

NGÔN NGỮ LẬP TRÌNH C#

Ôn tập kiến thức cơ bản

ThS. Nguyễn Chí Hiếu

2021

NỘI DUNG

1. Biến con trỏ
2. Truyền tham trị, tham chiếu
3. Mảng
4. Struct & Class
5. Đệ quy

NỘI DUNG

1. Biến con trỏ

2. Truyền tham trị, tham chiếu

3. Mảng

4. Struct & Class

5. Đề nghị

Biến con trỏ

Biến

Một biến khi được khai báo gồm 3 thuộc tính cơ bản:

- ▶ Tên biến
- ▶ Giá trị của biến
- ▶ Địa chỉ của biến trong vùng nhớ

Biến con trỏ

Cấp phát vùng nhớ của một biến

- ▶ Cấp phát tĩnh
 - ▶ Biến toàn cục (*global*) và biến tĩnh (*static*): chạy chương trình.
 - ▶ Biến cục bộ (*local*): gọi hàm.
- ▶ Cấp phát động
 - ▶ Sử dụng từ khóa `new` để tạo một vùng nhớ dữ liệu trong heap.
 - ▶ Sử dụng từ khóa `delete` để thu hồi vùng nhớ được cấp phát động.
 - ▶ Vùng nhớ cấp phát động trong heap được quản lý thông qua biến con trỏ.

Biến con trỏ

Khái niệm

- ▶ Biến con trỏ (*pointer*) là biến lưu địa chỉ của một kiểu dữ liệu nào đó.
- ▶ Cú pháp

`<Kiểu dữ liệu> *<Tên con trỏ>;`

```
1 int *p1;  
2 float *p2;  
3 SinhVien *sv;
```

- ▶ Chú ý, để sử dụng được biến con trỏ phải cấu hình theo các bước sau: mở Properties/Build của dự án → chọn mục Allow unsafe code → mã nguồn phải thêm từ khóa `unsafe` trước khi khai báo biến con trỏ.

Biến con trỏ

```
1 using System;
2 namespace DSA2021
3 {
4     unsafe class Program // unsafe
5     {
6         static void Main(string[] args)
7         {
8             int a = 10;
9             int *b = &a; // pointer
10            *b += 10;
11
12            Console.WriteLine("value of a = {0}", a);
13            // Console.WriteLine("address of a = {0}", b); // error
14            Console.WriteLine("address of a = {0}", (uint)b);
15        }
16    }
17 }
```

NỘI DUNG

1. Biến con trỏ

2. Truyền tham trị, tham chiếu

3. Mảng

4. Struct & Class

5. Đệ quy

Truyền tham trị và truyền tham chiếu

Truyền tham trị

- ▶ Giá trị của biến không thay đổi sau khi hàm thực hiện xong.

Truyền tham chiếu

- ▶ Giá trị của biến thay đổi sau khi hàm thực hiện xong, thường được dùng để hoán vị hai phần tử trong mảng.
- ▶ Hai loại truyền tham chiếu:
 - ▶ ref: không cần khởi tạo giá trị cho tham số
 - ▶ out: phải khởi tạo giá trị cho tham số

Truyền tham trị và truyền tham chiếu

```
1 using System;
2 namespace Example
3 {
4     class Program
5     {
6         static void Main(string[] args)
7         {
8             int a, b;
9             Console.Write("a = ");
10            a = Int32.Parse(Console.ReadLine());
11            Console.Write("b = ");
12            b = Int32.Parse(Console.ReadLine());
13            // ...
14            Console.WriteLine("a = {0}, b = {1}", a, b);
15        }
16    }
17 }
```

Truyền tham trị và truyền tham chiếu

```
1 static void HoanVi1(int a, int b)
2 {
3     int temp = a;
4     a = b;
5     b = temp;
6 }
```

```
1 static void HoanVi2(ref int a, ref int b)
2 {
3     int temp = a;
4     a = b;
5     b = temp;
6 }
```

NỘI DUNG

1. Biến con trỏ

2. Truyền tham trị, tham chiếu

3. Mảng

4. Struct & Class

5. Đệ quy

Mảng

Mảng 1 chiều

- ▶ Khai báo

```
type[] array;
```

- ▶ Truy xuất phần tử

```
array[i]
```

- ▶ Cần khai báo số lượng phần tử trước khi sử dụng.

