

# Bài: Vòng lặp Do While trong C#.

Xem bài học trên website để ủng hộ Kteam: [Vòng lặp Do While trong C#.](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

## Dẫn nhập

Ở các bài học trước, chúng ta đã cùng nhau tìm hiểu về [CẤU TRÚC VÒNG LẶP WHILE TRONG C#](#). Ở bài này chúng ta sẽ cùng đi sâu vào chi tiết cách sử dụng **vòng lặp do while trong C#**.

## Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về các phần:

- [CẤU TRÚC CƠ BẢN MỘT CHƯƠNG TRÌNH TRONG C# console application](#)
- [BIẾN](#) và [KIỂU DỮ LIỆU](#) trong C#
- [TOÁN TỬ TRONG C#](#)
- [CÂU ĐIỀU KIỆN TRONG C#](#)
- [CẤU TRÚC CƠ BẢN VÒNG LẶP](#)
- [CẤU TRÚC VÒNG LẶP WHILE TRONG C#](#)

Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

- Cấu trúc của một vòng lặp **do while** trong C#
- Các ví dụ sử dụng **do while** trong C#

## Cấu trúc của một vòng lặp do while trong C#

Cú pháp:

```
do
{
    // khối lệnh lặp lại
} while (<Điều kiện lặp>;
```

- **Điều kiện lặp** là một biểu thức logic bắt buộc phải có với kết quả trả về bắt buộc là **true** hoặc **false**.
- Từ khóa **do while** biểu thị đây là một vòng lặp **do while**. Các câu lệnh trong khối lệnh sẽ được lặp lại đến khi không còn thỏa mãn **điều kiện lặp** sẽ kết thúc vòng lặp **do while**.
- Tiến trình:
  - Đầu tiên trình biên dịch sẽ đi vào dòng **do** và thực hiện khối lệnh bên trong. Sau đó khi gặp ký tự **}** sẽ kiểm tra điều kiện lặp có thỏa mãn hay không. Nếu kết quả là **true** thì sẽ quay lại ký tự **{** thực hiện khối code. Quá trình chỉ kết thúc khi điều kiện lặp là **false**.
  - Điều kiện lặp luôn bằng **true** thì vòng lặp **while** sẽ trở thành vòng lặp vô tận.
  - Điều kiện lặp luôn bằng **false** thì vòng lặp sẽ không được thực thi.

**Lưu ý:** vòng lặp **do while** sẽ thực hiện câu lệnh trong khối code xong rồi mới kiểm tra điều kiện lặp. Cuối vòng lặp **do while** có dấu **;** ở cuối.

## Các ví dụ sử dụng do while trong C#

Chúng ta cùng thử một ví dụ đầy đủ về các phần của vòng lặp **do while** nhé.

- **Ví dụ:** In ra một ma trận số

:

```
static void Main(string[] args)
{
    int countLoop = 0;
    int countLoopTime = 0;
    int valueNum = 10;
    int loopTime = 5;

    // Vẽ từ trên xuống LoopTime lần
    do
    {
        // reset lại biến countLoop về 0 để viết lại từ đầu
        countLoop = 0;

        // vẽ từ trái qua valueNum lần
        while (countLoop < valueNum)
        {
            // in ra giá trị của countLoop trong 8 vị trí
            Console.Write("{0,8}", countLoop);

            // tăng giá trị của biến countLoop lên một đơn vị
            countLoop++;
        }

        // Mỗi khi hoàn thành một vòng lặp nhỏ thì lại xuống dòng chuẩn vị vẽ lần tiếp theo
        Console.WriteLine();

        // tăng giá trị countLoopTime lên một đơn vị
        countLoopTime++;

    } while (countLoopTime < loopTime);
    Console.ReadKey();
}
```

- Kết quả một ma trận số xuất hiện trên màn hình

- Chúng ta cũng có thể in ra ma trận số với giá trị ngẫu nhiên bằng cách sử dụng lớp **Random**. (Phần sử dụng lớp Random một cách chi tiết sẽ được hướng dẫn ở bài RANDOM TRONG C#)

