Từ khóa null trong C#

null là một giá trị cố định nó biểu thị không có đối tượng nào cả, có nghĩa là biến có giá trị null không có tham chiếu (trỏ) đến đối tượng nào (không có gì).

null chỉ có thể gán được cho các biến kiểu tham chiếu (biến có kiểu dữ liệu là các lớp), không thể gán null cho những biến có kiểu dữ liệu dạng tham trị như int, float, bool ...

Ví dụ lớp có lớp MyClass thì biến kiểu lớp này có thể gán giá trị null

class MyClass {

public void ShowData() {

Console.WriteLine("Show Data ... ");

}

}

MyClass refvar1, refvar2;

refvar1 = new MyClass(); // refvar1 tham chiếu (gán) bằng một đối tượng

refvar2 = refvar1; // refvar1, refvar2 cùng tham chiếu một đối tượng

refvar1 = null; // refvar1 gán bằng null => không trỏ đến đối tượng nào

refvar2.ShowData(); // refvar2 có trỏ đến đến tượng, nên có thể truy cập các thành viên của đối tượng

refvar1.ShowData(); // refvar1 không trỏ đến đối tượng nào, nên truy cập thành viên sẽ lỗi

int myvar = 10; // int là kiểu tham trị, nó có thể gán giá trị cho biến myvar (10)

int myvar = null; // lỗi - kiểu tham trị không được gán null hay bằng tham chiếu đến đến tượng

Thường để kiểm tra biến có tham chiếu đến đối tượng cụ thể nào sử dụng toán tử logic kiểm tra object != null

## Sử dụng nullable trong C#

Nếu bạn muốn sử dụng các kiểu dữ liệu nguyên tố như int, float, double ... như là một kiểu dữ liệu dạng tham chiếu, có thể gán giá trị null cho nó, có thể sử dụng như đối tượng ... thì khai báo nó có khả năng nullable, khi biến nullable có giá trị thì đọc giá trị bằng truy cập thành viên .Value, cách làm như sau:

Khi khai báo biến có khả năng nullable thì thêm vào ? sau kiểu dữ liệu

int? bienkieuint; // Hoặc Nullable<int> bienkieuint;

bienkieuint = null; // có thể gán null cho biến

bienkieuint = 10; // có thể gán giá trị cho biến

if (bienkieuint != null)

{

int val = bienkieuint.Value; // đọc giá trị trong biến nullable

}

Chú ý dạng khai báo đầy đủ của int? là Nullable<int>