TRƯỜNG ĐẠI HỌC KINH TẾ TP. HỒ CHÍ MINH
UNIVERSITY OF ECONOMICS HO CHI MINH CITY

Ôn tập: Thủ tục-Hàm & CTDL

Thanh Le, Ph.D.
Khoa Công nghệ Thông tin Kinh doanh

July 16, 2018

Nội dung

- Thủ tục – Hàm
- Cấu trúc dữ liệu
  - Mảng 1 chiều
  - Mảng 2 chiều
  - Mảng nhiều chiều
  - Danh sách, từ điển
- Sử dụng website: https://dotnetfiddle.net/

UEH

Thủ tục-Hàm

- Đơn vị xử lý một vấn đề cụ thể
- Tiếp nhận thông tin từ nơi yêu cầu xử lý thông qua danh sách các tham số
- Có thể trực tiếp phản hồi 1 kết quả xử lý (Hàm) hoặc không (Thủ tục)
- Có thể gián tiếp phản hồi 1 hoặc nhiều kết quả thông qua danh sách tham số

Thủ tục-Hàm trong C#

- C# là ngôn ngữ hướng đối tượng
- Thủ tục-Hàm trong C# được cài đặt thành phương thức của 1 lớp đối tượng
- Việc thực thi Thủ tục-Hàm (của người lập trình) cũng như với hàm thư viện của .Net:

[namespace.][lớp/đốitượng.]tênthủtục/hàm([danh sách tham số])

Cài đặt Thủ tục-Hàm

- Chương trình console đơn giản

```
using System;
public class Program
{
        public static void Main()
        {
                Console.WriteLine("Hello World");
        }
        // các thủ tục / hàm sẽ bắt đầu từ đây
        …
}
```

- Để tiện thực hành, các thủ tục và hàm sẽ đặt trong cùng lớp với hàm Main

UEH

Khai báo Thủ tục (đơn giản)

```
static void tên-thủ-tục([các tham số]) {
        // … các lệnh xử lý của thủ tục
}
```

- static: sử dụng phương thức qua LỚP
- Tên thủ tục: do người lập trình tự đặt
- Mỗi tham số được khai báo như sau

  [ref] kiểu-tham-số tên-tham-số

  ref được sử dụng cho tham số loại tham biến

UEH

Ví dụ thủ tục

```
using System;

public class Program
{
    public static void Main()
    {
        inra("Hello World");
        inra("Hello World again");
    }
    // inra: hiển thị lên màn hình nội dung được Main gửi đến
    //       thông qua tham số "thongbao"
    static void inra(string thongbao) {
        Console.WriteLine(thongbao);
    }
}
```


UEH

Thủ tục gián tiếp trả về giá trị

```
using System;

public class Program
{
    public static void Main()
    {
        string noidung = "Hello World";
        inra(ref noidung);
        inra(ref noidung);
    }
    // inra: hiển thị lên màn hình nội dung được Main gửi đến
    //       thông qua tham số "thongbao"
    //       sau đó, inra sẽ gán chuỗi "DONE" vào tham số
    //       việc đó đó làm thay đổi biến noidung trong Main
    static void inra(ref string thongbao) {
        Console.WriteLine(thongbao);
        thongbao = "DONE";
    }
}
```


UEH

Khai báo Hàm (đơn giản)

static kiểu-trả-về tên-hàm([các tham số]) {
    // … các lệnh xử lý của thủ tục
    return <giá-trị-trả-về>;
}

- Tên hàm: do người lập trình tự đặt
- Mỗi tham số được khai báo như sau
  [ref] kiểu-tham-số tên-tham-số
  ref được sử dụng cho tham số loại tham biến

UEH

Ví dụ hàm

```
using System;

public class Program
{
    public static void Main()
    {
        int kq = tong(10,20);
        Console.WriteLine( "Ket qua: {0}", kq );
        Console.WriteLine( "Ket qua: {0}", tong(40,50) );
    }
    // tong: tính tổng của 2 giá trị được gửi đến trong 2 tham số
    //       so1 và so2
    static int tong(int so1, int so2) {
        return so1 + so2;
    }
}
```


UEH

Các hàm chuỗi System.String

- Length()
- IndexOf("chuỗi") : tìm chuỗi đầu tiên
- LastIndexOf("chuỗi"): tìm chuỗi cuối
- EndsWidth("postfix"): xác minh cuối
- StartWidth("prefix"): xác minh ở đầu
- ToLower() : chuyển chữ thường
- ToUpper() : chuyển chữ hoa
- Trim(): xóa ký tự khoảng trắng thừa 2 đầu