Mảng

Mảng 2 chiều

- ▶ Khai báo

```
type[,] array;
```

- ▶ Truy xuất phần tử

```
array[i, j]
```

- ▶ Cần khai báo số lượng phần tử trước khi sử dụng.

Duyệt mảng

for, while, do ... while

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

foreach

```
1 foreach (int num in numbers)
2 {
3     sum += num;
4 }
```

Các vấn đề liên quan đến mảng

- ▶ In mảng
- ▶ Tìm phần tử trong mảng
- ▶ Sắp xếp mảng theo thứ tự tăng hay giảm

NỘI DUNG

1. Biến con trỏ

2. Truyền tham trị, tham chiếu

3. Mảng

4. Struct & Class

5. Đề nghị

Struct & Class

Kiểu cấu trúc (struct)

Đây là một kiểu dữ liệu có cấu trúc và được kết hợp từ nhiều kiểu dữ liệu nguyên thủy khác nhau.

```
struct StructName
{
    type fieldName1;
    type fieldName2;
}
```

Ví dụ 1

Cấu trúc SinhVien quản lý các thông tin sau: Mssv, HoTen, NgaySinh, GioiTinh, QueQuan.

Struct & Class

```
1 struct SinhVien
2 {
3     public string MSSV;
4     public string HoTen;
5     public string NgaySinh;
6     public bool GioiTinh;
7     public string QueQuan;
8     public void Print()
9     {
10         Console.WriteLine(MSSV);
11         Console.WriteLine(HoTen);
12         Console.WriteLine(NgaySinh);
13         Console.WriteLine(GioiTinh);
14         Console.WriteLine(QueQuan);
15     }
16 }
```

Struct & Class

```
1 //...
2 static void Main(string[] args)
3 {
4     SinhVien sv = new SinhVien();
5     sv.MSSV = "123456789";
6     //...
7     //Console.WriteLine(sv.MSSV);
8     sv.Print();
9 }
```


Struct & Class

Kiểu đối tượng

- ▶ Một lớp đối tượng (class) là tập hợp các đối tượng có cùng thuộc tính và hành vi.
- ▶ Cú pháp:

```
class ClassName
{
    type fieldName1;
    type fieldName2;
}
```

Ví dụ 2

Lớp đối tượng SinhVien quản lý các thông tin sau: Mssv, HoTen, NgaySinh, GioiTinh, QueQuan

Struct & Class

```
1 class SinhVien
2 {
3     public string MSSV;
4     public string HoTen;
5     public string NgaySinh;
6     public bool GioiTinh;
7     public string QueQuan;
8 }
```

Struct & Class

Chọn struct hay class?

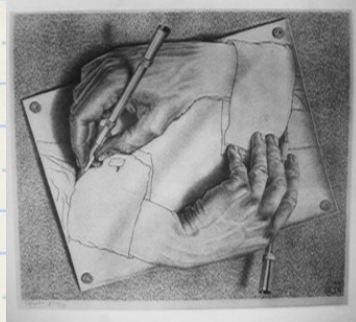
- ▶ Struct: các biến được khai báo trong vùng nhớ stack.
- ▶ Class: các biến thành viên thuộc loại biến tham chiếu và được khai báo trong vùng nhớ heap.

NỘI DUNG

1. Biến con trỏ
2. Truyền tham trị, tham chiếu
3. Mảng
4. Struct & Class
5. đệ quy

Đệ quy

Khái niệm



<https://mcescher.com/gallery/back-in-holland>

- ▶ Đệ quy (*recursion*) là vấn đề/bài toán được định nghĩa bằng chính nó.
- ▶ Một hàm được gọi là đệ quy, nếu bên trong thân của hàm đó có gọi lại chính nó một cách trực tiếp hay gián tiếp.

