## 1. KIỂU DỮ LIỆU

* Byte: số dương từ 0 tới 255
* Short: số âm và dương
* Int: số nguyên
* Long: số nguyên có miền giá trị lớn
* Float: số thực
* Double: số thực có miền giá trị lớn
* Decimal: tương tự trên
* Char: ký tự
* Bool: logic (true hoặc false)

## 2. HẰNG

* Cú pháp: const <tên kiểu> <tên hằng>=<giá trị>;
* Trong đó: const là từ khóa khai báo
* Ví dụ: const double PI=3.14159265....

## 3. BIẾN

* Cú pháp khai báo 1 biến: <tên kiểu> <tên biến>;
* Ví dụ: float diemso;
* Cú pháp gán giá trị cho biến: <tên biến>=<giá trị>;
* Ví dụ: diemso=1.57;
* Khai báo biến, khởi tạo giá trị cho biến: <tên kiểu> <tên biến>=<giá trị>;
* Ví dụ: float diemso=1.57;
* Biến cục bộ nằm trong hàm Main() còn biến toàn cục và hằng nằm sau class, ngoài hàm Main()

## 4. TOÁN TỬ

* ***Toán tử tăng và giảm***

Cho trước a=10

* ++a: phép tăng trước với biểu thức b=++a, kết quả b=11 và a=11 - a tăng rồi mới gán giá trị cho b
* a++: phép tăng sau với biểu thức b=a++, kết quả b=10 và a=11 - a gán giá trị cho b rồi mới tăng
* Ví dụ: gán a=10. biểu thức a+=5 có biểu thức tương đương là a=a+5, kq là 10
* ***Toán tử logic***
* "&": Phép và sẽ trả về True nếu tất cả điều kiện đều đúng, trả về False nếu có ít nhất một điều kiện bị sai
* + "|" & "||": Phép hoặc sẽ trả về True nếu có ít nhất một điều kiện đúng, trả về False nếu tất cả điều kiện đều sai
* + "^": Phép duy nhất sẽ trả về True nếu có đúng một điều kiện đúng, trả về False nếu có nhiều hơn 1 hoặc không có điều kiện nào đúng.
* ***Toán tử gán***

Cho trước a=10,

* a+=5 đồng nghĩa với a=a+5=15
* a-=5 đồng nghĩa với a=a-5=5
* a\*=5 đồng nghĩa với a=a\*5=50
* a/=5 đồng nghĩa với a=a/5=2 (chia bình thường)
* a%=5 đồng nghĩa với a=a%5=0 (chia lấy nguyên)

## 5. HÀM NHẬP XUẤT DỮ LIỆU

#### Cấu trúc hàm

[Từ khóa 1] [Từ khóa 2] [Từ khóa n] <Kiểu dữ liệu trả về> <Tên hàm>

Trong đó:

* [Từ khóa 1], [Từ khóa 2], [Từ khóa n] là các từ khóa như: public, static, read only … và có thể không điền.
* Kiểu dữ liệu trả về như: từ khóa void, hay mọi kiểu dữ liệu như int, long, bool, SinhVien…
* Tên hàm:
  + Là tên gọi của hàm.
  + Tên bạn có thể đặt tùy ý nhưng nên đặt tên theo quy tắc đặt tên để có sự đồng bộ ngầm định giữa các lập trình viên và dễ tìm, dễ nhớ (xem lại quy tắc đặt tên ở bài [BIẾN TRONG C#](https://howkteam.vn/redirect?Id=Y%2fHKZv6fbvNJnzXtx4Ic0rZRqPOVzQSxG20E6RGIOYFW%2fcTAzGeIqXw0q0jvaA%2beDnsHZapblQXaOVDtPxOtpg%3d%3d)).

#### Thông tin khác

* Hàm in dữ liệu lên màn hình: Console.Write("<Chuỗi ký tự>");

hoặc Console.Write("<Chuỗi ký tự>" + <Biến>);

* Sử dụng line hoặc /n để xuống dòng
* Nhập 1 ký tự: Console.ReadKey();
* Nhập 1 ký tự hoặc 1 chuỗi ký tự: <biến>=Console.ReadLine();
* Nhập 1 số: <biến>=<ép kiểu>Console.ReadLine();
* Ví dụ: Console.Write("tuoi ban");
* tuoi=Convert.ToInt32(Console.ReadLine());
* In ra: Console.WriteLine("Xin chào bạn:{0}, Chúc mừng sinh nhật lần thứ:{1}",Hoten,Tuoi);

ví dụ: in lên màn hình giá trị của a và b

*class program*

*{*

*static void Main(string[] args)*

*{*

*int a=5, b=10;*

*Console.Write("a="+a)*

*Console.WriteLine("b="+b)*

*Console.ReadKey()*

*}*

*}*

ví dụ: in lên màn hình giá trị của a + b

*class program*

*{*

*static void Main(string[] args)*

*{*

*int a=5,b=10;*

*Console.WriteLine("a={0}, b={1}",a,b);*

*Console.Write("a+b={0}",(a+b));*

*Console.ReadKey();*

*}*

***MỘT SỐ HÀM ÉP KIỂU***

* Convert.ToInt32 or int.Parse : ép thành số nguyên
* Convert.ToDouble or double.Parse : ép thành số thực
* Convert.ToChar or char.Parse : ép thành một ký tự

