زنگ سی‌شارپ – قسمت چهاردهم

نوشته‌ی مسعود درویشیان

لینک مستقیم این مطلب در وب‌تارگت

در قسمت سیزدهم با دستور goto و توضیحات تکمیلی دستور switch آشنا شدید. اگر به‌یاد داشته باشید در قسمت نهم با حلقه‌ی for آشنا شدیم. تا به این‌جا ما تنها با یک حلقه‌ی for و در یک سطح کار می‌کردیم ولی اکنون قصد داریم با حلقه‌های for تودرتو آشنا شویم.

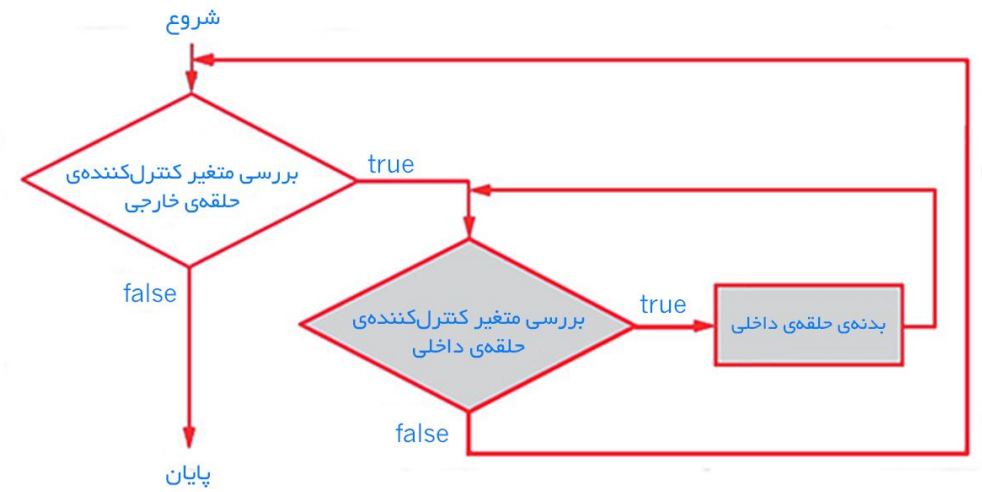
به مثال زیر و خروجی آن دقت کنید تا در ادامه به شرح و تفصیل آن بپردازیم:

```
using System;
class Example
{
    static void Main()
    {
        for (int i = 1; i < 10; i++)
        {
            for (int j = 1; j <= i; j++)
            {
                Console.Write(j);
            }
            Console.WriteLine();
        }
    }
}
```

خروجی:

```
1
12
123
1234
12345
123456
1234567
12345678
123456789
```

شاید اگر با دقت تمام خودتان این برنامه را تجزیه و تحلیل کنید به ماهیت و چگونگی آن پی ببرید اما بهتر است توضیحی کامل در مورد این برنامه داده شود. هنگامی‌که حلقه‌ها تودرتو می‌شوند شامل یک حلقه‌ی داخلی و یک حلقه‌ی خارجی هستند که حلقه‌ی داخلی کاملاً در حلقه‌ی خارجی قرار دارد. اگر به فلوچارت زیر دقت کنید کاملاً متوجه خواهید شد که چه اتفاقی در حلقه‌های تودرتو می‌افتد:



در فلوچارت بالا دیاگرام خاکستری رنگ، حلقه‌ی داخلی (inner loop) است که درون حلقه‌ی خارجی (outer loop) قرار دارد. همان‌طور که می‌بینید ابتدا شرط حلقه‌ی خارجی بررسی شده و در صورت true بودن وارد حلقه‌ی داخلی می‌شود (درواقع تمام حلقه‌ی داخلی، بدنه‌ی حلقه‌ی خارجی است). اکنون شرط حلقه‌ی داخلی بررسی شده و تا زمانی‌که شرط true است بدنه‌ی حلقه‌ی داخلی اجرا می‌شود. به‌محض این‌که شرط حلقه‌ی داخلی false شود کامپایلر به ابتدای حلقه‌ی خارجی برمیگردد و متغیر کنترل‌کننده‌ی حلقه‌ی خارجی را به‌روزرسانی کرده و مجدداً شرط حلقه‌ی خارجی را بررسی می‌کند که در صورت true بودن دوباره حلقه‌ی داخلی اجرا شده و در صورت false بودن اجرای حلقه‌های تودرتو پایان می‌یابد.