Đệ quy

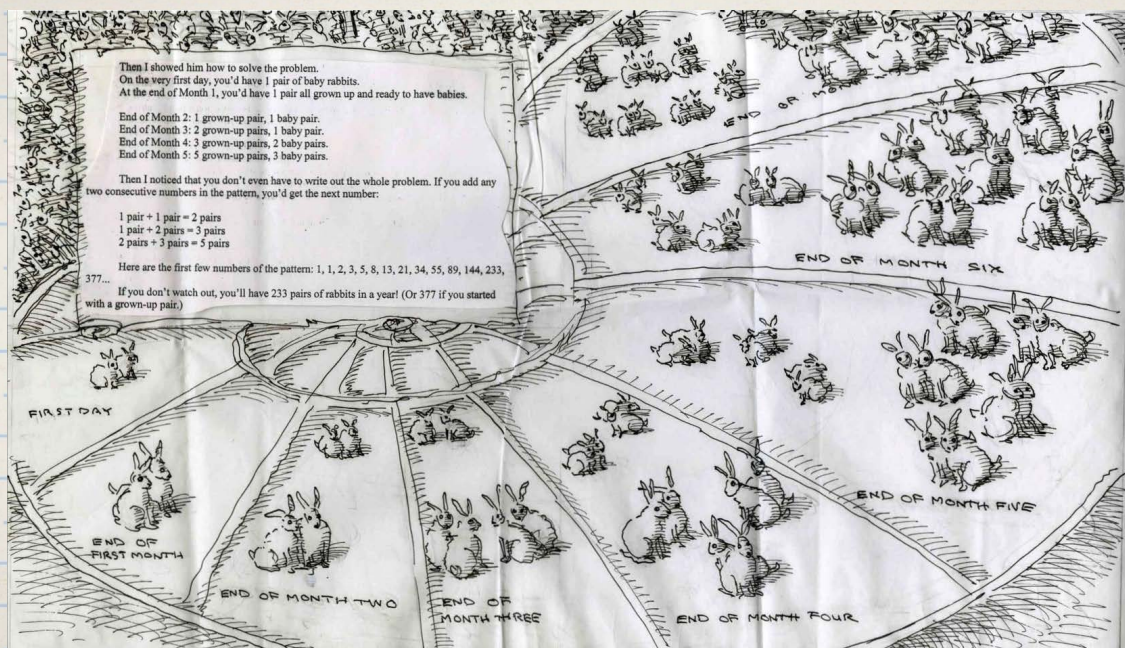
Ví dụ 3

Bài toán nuôi thỏ

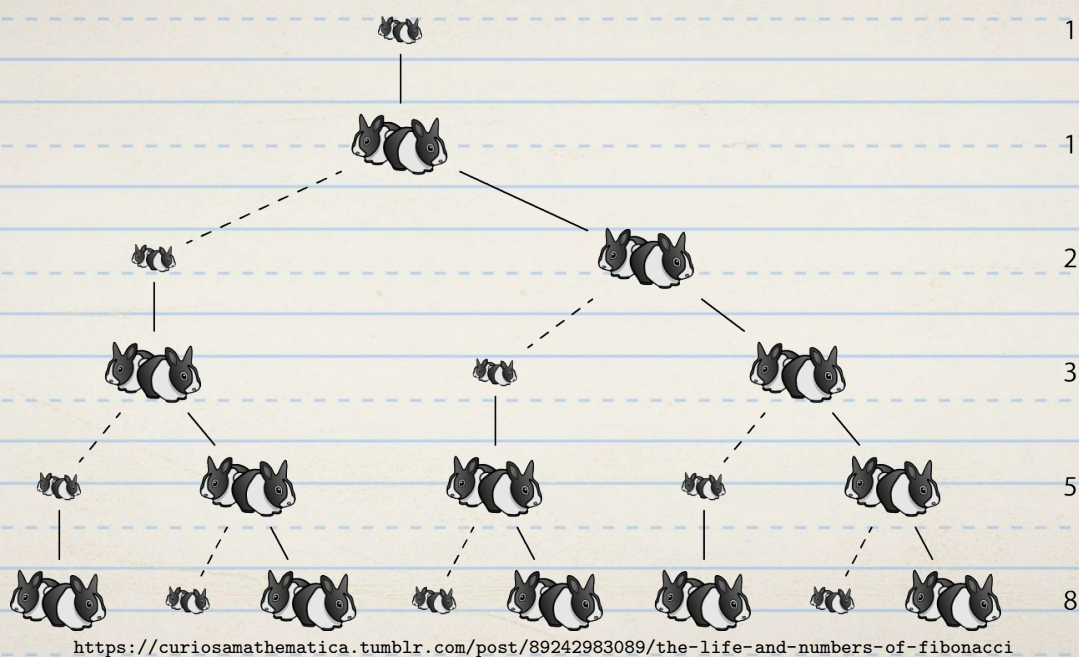
- ▶ Bắt đầu với một thỏ đực và một thỏ cái vừa mới chào đời.
- ▶ Thỏ đạt tới tuổi sinh sản sau một tháng.
- ▶ Thời gian mang thai của một con thỏ là một tháng.
- ▶ Sau khi tuổi sinh sản, thỏ cái đẻ đều đều mỗi tháng.
- ▶ Một thỏ cái sinh ra một thỏ đực và một thỏ cái.
- ▶ Không có thỏ chết.

Hỏi sau một năm sẽ có bao nhiêu cặp thỏ?

Đệ quy



Đệ quy



Đệ quy

Khái niệm

- ▶ Hàm đệ quy gồm hai phần
 - ▶ *Phần cơ sở*: điều kiện dừng quá trình gọi đệ quy.
 - ▶ *Phần đệ quy*: thân hàm chứa lời gọi đệ quy.
- ▶ *Bất kỳ một hàm đệ quy nào cũng phải có điều kiện dừng.*

Ví dụ 4

Cho n là số nguyên không âm, tính $n!$ theo công thức

$$f(n) = \begin{cases} 1 & , n = 0 \\ n \cdot f(n-1) & , n > 0 \end{cases}$$

Phân loại đệ quy

Đệ quy tuyến tính (*Linear recursion*)