## 6. CẤU TRÚC ĐIỀU KHIỂN

***Hàm if***

*If (<Biểu thức logic>)*

*{*

*<lệnh>;*

*}*

*Else if (<Biểu thức>)*

*{*

*<lệnh>;*

*}*

*Else*

*{*

*<lệnh>;*

*}*

*\*Trong hàm if nếu chỉ có 1 câu lệnh trong thì không cần {}*

#### Hàm While

*While (<Biểu thức logic>)*

*{*

*<lệnh>;*

*}*

\*Hoạt động:

B1: Tính giá trị biểu thức logic

B2: Nếu true thì thực hiện rồi quay lại b1, false thì kết thúc

#### Hàm Do-While: ngược lại với while - thực hiện lệnh trước rồi mới kiểm tra điều kiện.

*Do*

*<lệnh>*

*While (<Biểu thức logic>);*

\*Hoạt động:

B1: thực hiện lệnh/khối lệnh

B2: tính giá trị biểu thức logic, đúng thì quay lại b1, sai thì kết thúc

#### Hàm for

*For (<Lệnh 1>; <Biểu thức logic>; <Lệnh 2>)*

*<Lệnh>;*

\*Hoạt động

B1: thực hiện biểu thức 1

B2: tính biểu thức logic. đúng thì thực hiện rồi xuống b3, sai thì kết thúc

B3: thực hiện biểu thức 2 rồi quay lại b2

#### Câu lệnh nhảy BREAK, CONTINUE

\*Lệnh Break: thoát khỏi cấu trúc lặp gần nhất

\*Lệnh Continue: bỏ qua các lệnh còn lại của vòng lặp, bắt đầu chu trình lặp tiếp theo

## 7. MẢNG MỘT CHIỀU

#### Khai báo mảng

Cách 1

<tên kiểu>[ ]<tên mảng>; //int[ ] A;

<tên mảng>=new <tên kiểu> [size]; // A=new int [100];

Cách 2

<tên kiểu>[ ] <tên mảng>=new<tên kiểu>[size];

Cách 3

int n=int.Parse(console.readline());

int [ ] A=new int[n]

Ví dụ: khai báo mảng điểm, lưu trữ điểm tối đa của 10 sinh viên

C2: float [ ] diem=new float [10]

C3:

int n;

float [ ] diem;

console.write (" số lượng sinh viên: ")

n=int.parse(console.readline());

diem=new float [n];

#### Gán giá trị cho mảng

<Tên kiểu>[] <Tên mảng> = {giá trị 1, giá trị 2, ..., giá trị n}; //int [ ] A= {10,15,11};

or

<Tên mảng>[<số chỉ mục>]=<Giá trị>; //int[ ]A=new int [10]

#### Truy xuất mảng

Cho biết độ dài của mảng: <Tên mảng>.Length

Trả về số phần tử ở chiều tương ứng (độ dài): < tên mảng>.GetLength(<số chiều>)

#### Cấu trúc lặp foreach - duyệt qua từng phần tử trong mảng

foreach (<Tên kiểu> <Tên biến> in <Tên mảng>)

Lệnh;

Vd:

int [ ] A={1,2,3,4,5};

C1:

console.writeline(“cach 1:”);

for (int i=0;i<5;i++);

console.writeline(A[i]);

C2:

console.writeline(“cach 2:”);

foreach (int So in A)

console.writeline(So);

#### Sao chép mảng

Cách 1: Hàm Array.Copy()

Array.Copy(<Mảng nguồn>, <Mảng đích>, <Số phần tử>)

Trong đó: mảng nguồn là mảng cần copy, mảng đích là mảng đc tạo ra

Cách 2: sao chép từng phần tử trong mảng

int [ ] A={1,2,3,4,5}

int [ ] B= new int [5]

for (int i=0; i<A.length; i++)

B[i]=A[i]

B=A: hai mảng này như nhau, khi thay đổi 1 phần tử của 1 trong 2 thì mảng còn lại cũng thay đổi

## 8. MẢNG HAI CHIỀU

Cách 1

<Tên kiểu>[] <Tên mảng>;

<Tên mảng> = new <Tên kiểu>[size1,size2];

Cách 2

<Tên kiểu>[,] <Tên mảng> = new <Tên kiểu>[size1,size2];

size 1: số dòng

size 2: số cột

Cách 3

int size1,size2;

int[,] A;

size1=int.Parse(console.readline()); - dòng

size2=int.Parse(console.readline()); - cột

A=new int [size1, size2];

#### Nhập mảng 2 chiều

<Tên kiểu>[,] <Tên mảng> = new <Tên kiểu>[số dòng,số cột]

*for (int i = 0, i < size1; i++)*

*for (int j = 0, j < size2; i++)*

*{*

*Console.Write("Nhap A[{0},{1}] =",i+1,j+1);*

*A[i,j] = int.Parse(Console.Readline());*

*}*

***Xuất mảng 2 chiều***

*for (int i = 0, i < <Mảng>.GetLength(0) ; i++)*

*{*

*for (int j = 0, j < <Mảng>.GetLength(1); i++)*

*{*

*Console.Write(A[i,j]+" ");*

*}*

*Console.WriteLine();*

*}*

#### Khai báo hàm:

[<Phạm vi>] [static] <Kiểu> <Tên hàm> (<DS/Tham số>)

{

<Lệnh>;

}

#### DS/Tham số:

[ref/out] <kiểu> <biến>, [ref/out] <kiểu> <biến>, …

ref là input và output, out là output, để trống là input