UEH

Ví dụ sử dụng hàm chuỗi

```
using System;

public class Program
{
    public static void Main()
    {
        string hoten = "le nguyen   tuong Vy  ";
        Console.WriteLine( "Vi tri 'le': {0}", hoten.IndexOf("le") );
        Console.WriteLine( "Vi tri 'ng': {0}", hoten.IndexOf("ng") );
        Console.WriteLine( "Ho ten: [{0}]", hoten.Trim() );
        Console.WriteLine( "Viet hoa: {0}", hoten.ToUpper() );
        Console.WriteLine( "Viet thuong: {0}", hoten.ToLower() );
    }
}
```


UEH

Object.ToString()

- <biến>.ToString([“chuỗi-đinh-dạng”)

UEH

Sử dụng ToString()

```
using System;

public class Program
{
    public static void Main()
    {
        double doam = 0.75;
        int nhietdo = 78;
        DateTime thoidiem = DateTime.Now;

        Console.WriteLine( "Ngày: {0}",  thoidiem.Day.ToString() );
        Console.WriteLine( "Tháng: {0}",  thoidiem.Month.ToString() );
        Console.WriteLine( "Năm: {0}",  thoidiem.Year.ToString() );
        Console.WriteLine( "Lúc: {0}",  thoidiem.ToString("hh:mm:ss") );
        Console.WriteLine( "Nhiệt độ: {0}F",  nhietdo.ToString() );
        Console.WriteLine( "Độ ẩm: {0}, hay là: {1}",  doam.ToString("0.000"), doam.ToString("0%") );
        Console.WriteLine( "Thời tiết: {0}",  thoidiem.ToString("dd/MM/yyy hh:mm:ss") );
    }
}
```


UEH

Giải thuật đệ qui (recursion)

- Giải thuật xây dựng trên ý tưởng phân tách 1 vấn đề phức tạp thành vấn đề đơn giản hơn, trong đó vấn đề đơn giản nhất có thể giải được ngay.

UEH

Ví dụ thuật giải đệ qui

- Đặt P_n là vấn đề tính tổng n số nguyên dương đầu tiên: P_n= 1+2+…+n
- Dễ thấy:
  - P_1 = 1
- Theo định nghĩa tổng các số nguyên, thì:
  - $P_2 = P_1 + 2$
  - $P_3 = P_2 + 3$
  - …
  - $P_n = P_{n-1} + n$

UEH

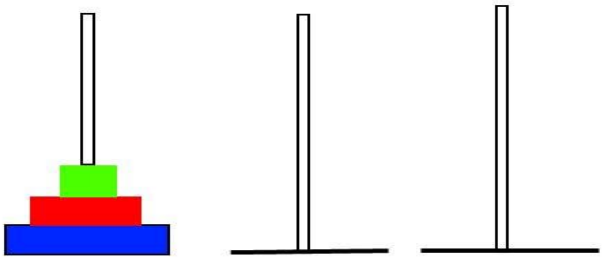
Cài đặt hàm P sử dụng đệ qui

```
using System;

public class Program
{
    public static void Main()
    {
        Console.WriteLine( "Tổng  3 số nguyên dương đầu tiên: {0}", P(3) );
        Console.WriteLine( "Tổng  5 số nguyên dương đầu tiên: {0}", P(5) );
        Console.WriteLine( "Tổng 10 số nguyên dương đầu tiên: {0}", P(10) );
    }
    static int P(int n) {
        if (n == 1) return 1;
        return P(n-1) + n;
    }
}
```


UEH

Tháp Hà nội



UEH

Thuật toán đệ qui tháp HN

```
using System;

public class Program
{
    public static void Main()
    {
        // bài toán tháp Hà nội với 3 đĩa
        chuyen(3, "cột trái", "cột phải", "cột giữa");
    }
    static void chuyen(int sodia, string xuatphat, string dich, string trunggian) {
        if (sodia==1) {
            Console.WriteLine("Chuyển 1 đĩa từ '{0}' sang '{1}'", xuatphat, dich);
        } else {
            // chuyển n-1 đĩa trên cùng từ cột 'xuatphat' sang cột 'trunggian'
            chuyen(sodia-1, xuatphat, trunggian, dich);
            // chuyển 1 đĩa dưới cùng từ cột 'xuatphat' sang cột 'dich'
            chuyen(1, xuatphat, dich, trunggian);
            // chuyển n-1 đĩa trên cùng từ cột 'trunggian' sang cột 'dich'
            chuyen(sodia-1, trunggian, dich, xuatphat);
        }
    }
}
```


UEH

Cấu trúc mảng trong C#

Mảng là một tập hợp có thứ tự của những đối tượng, tất cả các đối tượng này cùng một kiểu. Mảng trong ngôn ngữ C# có một vài sự khác biệt so với mảng trong ngôn ngữ C++ và một số ngôn ngữ khác, bởi vì chúng là những đối tượng. Điều này sẽ cung cấp cho mảng sử dụng các phương thức và những thuộc tính.