:

```

static void Main(string[] args)
{
    int countLoop = 0;
    int countLoopTime = 0;
    int valueNum = 10;
    int loopTime = 5;
    int minRandomValue = 0;
    int maxRandomValue = 100;

    Random rand = new Random();

    // Vẽ từ trên xuống LoopTime lần
    do
    {
        // reset lại biến countLoop về 0 để viết lại từ đầu
        countLoop = 0;

        // vẽ từ trái qua valueNum lần
        do
        {
            // giá trị sinh ngẫu nhiên trong khoảng [minRandomValue . . maxRandomValue - 1]
            int showValue = rand.Next(minRandomValue, maxRandomValue);

            // in ra giá trị của showValue trong 8 vị trí
            Console.Write("{0,8}", showValue);

            // tăng giá trị của biến countLoop lên một đơn vị
            countLoop++;
        } while (countLoop < valueNum);

        // Mỗi khi hoàn thành một vòng lặp nhỏ thì lại xuống dòng chuẩn vị vẽ lần tiếp theo
        Console.WriteLine();

        // tăng giá trị countLoopTime lên một đơn vị
        countLoopTime++;
    } while (countLoopTime < loopTime);
    Console.ReadKey();
}

```

- Kết quả một ma trận số hoàn toàn ngẫu nhiên được xuất ra màn hình

```

file:///D:/Project/Web/Recuirement Web HowkTeam/Git/How/Tutorials/C#/14.While/Project/While/While...
11    94    8    19    4    41    77    11    12    75
59    77    12    30    20    15    32    56    11    40
28    32    78    55    29    30    38    52    29    18
78    31    21    21    81    74    16    82    62    30
77    71    87    29    73    89    82    3    34    81

```

- Chúng ta cũng có thể vẽ một hình chữ nhật rỗng NxM với vòng lặp `do while`.