- ▶ Trong thân hàm có duy nhất một lời gọi hàm gọi lại chính nó một cách tường minh.
- ▶ Đệ quy đuôi (*Tail recursion*) là một trường hợp của đệ quy tuyến tính với lời gọi hàm nằm ở cuối mỗi hàm.

```
1  type TenHam(ThamSo)
2  {
3      if (DieuKienDung)
4      {
5          ...;
6          return GiaTri;
7      }
8      ...;
9      TenHam(ThamSo);
10 }
```

Phân loại đệ quy

Ví dụ 5

Cho n là số nguyên không âm, $n!$ được định nghĩa như sau

$$f(n) = \begin{cases} 1 & , n = 0 \\ n \cdot f(n-1) & , n > 0 \end{cases}$$

```
1 long Factorial(int n)
2 {
3     if (n == 0)
4         return 1;
5     return n * Factorial(n - 1);
6 }
```

Phân loại đệ quy

Đệ quy nhị phân (*Binary recursion*)

- ▶ Trong thân hàm có hai lời gọi hàm gọi lại chính nó một cách tường minh.

```
1 type TenHam(ThamSo)
2 {
3     if (DieuKienDung)
4     {
5         ...;
6         return GiaTri;
7     }
8     ...;
9     TenHam(ThamSo);
10    ...;
11    TenHam(ThamSo);
12 }
```


