زنگ سی‌شارپ – قسمت دوازدهم

نوشته‌ی مسعود درویشیان

لینک مستقیم این مطلب در وب‌تارگت

در قسمت یازدهم با مثال‌های بیشتری در مورد حلقه‌ی while آشنا شدید همچنین حلقه‌ی do-while توضیح داده شد. در این قسمت با دستور switch و چند نکته‌ی دیگر در مورد حلقه‌ها (break و continue از بخش Jump Statements) آشنا می‌شویم.

استفاده از break برای خارج شدن از حلقه

شما می‌توانید با استفاده از کلمه‌ی break بلافاصله از حلقه خارج شوید. وقتی از کلمه‌ی break برای خارج شدن از حلقه استفاده می‌کنید دیگر مهم نیست که چقدر از کدها در حلقه باقی مانده‌اند حتی با این‌که ممکن است شرط حلقه هنوز برقرارباشد، برنامه از حلقه خارج می‌شود و به خواندن ادامه کدهای بعد از حلقه می‌پردازد.

به مثال زیر توجه کنید:

```
using System;
class Example
{
    static void Main()
    {
        // Use break to exit this loop.
        for (int i = -10; i <= 10; i++)
        {
            if (i > 0) break; // terminate loop when i is positive
            Console.Write(i + " ");
        }
        Console.WriteLine("Done");
    }
}
```

خروجی:

```
-10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 Done
```

همان‌طور که مشاهده می‌کنید، حلقه طوری طراحی شده که باید از ۱۰- تا ۱۰ را نمایش دهد ولی با استفاده از کلمه break، حلقه در زمانی که i مثبت می‌شود به پایان می‌رسد.

مثالی دیگر از break:

```
using System;
class Example
```

```
{
    static void Main()
    {
        int x = 0;
        while (true)
        {
            if (x++ > 5)
                break; // break from the loop
        }
        // execution continues here after break...
    }
}
```

نکته‌ی قابل توجه در این برنامه، حلقه‌ی while(true) است. در این حلقه پیوسته شرط حلقه (Condition) برقرار است (برابر با true است) و این حلقه تا بی‌نهایت اجرا می‌شود ولی با توجه به برنامه ما، دستور if درون حلقه مانع از اجرای بی‌نهایت حلقه شده و پس از این‌که x++ بزرگ‌تر از ۵ بود برنامه از حلقه خارج می‌شود.

بنابراین حلقه‌ی بی‌نهایت while این‌گونه است:

```
while (true)
{
    // This is an infinite while loop
}
```

در حلقه‌ی for تمام قسمت‌ها اختیاری هستند و شما می‌توانید با خالی گذاشتن آن قسمت‌ها یک حلقه‌ی بی‌نهایت به‌وجود آورید (توجه داشته باشید که گذاشتن سمی‌کالن‌ها در حلقه‌ی for اجباری است):

```
for (; ; )
{
    // this is an infinite for loop
}
```

در واقع حلقه‌های بی‌نهایت تا بی‌نهایت اجرا نمی‌شوند چراکه تمام برنامه‌ها با کمک سخت‌افزار و حافظه‌ی کامپیوتری اجرا می‌شوند که هر دوی آن‌ها ظرفیت محدودی دارند. اگر یک حلقه‌ی بی‌نهایت را اجرا کردید برای متوقف کردن آن کافی است کلید Ctrl و C را هم‌زمان بگیرید.

استفاده از continue

شما می‌توانید در بدنه‌ی حلقه از کلمه‌کلیدی continue استفاده کنید. با این‌کار حلقه وادار می‌شود عملیات تکرار بعدی را انجام دهد و مابقی کد را نادیده بگیرد.

به این مثال توجه کنید:

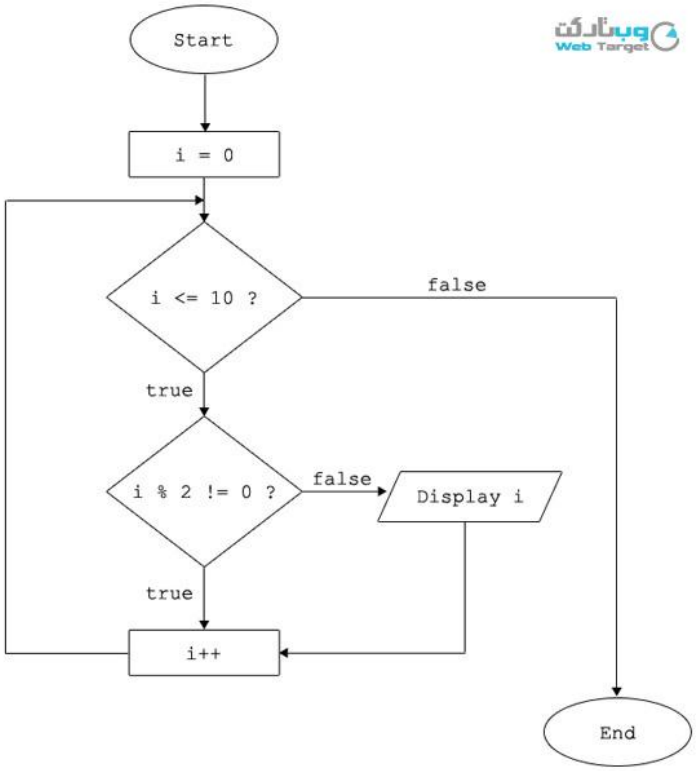
```
using System;
class Example
{
    static void Main()
    {
        for (int i = 0; i <= 10; i++)
        {
            if (i % 2 != 0)
                continue;
            Console.Write(i + " ");
        }
    }
}
```