:

```

static void Main(string[] args)
{
    int Vertical = 20;
    int Horizontal = 50;
    char drawChar = '*';
    char insideChar = ' ';
    int countLoopVertical = 0;
    int countLoopHorizontal = 0;

    // Vẽ từ trên xuống
    do
    {
        // khởi tạo lại giá trị countLoopHorizontal = 0 mỗi lần lặp mới
        countLoopHorizontal = 0;

        // Vẽ từ trái sang
        do
        {
            /*
             * Nếu đang ở tọa độ là cạnh trên hoặc dưới i % (N - 1) == 0
             * hoặc đang ở cạnh trái hoặc phải (j % (M - 1) == 0)
             * mà không nằm ở cạnh trên hoặc dưới (i % (N - 1) != 0)
             * ((i % (N - 1) != 0) && (j % (M - 1) == 0))
             * thì vẽ ra ký tự của hình chữ nhật
             * ngược lại vẽ ra ký tự không thuộc hình chữ nhật
             */

            if (countLoopVertical % (Vertical - 1) == 0 || ((countLoopVertical % (Vertical - 1) != 0) &&
(countLoopHorizontal % (Horizontal - 1) == 0)))
            {
                Console.Write(drawChar);    // lúc này là ký tự *
            }
            else
            {
                Console.Write(insideChar); // lúc này là ký tự rỗng ' '
            }

            // tăng giá trị của countLoopHorizontal lên 1 đơn vị
            countLoopHorizontal++;
        } while (countLoopHorizontal < Horizontal);

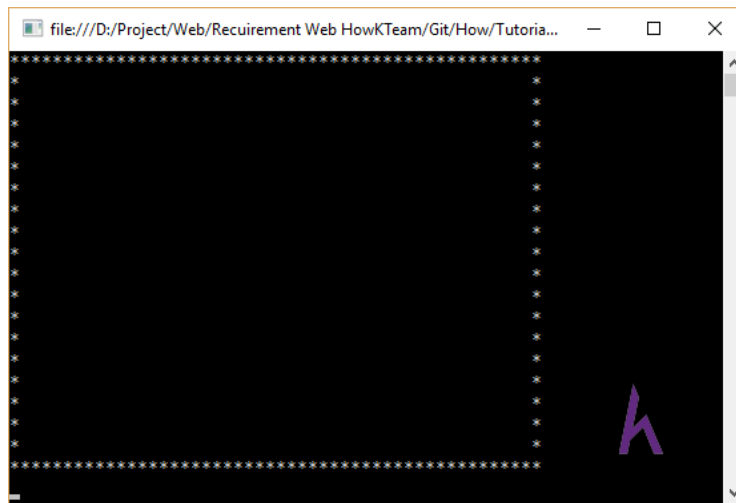
        // mỗi lần vẽ xong một hàng thì xuống dòng
        Console.WriteLine();

        // tăng giá trị của countLoopVertical lên 1 đơn vị
        countLoopVertical++;
    } while (countLoopVertical < Vertical);

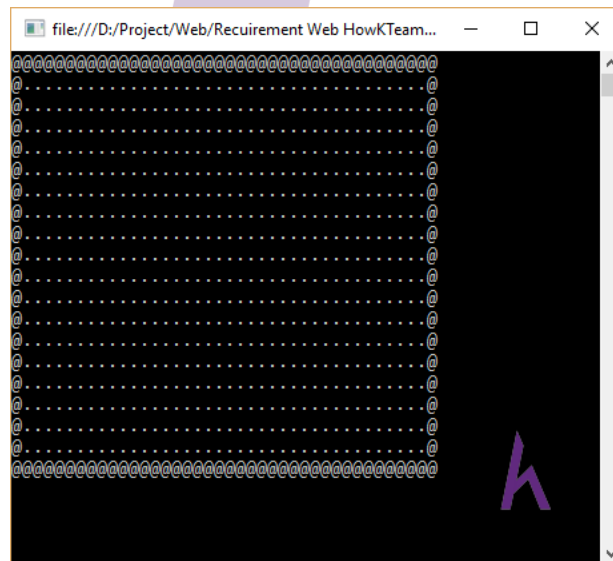
    Console.ReadKey();
}

```

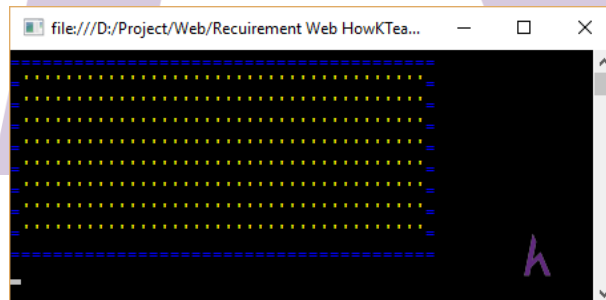
- Kết quả sẽ thấy một hình chữ nhật rỗng chiều ngang 20 chiều dài 50 được vẽ lên màn hình.



- Ta có thể thay đổi giá trị của M, N, drawChar và insideChar để có những hình chữ nhật màu mè cùng kích thước khác nhau.



- Bạn hãy thử thay đổi màu sắc hoặc vẽ hình chữ nhật ở bất cứ đâu trên màn hình console nhé. Thử vẽ hình chữ nhật đặc hay hình tam giác, hình tròn cũng là một thử thách thú vị đấy. Chúc các bạn thành công.



## Kết luận

Qua bài này chúng ta đã nắm được cách sử dụng vòng lặp `do while`. Một cấu trúc rất mạnh mẽ và tần xuất sử dụng cực kỳ nhiều trong lập trình. Những đặc điểm của vòng lặp `do while`. Cùng những điều cần lưu ý.

Bài sau chúng ta sẽ tìm hiểu qua một khái niệm mới – đó là [CẤU TRÚC HÀM TRONG C#.](#)

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên “**Luyện tập – Thử thách – Không ngại khó**”.

