زنگ سی‌شارپ – قسمت یازدهم

نوشته‌ی مسعود درویشیان

لینک مستقیم این مطلب در وب‌تارگت

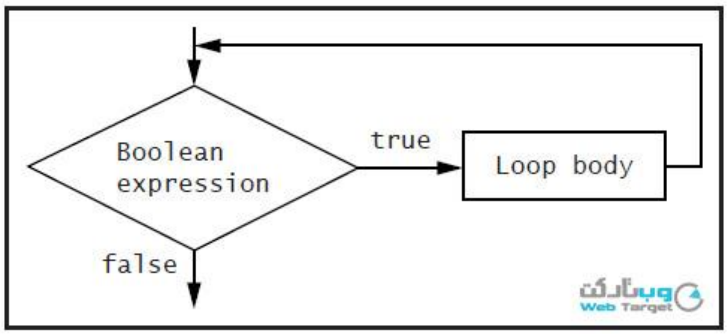
در قسمت قبل در مورد حلقه‌ی for مثال بیشتری زده شد و تمرین‌های ۱ تا ۴ حل شدند. در این قسمت به چندین مثال تکمیلی می‌پردازیم و در نهایت تمرین شماره ۵ حل خواهد شد.

همان‌طور که پیش‌از این در مورد حلقه‌ی while توضیح داده شد، ساختار این حلقه به صورت زیر است:

```
while (condition) statement;
```

که statement می‌تواند تنها یا بلاکی از statement ها باشد و condition یک عبارت شرطی است که شرط حلقه را چک می‌کند و می‌تواند هر نوع عبارت بولینی باشد. همین‌طور دانستید تا زمانی‌که condition برابر با true است statement اجرا می‌شود. در ادامه برای فهم بهتر مطالب از الگوریتم و فلوچارت نیز بهره می‌بریم، برای یادگیری الگوریتم و فلوچارت پیشنهاد می‌شود این مطلب کوتاه و مفید را که نوشته‌ی امید شمس دیوبند است مطالعه بفرمایید.

به فلوچارت زیر دقت کنید:



همان‌طور که پیداست، شرط بررسی شده و در صورت true بودن بدنه‌ی حلقه اجرا می‌شود.

به نمونه‌ی زیر دقت کنید:

```
static void Main()
{
    int number = 1;

    while (number > 0)
        Console.WriteLine("Hello");
}
```

در این‌جا، شرط number > 0 بررسی می‌شود که برابر با true است، بنابراین عبارت Hello نمایش داده می‌شود. دوباره شرط number > 0 بررسی می‌شود و مجدداً Hello چاپ می‌شود. به‌دلیل این‌که هیچ‌چیزی متغیر number را تغییر نمی‌دهد، این حلقه تا بی‌نهایت اجرا خواهد شد و پیوسته شرط را چک می‌کند (که همیشه برابر با true است) و Hello را نمایش می‌دهد. برای این‌که یک حلقه‌ی while به‌درستی پایان یابد، باید ۳ کار به‌صورت جداگانه انجام شود:

- قبل از ورود به حلقه باید یک متغیر که کنترل‌کننده‌ی حلقه است معرفی شود.
- متغیر کنترل‌کننده‌ی حلقه باید در قسمت condition مورد بررسی قرار گیرد.
- در بدنه‌ی حلقه باید عملی انجام شود که متغیر کنترل‌کننده‌ی حلقه تغییر یابد (تا در نهایت مقدار condition برابر با false شود).

به مثال زیر دقت کنید:

