác với chuỗi (tt)

- Trích chuỗi con: SubString

- SubString (int index)
- SubString (int index, int length)

```
static void Main(string[] args)
{
    string letters = "abcdefghijklmnopqrstuvwxyz";
    Console.WriteLine("Chuoi con tu vi tri 20 den cuoi chuoi la \""
    letters.Substring(20) + "\"");
    Console.WriteLine("Chuoi con tu vi tri 0, lay 6 ki tu la \""
    letters.Substring(0, 6) + "\"");
    Console.ReadLine();
}
```



```
file:///E:/Tailieu/Lap trinh giao dien/Tai lieu hoc tap/... - □ ×
Chuoi con tu vi tri 20 den cuoi chuoi la "hijklm"
Chuoi con tu vi tri 0, lay 6 ki tu la "abcdef"
```

ThS. Nguyễn Thị Mai Trang

61

61

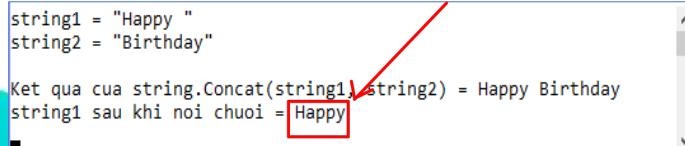
Các thao tác với chuỗi (tt)

- Nối chuỗi:

- Toán tử +

• String.Concat → *đưa vào 1 string → o ảnh hưởng string +*

```
string string1 = "Happy ";
string string2 = "Birthday";
Console.WriteLine("string1 = \"" + string1 + "\"\n" + "string2 = \""
+ string2 + "\"");
Console.WriteLine("\nKet qua cua string.Concat(string1, string2) = "
+ string.Concat(string1, string2));
Console.WriteLine("string1 sau khi noi chuoi = " + string1);
```



```
string1 = "Happy "
string2 = "Birthday"

Ket qua cua string.Concat(string1, string2) = Happy Birthday
string1 sau khi noi chuoi = Happy
```

ThS. Nguyễn Thị Mai Trang

62

62

Các thao tác với chuỗi (tt)

- **Tìm ký tự, tập ký tự, chuỗi con**

- IndexOf, IndexOfAny: hướng tìm từ đầu chuỗi.
- LastIndexOf, LastIndexOfAny: hướng tìm từ cuối chuỗi

```
String str = "Hello World";
char[] ch = { 'o' };
Console.WriteLine(str.IndexOfAny(ch)); //4
char[] ch1 = { 'a', 'l', 'o', 'e', 'd' };
Console.WriteLine(str.IndexOfAny(ch1)); //1
char[] ch2 = { 'a', 'b', 'c' };
Console.WriteLine(str.IndexOfAny(ch2)); //-1
```



e _ d _ o _ l _ l _ e _ W _ o _ r _ l _ d
↓
e _ d _ o _ l _ l _ e _ W _ o _ r _ l _ d

ThS. Nguyễn Thị Mai Trang

63

63

Các thao tác với chuỗi (tt)

- Thay thế chuỗi con trong chuỗi
 - Replace(string oldValue, string newValue)
- Loại bỏ khoảng trắng/ký tự trong chuỗi
 - Trim()
 - TrimStart()/TrimEnd()

```
string s1 = "hello"; string s2 = "GOOD BYE ";
string s3 = "    spaces    ";
Console.WriteLine(" " + s1 + " "); // "hello"
Console.WriteLine(" " + s2 + " "); // "GOOD BYE" → ? ? ?
Console.WriteLine(" " + s3 + " "); // "    spaces    "
Console.WriteLine(s1.Replace('l', 'L'));//heLLo
Console.WriteLine(s1.ToUpper());//HELLO
Console.WriteLine(s2.ToLower());//good bye
Console.WriteLine(" " + s3.Trim() + " "); // "space"
```

ThS. Nguyễn Thị Mai Trang

64

64

Các thao tác với chuỗi (tt)

- Xóa chuỗi con trong chuỗi

- Remove (int index)
- Remove (int index, int count)

```
string s = "ABC---DEF";
string s1 = s.Remove(3);
string s2 = s.Remove(3, 3);
Console.WriteLine("s = {0}", s);
Console.WriteLine("s1 = {0}", s1);
Console.WriteLine("s2 = {0}", s2);
```

→ có ảnh hưởng tới mảng



file:///E:/Tailieu/C#2010... — X

s = ABC---DEF
s1 = ABC
s2 = ABCDEF

Kết quả

ThS. Nguyễn Thị Mai Trang

65

65

Các thao tác với chuỗi (tt)