Ngôn ngữ C# cung cấp cú pháp chuẩn cho việc khai báo những đối tượng Array. Tuy nhiên, cái thật sự được tạo ra là đối tượng của kiểu System.Array. Mảng trong ngôn ngữ C# kết hợp cú pháp khai báo mảng theo kiểu ngôn ngữ C và kết hợp với định nghĩa lớp do đó thể hiện của mảng có thể truy cập những phương thức và thuộc tính của System.Array.

UEH

Khai báo mảng 1 chiều

Chúng ta có thể khai báo một mảng trong C# với cú pháp theo sau:

<kiểu dữ liệu>[] <tên mảng>

Ví dụ ta có khai báo như sau:

```
int[]    myIntArray;
```


UEH

Khởi động giá trị mảng

```
using System;

public class Program
{
    public static void Main()
    {
        int[] intArray = new int[5]{1,2,3,4,5};
        foreach (int i in intArray) {
            Console.WriteLine(i);
        }
    }
}
```


UEH

Duyệt mảng sử dụng []

```
using System;

public class Program
{
    public static void Main()
    {
        int[] intArray = new int[5]{1,2,3,4,5};
        for (int i=0; i<intArray.Length; i++) {
            Console.Write( "{0} ", intArray[i] );
        }
    }
}
```


UEH

Mảng (ma trận) 2 chiều

```
using System;

public class Program
{
    public static void Main()
    {
        int[,] m2 = new int[2,3]{ {1,2,3} , {4,5,6} };
        for (int i=0; i<m2.GetLength(0); i++) {
            Console.WriteLine(); // new line for each array row
            for (int j=0; j<m2.GetLength(1); j++)
                Console.Write( "{0} ", m2[i,j] );
        }
        Console.WriteLine("\n\nTong so phan tu: {0}", m2.Length);
    }
}
```


UEH

Mảng 2 chiều khác kích thước

```
using System;

public class Program
{
    public static void Main()
    {
        int[] row1 = new int[3]{1,2,3};
        int[] row2 = new int[4]{4,5,6,7};
        int[][] m2 = new int[2][]{row1,row2};
        for (int i=0; i<m2.Length; i++) {
            Console.WriteLine(); // new line for each array row
            for (int j=0; j<m2[i].Length; j++)
                Console.Write( "{0} ", m2[i][j] );
        }
        Console.WriteLine("\n\nTong so dong: {0}", m2.Length);
    }
}
```


UEH

Hàm mảng

- Sort
- Reverse

UEH

Ví dụ hàm mảng

```
using System;

public class Program
{
    public static void Main()
    {
        string[] words = new string[4]{"this","is","a","test"};

        foreach (string s in words)
            Console.Write("{0} ", s);

        Console.WriteLine();

        Array.Sort(words);
        foreach (string s in words)
            Console.Write("{0} ", s);

        Console.WriteLine();

        Array.Reverse(words);
        foreach (string s in words)
            Console.Write("{0} ", s);
    }
}
```


UEH

Danh List và Dictionary

- List là tập hợp các phần tử cùng kiểu
- Dictionary là tập hợp các phần tử cùng kiểu; có thể truy cập các phần tử thông qua mã phần tử (khóa)
- System.Collections.Generic

  Là thư viện .Net cho List và Dictionary

UEH

List các phần tử chuỗi

```
using System;
using System.Collections.Generic;

public class Program
{
    public static void Main()
    {
        List<string> list = new List<string>(){ "mot", "hai", "ba" };
        foreach (string s in list)
            Console.WriteLine(s);
    }
}
```

29

UEH

Dictionary: khóa là số nguyên

```
using System;
using System.Collections.Generic;

public class Program
{
    public static void Main()
    {
        Dictionary<int, string> dict = new Dictionary<int, string>()
        {
            {1, "Một"}, {2, "Hai"}, {3, "Ba"}
        };
        dict.Add(4,"số bốn");

        foreach (int key in dict.Keys) {
            Console.Write("\n{0} => {1}", key, dict[key]);
        }
    }
}
```