Phân loại đệ quy

Ví dụ 6

Dãy Fibonacci được định nghĩa như sau:

$$f(n) = \begin{cases} 1 & , n = 0, 1 \\ f(n-1) + f(n-2) & , n > 1 \end{cases}$$

```
1 long Fibonacci(int n)
2 {
3     if (n <= 1)
4         return 1;
5     return Fibonacci(n - 1) + Fibonacci(n - 2);
6 }
```

Phân loại đệ quy

Đệ quy phi tuyến (*Nonlinear recursion*)

- ▶ Trong thân hàm có lời gọi hàm lại chính nó được đặt bên trong thân vòng lặp.

```
1 type TenHam(ThamSo)
2 {
3     if (DieuKienDung)
4     {
5         ...;
6         return GiaTri;
7     }
8     loop (DieuKieuLap)
9     {
10         ....;
11         TenHam(ThamSo);
12     }
13 }
```

Phân loại đệ quy

Ví dụ 7

Cho hàm $f(n)$ được định nghĩa như sau

$$f(n) = \begin{cases} n & , n \leq 4 \\ f(n-1) + f(n-2) + f(n-3) + f(n-4) & , n > 4 \end{cases}$$

```
1 long F(int n)
2 {
3     int i, result = 0;
4     if (n <= 4)
5         return n;
6     for (i = 1; i <= 4; i ++ )
7         result += F(n - i);
8     return result;
9 }
```

Phân loại đệ quy

Đệ quy tương hỗ (*Mutual recursion*)

- ▶ Trong thân hàm 1 có lời gọi hàm tới hàm 2 và bên trong thân hàm 2 có lời gọi hàm đến hàm 1.

```
1 type TenHam1(ThamSo)
2 {
3     if (DieuKienDung)
4         return GiaTri;
5     TenHam2(ThamSo);
6 }
7 type TenHam2(ThamSo)
8 {
9     if (DieuKienDung)
10        return GiaTri;
11    TenHam1(ThamSo);
12 }
```


Phân loại đệ quy

Ví dụ 8

Cho n là một số nguyên không âm, hãy cho biết n là số chẵn hay lẻ?

$$\begin{cases} n \text{ is even} & , n = 0 \\ \text{IsOdd}(n - 1) & , n > 0 \\ \text{IsEven}(n - 1) & , n > 0 \end{cases}$$

trong đó, kết quả trả về 1 là số lẻ và 0 là số chẵn.

```
1 bool IsOdd(int n)
2 {
3     if (n == 0)
4         return false;
5     return IsEven(n - 1);
6 }
```

```
1 bool IsEven(int n)
2 {
3     if (n == 0)
4         return true;
5     return IsOdd(n - 1);
6 }
```

Bài tập

Cài đặt hàm đệ quy cho các bài toán

1. Tính tổng của n số nguyên dương đầu tiên

$$S(n) = 1 + 2 + 3 + \dots + n - 1 + n, n > 0.$$

2. Tính tổ hợp chập k của n phần tử

$$C_n^k = \begin{cases} 1 & , k = 0 \vee k = n \\ C_{n-1}^k + C_{n-1}^{k-1} & , 0 < k < n \end{cases}$$

3. Tính $f(n)$

$$f(n) = \begin{cases} n & , n \leq 3 \\ f(n-1) + 2f(n-2) + 3f(n-3) & , n > 3 \end{cases}$$

Tài liệu tham khảo



Donald E. Knuth.
The Art of Computer Programming, Volume 3.
Addison-Wesley, 1998.



Dương Anh Đức, Trần Hạnh Nhi.
Nhập môn Cấu trúc dữ liệu và Thuật toán.
Đại học Khoa học tự nhiên TP Hồ Chí Minh, 2003.



Niklaus Wirth.
Algorithms + Data Structures = Programs.
Prentice-Hall, 1976.



Robert Sedgewick.
Algorithms in C.
Addison-Wesley, 1990.