در مثال ابتدای مقاله، در حلقه‌ی for خارجی، متغیر i شامل مقدار ۱ بوده که کمتر از ۱۰ است بنابراین شرط برقرار بوده و کامپایلر وارد حلقه می‌شود. درون این حلقه (حلقه‌ی خارجی)، حلقه‌ی دیگری قرار دارد (حلقه‌ی داخلی) که شامل متغیر j با مقدار ۱ است. در قسمت condition حلقه‌ی داخلی بررسی می‌شود که مقدار j کوچک‌تر یا مساوی مقدار i باشد. در این‌جا مقدار j و i هردو برابر با ۱ هستند بنابراین شرط برقرار است و کامپایلر وارد حلقه‌ی داخلی می‌شود. درون حلقه‌ی داخلی، مقدار j چاپ شده سپس متغیر کنترل‌کننده‌ی حلقه‌ی داخلی به‌روز رسانی می‌شود (j++) و شرط حلقه‌ی داخلی بررسی خواهد شد. همان‌طور که می‌دانید اکنون مقدار j برابر با ۲ و مقدار i برابر با ۱ است. بنابراین دیگر شرط برقرار نبوده و برنامه از حلقه‌ی داخلی خارج می‌شود.

در خط بعد، دستور Console.WriteLine() خروجی برنامه را به خط جدید سوق می‌دهد.

خروجی برنامه تا این‌جا (مرحله‌ی اول):

```
1
```

اکنون از بلاک حلقه‌ی داخلی خارج شدیم ولی هنوز در بلاک حلقه‌ی خارجی قرار داریم. پس متغیر کنترل‌کننده‌ی حلقه‌ی خارجی به‌روز رسانی شده و شرط حلقه‌ی خارجی ارزیابی می‌شود. پس از برقراری شرط، برنامه مجدداً وارد حلقه شده و شرط حلقه‌ی داخلی را ارزیابی می‌کند. اکنون مقدار **j** برابر با ۱ و مقدار **i** نیز برابر با ۲ است (متغیر **j** روی حلقه‌ی داخلی مجدداً برابر با ۱ شده و هربار که برنامه می‌خواهد از نو وارد حلقه شود متغیر کنترل‌کننده‌ی آن به اصطلاح ریست - Reset می‌شود). همان‌طور که می‌بینید شرط برقرار است و برنامه وارد بدنه‌ی حلقه‌ی داخلی می‌شود و مقدار **j** را در خروجی چاپ می‌کند. سپس متغیر کنترل‌کننده‌ی حلقه‌ی داخلی به‌روز رسانی شده (برابر با ۲ می شود) و دوباره شرط بررسی می‌شود. اکنون **i** و **j** هردو برابر با ۲ هستند پس شرط برقرار است و یک‌بار دیگر مقدار **j** چاپ خواهد شد.

خروجی برنامه در مرحله‌ی دوم:

```
1
12
```

خروجی برنامه در مرحله‌ی سوم:

```
1
12
123
```

خروجی برنامه در مرحله‌ی چهارم:

```
1
12
123
1234
```

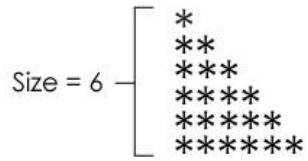
این روند همین‌طور ادامه دارد تا زمانی‌که برنامه به مرحله‌ی نهم برسد و شرط حلقه‌ی خارجی نقض شود:

```
1
12
123
1234
12345
123456
1234567
12345678
123456789
```

تمرین

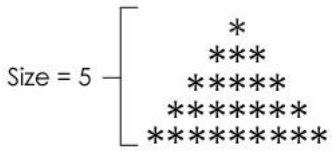
تمرین شماره ۶: برنامه‌ای بنویسید که شکل مثلث قائم‌الزاویه را توسط کاراکتر * چاپ کند (سایز مثلث را از کاربر بگیرید).

نمونه:



تمرین شماره ۷: برنامه‌ای بنویسید که شکل مثلث متساوی‌الساقین را توسط کاراکتر * چاپ کند (سایز مثلث را از کاربر بگیرید).

نمونه:



تمرین شماره ۸: برنامه‌ای بنویسید که دو عدد (پایه و توان) را از کاربر گرفته و عدد اول را به توان عدد دوم برساند.

تمرین شماره ۹: توسط دستور `switch` یک فرهنگ لغت کوچک بسازید.

تمرین شماره ۱۰: توسط کاراکتر ▉ یک مستطیل طراحی کنید (سایز مستطیل را از کاربر بگیرید).

تمرین شماره ۱۱: برنامه‌ی جدول ضربی بنویسید که طول و عرض جدول را از کاربر بگیرد و جدول را نمایش دهد.

بدیهی است که تمرین‌های ۶، ۷، ۱۰ و ۱۱ باید توسط حلقه‌های تودرتو انجام شوند.

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.