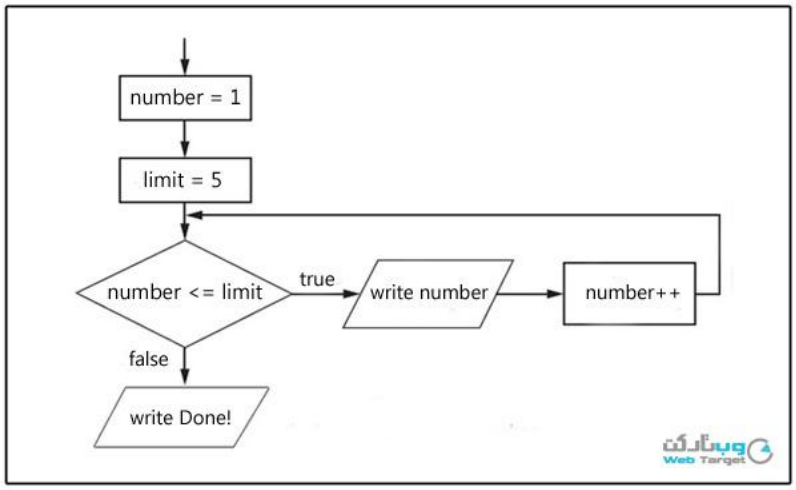
```
using System;
class Example
{
    static void Main()
    {
        int number = 1;
        int limit = 5;

        while (number <= limit)
        {
            Console.WriteLine("Number is " + number);
            number++;
        }

        Console.WriteLine("Done!");
    }
}
```

همان‌طور که می‌بینید ابتدا متغیر کنترل‌کننده‌ی حلقه تعریف شده، سپس در قسمت condition بررسی می‌شود و در نهایت در بدنه‌ی حلقه تغییر می‌یابد.

فلوچارت این برنامه:



حلقه‌ی do-while

یکی دیگر از حلقه‌های سی‌شارپ، حلقه‌ی do-while است. برخلاف حلقه‌ی for و while که شرط در بالای حلقه بررسی می‌شد، در do-while شرط در پایین حلقه بررسی می‌شود و به‌همین دلیل، این حلقه همیشه حداقل یک‌بار اجرا می‌شود. فرم کلی آن به شکل زیر است:

```
do {
    statements;
} while(condition);
```

در این حلقه نیز تا زمانی‌که condition برابر با true است، statement اجرا می‌شود. البته قبل از این‌که برنامه به condition برسد statement یک‌بار اجرا می‌شود.

به مثال زیر توجه کنید:

```
using System;
class Example
{
    static void Main()
    {
        int number = 1;
        int limit = 5;

        do
        {
            Console.WriteLine("Number is " + number);
            number++;
        } while (number <= limit);
    }
}
```

خروجی و عمل‌کرد این برنامه همانند مثال قبلی در حلقه‌ی while است با این تفاوت که شرط در پایین حلقه چک می‌شود.

به این مثال دقت کنید:

```
using System;
class Example
{
    static void Main()
    {
        int number = 3;

        do
        {
            Console.WriteLine("False condition, once execute!");
        } while (number < 2);
    }
}
```

اگر این برنامه را اجرا کنید متوجه می‌شوید با این‌که شرط برقرار نیست ولی پیغام نمایش داده می‌شود. علت این است که شرط در پایین حلقه بررسی می‌شود و statement حداقل یک‌بار اجرا خواهد شد.

حل تمرین شماره ۵:

```
using System;
class Example
{
    static void Main()
    {
        /* Solution 05 */

        Console.WriteLine("Even numbers from 1 to 100:");
        int i = 1;
        while (i <= 100)
        {
            if (i % 2 == 0)
            {
                Console.Write(i + "\t");
            }
            i++;
        }
        Console.WriteLine();
        Console.WriteLine();

        Console.WriteLine("Counting backwards from 1 to 100:");
        int j = 100;
        while (j >= 1)
        {
            Console.Write(j + "\t");
            j--;
        }
        Console.WriteLine();
        Console.WriteLine();
```

```
        Console.WriteLine("Odd numbers from 1 to 100:");
        int h = 1;
        while (h <= 100)
        {
            if (h % 2 != 0)
            {
                Console.Write(h + "\t");
            }
            h++;
        }
        Console.WriteLine();
        Console.WriteLine();

        Console.WriteLine("Multiple of 5 from 1 to 100");
        int k = 1;
        while (k <= 100)
        {
            if (k % 5 == 0)
            {
                Console.Write(k + "\t");
            }
            k++;
        }
        Console.WriteLine();
    }
}
```

در قسمت بعد به مثال‌های بیشتر و توضیحات تکمیلی‌تر می‌پردازیم.

کلیه حقوق مادی و معنوی برای وب‌سایت وب‌تارگت محفوظ است.
استفاده از این مطلب در سایر وب‌سایت‌ها و نشریات چاپی تنها با ذکر و درج لینک منبع مجاز است.