30

UEH

Dictionary: khóa là chuỗi

```
using System;
using System.Collections.Generic;

public class Program
{
    public static void Main()
    {
        Dictionary<string, string> dict = new Dictionary<string, string>()
        {
            {"one", "Một"}, {"two", "Hai"}, {"three", "Ba"}
        };
        dict.Add("four","số bốn");

        foreach (string key in dict.Keys) {
            Console.Write("\n{0} => {1}", key, dict[key]);
        }
    }
}
```

31

UEH

Hàm List và Dictionary

- List
  - Add(phần-tử-mới)
  - Remove(giá-trị-xóa)
  - Insert(vị-trí, giá-trị);
- Dictionaty
  - Add(khóa-mới, gia-trị-mới)
  - Remove(khóa-phần-tử-xóa)

32

UEH

Cập nhật List

```
using System;
using System.Collections.Generic;

public class Program
{
    public static void Main()
    {
        List<string> list = new List<string>(){ "mot", "hai", "ba" };

        list.Add("bon");
        foreach (string s in list)
            Console.WriteLine(s);

        Console.WriteLine();
        list.Remove("ba");
        foreach (string s in list)
            Console.WriteLine(s);
    }
}
```

33

UEH

Chèn List

```
/*
 * Tạo 1 List các kiểu string và thêm 2 phần tử vào List.
 */
List<string> MyList4 = new List<string>();
MyList4.Add("Free");
MyList4.Add("Education");

// In giá trị các phần tử trong List
Console.WriteLine(" List ban dau: ");
Console.WriteLine(" So luong phan tu trong List la: {0}", MyList4.Count);
foreach (string item in MyList4)
{
    Console.Write(" " + item);
}
Console.WriteLine();

// Chèn 1 phần tử vào đầu List.
MyList4.Insert(0, "HowKteam");
```

34

UEH

Cập nhật Dictionary

```
using System;
using System.Collections.Generic;

public class Program
{
    public static void Main()
    {
        Dictionary<string, string> dict = new Dictionary<string, string>()
        {
            {"one", "Một"}, {"two", "Hai"}, {"three", "Ba"}
        };

        dict.Add("four","số bốn");
        foreach (string key in dict.Keys) {
            Console.Write("\n{0} => {1}", key, dict[key]);
        }


        Console.WriteLine();
        dict.Remove("three");
        foreach (string key in dict.Keys) {
            Console.Write("\n{0} => {1}", key, dict[key]);
        }
    }
}
```


UEH

Ví dụ: xử lý mảng 1 chiều

```
public static void Main()
{
    int []  n = new int[10]; /* n la mot mang gom 10 so nguyen */
    int i,j;
    /* khoi tao cac phan tu cua mang n */
    for (i = 0; i < 10; i++)
    {
        n[i] = i + 100;
    }

    /* hien thi gia tri cac phan tu cua mang n */
    for (j = 0; j < 10; j++)
    {
        Console.WriteLine("Phan tu [{0}] = {1}", j, n[j]);
    }
}
```


UEH

Ví dụ: xử lý mảng 1 chiều

```
public static void Main()
{
    int []  n = new int[5]; /* n la mot mang gom 10 so nguyen */
    int i,j;
    /* khoi tao cac phan tu cua mang n */
    for (i = 0; i < 5; i++)
    {
        n[i] = i + 100;
    }

    /* hien thi gia tri cac phan tu cua mang n */
    for (j = 0; j < 5; j++)
    {
        Console.WriteLine("Phan tu [{0}] = {1}", j, j+n[j]);
    }
}
```

37

UEH

List và đệ qui: tam giác pascal

```
          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1
1   5  10  10   5   1
```

38

UEH

Tam giác pascal đệ qui

```
using System;
using System.Collections.Generic;

public class Program
{
    public static void Main()
    {
        // dòng thứ 1 trên tam giác pascal
        List<int> ps = pascal(1);

        // dòng thứ 5 trên tam giác pascal
        ps = pascal(6);
        foreach (int m in ps)
            Console.Write("{0} ", m);
    }

    static List<int> pascal(int N) {
        if (N == 1) return new List<int>(){1};
        List<int> pascalN_1 = pascal(N-1);
        List<int> pascalN = new List<int>();
        pascalN.Add(1);
        for (int i=1; i<N-1; i++)
            pascalN.Add( pascalN_1[i-1]+pascalN_1[i] );
        pascalN.Add(1);
        return pascalN;
    }
}
```

39

UEH

Xin cảm ơn!

40

UEH