- Tách chuỗi vào mảng: Split

```
private void SplitString()
{
    string str = "Hello! How are you?.Thank you! ";
    char[] token = { ' ', '\t', '?', '.', '!' };
    string[] arrStr = str.Split(token,
                                StringSplitOptions.RemoveEmptyEntries);
    listBox1.Items.AddRange(arrStr);
}
```



listBox1

Hello
How
are
you
Thank
you

Kết quả

ThS. Nguyễn Thị Mai Trang

66

66

Các thao tác với chuỗi (tt)

- Tạo chuỗi từ mảng: Join

```
public static void Main( string[] args )
{
    string[] arrString = { "This", "is", "a", "text" };
    string s1 = String.Join(" ", arrString);
    string s2 = String.Join("-", arrString);
    Console.WriteLine("s1 = {0}", s1);
    Console.WriteLine("s2 = {0}", s2);
    Console.Read();
} // end Main
```

s1 = This is a text
s2 = This-is-a-text

ThS. Nguyễn Thị Mai Trang

67

67

LẬP TRÌNH GIAO DIỆN

Chương 2. Ngôn ngữ lập trình C#



2.8 Xử lý ngoại lệ

68

Xử lý ngoại lệ

- Ngoại lệ là một lỗi phát sinh không mong muốn khi chương trình thực thi.
- Nguyên nhân: thiếu bộ nhớ, thiếu tài nguyên, thao tác của người sử dụng,...
- Ví dụ:
 - Lỗi chia cho 0
 - Lỗi sai định dạng
 - Lỗi truy xuất một tập tin không tồn tại
 - ...

ThS. Nguyễn Thị Mai Trang

69

69

Xử lý ngoại lệ

• try...catch...finally

- Cú pháp:

```
try{
    // các lệnh có thể phát sinh ra lỗi
}
catch ([Exception]){
    // các lệnh xử lý lỗi
}
finally{
    //các lệnh kết thúc xử lý, luôn thực hiện
}
```

ThS. Nguyễn Thị Mai Trang

70

70

Xử lý ngoại lệ

```

try {
    /*hai lệnh bên dưới sẽ sinh ra lỗi nếu giá trị nhập sai định dạng*/
    int so1 = int.Parse(txtSo1.Text);
    int so2 = int.Parse(txtSo2.Text);
    int ketqua = so1 / so2; //lệnh này có khả năng sinh ra lỗi chia cho 0
    lbKetqua.Text = ketqua.ToString();
}
catch (FormatException) {
    MessageBox.Show("Bạn phải nhập hai số");
}
catch (DivideByZeroException ex) {
    MessageBox.Show(ex.Message, "Lỗi chia cho 0");
}

```

ThS. Nguyễn Thị Mai Trang 71

71

Xử lý ngoại lệ

- throw:** phát sinh một ngoại lệ.

```

try
{
    Console.WriteLine("Cau lenh dau tien");
    throw new Exception("Day la ngoai le duoc nem ra tu lenh throw");
    Console.WriteLine("Lenh nay khong bao gio thuc hien");
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
finally
{
    Console.WriteLine("Lenh nay luon duoc thuc hien");
}

```

ThS. Nguyễn Thị Mai Trang 72

72

Xử lý ngoại lệ

- **checked** và **unchecked**: xử lý lỗi "tràn số"

- checked

```
try {
    int a = 999999, b = 999999;
    int tich = checked(a * b);
    Console.WriteLine("{0} * {1} = {2}", a, b, tich);
}
catch (OverflowException ex) {
    Console.WriteLine(ex.Message);
}
```

- unchecked

```
int number = int.MaxValue;
unchecked
{
    number++;
    Console.WriteLine("Chung toi se nghien cuu loi nay!");
```

ThS. Nguyễn Thị Mai Trang

73

73

Xử lý ngoại lệ

- Một số lớp ngoại lệ trong .NET

Tên lớp ngoại lệ	Ý nghĩa
MethodAccessException	Truy cập đến các thành phần không được phép truy cập
ArrayTypeMismatchException	Kiểu mảng không phù hợp
ArithmetiException	Lỗi liên quan đến các phép toán
DivideByZeroException	Lỗi chia cho 0
FormatException	Lỗi sai định dạng một kiểu dữ liệu
IndexOutOfRangeException	Lỗi truy xuất ngoài chỉ số của mảng
InvalidCastException	Phép gán không hợp lệ
NullReferenceException	Tham chiếu đến một đối tượng null
OutOfMemoryException	Tràn bộ nhớ
OverflowException	Tràn phép toán

ThS. Nguyễn Thị Mai Trang

74

74



75