خروجی:

۰ ۲ ۴ ۶ ۸ ۱۰

همان‌طور که می‌بینید این برنامه اعداد زوج ۰ تا ۱۰ نمایش می‌دهد. در این مثال هنگامی‌که i عددی فرد است، برنامه توسط کلمه continue به تکرار بعدی حلقه می‌پردازد و مقدار i را نمایش نمی‌دهد.

به فلوچارت برنامه بالا دقت کنید:



همان‌طور که در فلوچارت و خط‌کد برنامه مشخص است، اگر شرط if درون حلقه برقرار باشد، دستور continue اجرا شده و برنامه به ابتدای حلقه باز می‌گردد. در حلقه‌های while و do-while کلمه‌ی continue موجب می‌شود برنامه مستقیماً به قسمت condition حلقه برود و مجدداً حلقه را اجرا کند.

The switch Statement

اگر به‌خاطر داشته باشید بیان شد که سه دسته‌بندی برای Program Control Statement وجود دارد. در دسته‌بندی Selection Statements دستورهای if و switch بودند. همچنین دانستید که توسط روش نردبانی (The if-else-if Ladder) می‌توانید چنین شرط را چک کنید. برای مثال، تصور کنید که می‌خواهید با توجه به مقدار سال تحصیلی، اسم یک دانش‌آموز را نمایش دهید. توسط روش نردبانی از این روش استفاده می‌کردیم:

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

        if (year == 1)
            Console.WriteLine("Freshman");
        else if (year == 2)
            Console.WriteLine("Sophomore");
        else if (year == 3)
            Console.WriteLine("Junior");
        else if (year == 4)
            Console.WriteLine("Senior");
        else
            Console.WriteLine("Invalid year");
    }
}
```

این مثال را نیز از طریق if های تودرتو (Nested ifs) می‌توانید انجام دهید. راه دیگر برای حل این‌گونه مسائل استفاده از دستور switch است. دستور switch یک متغیر را با چندین مورد مقایسه می‌کند و آن مورد را که با متغیر مطابقت دارد، انتخاب می‌کند. ساختار دستور switch از if های تودرتو و روش نردبانی ساده‌تر و خواناتر است. در دو روش (تودرتو و نردبانی) قبلی هنگامی‌که چندین شرط را پی‌درپی بررسی می‌کنیم، خواندن کدها سخت‌تر شده و امکان به‌وجود آمدن خطا بیشتر می‌شود در حالی‌که ساختار switch در این موارد ساده و خواناتر است. ساختار کلی دستور switch به شکل زیر است:

```
switch(expression)
{
    case constant1:
        statement sequence
        break;
```

```
    case constant2:
        statement sequence
        break;
    case constant3:
        statement sequence
        break;
.
.
.
    default:
        statement sequence
        break;
}
```

قسمت expression باید یا integer type (مثل char, byte, short, int) یا string type باشد. بنابراین در این‌جا عبارات floating-point برای استفاده مجاز نیستند. غالباً expression، کنترل‌کننده‌ی switch، یک متغیر است. بعد از کلمه‌کلیدی case یک مقدار ثابت (constant) قرار می‌گیرد که حتماً باید هم‌نوع expression باشد. هیچ دو case ای در یک switch نباید constant یکسان داشته باشند. کلمه‌کلیدی break در پایان هر case قرار دارد و برای پایان دادن به ساختار switch استفاده می‌شود. استفاده از کلمه‌کلیدی default اختیاری است. اگر مطابقت با expression در case ها یافت شد، کدهای قسمت همان case تا رسیدن به break اجرا می‌شود ولی اگر مقدار هیچ case ای با expression مطابقت نداشت، قسمت default اجرا می‌شود. اگر از default استفاده نکردید، هنگامی‌که هیچ تطابقی یافت نشد، هیچ اتفاقی نیز نخواهد افتاد.

به مثال زیر دقت کنید:

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

        switch (year)
        {
            case 1:
                Console.WriteLine("Freshman");
                break;
            case 2:
                Console.WriteLine("Sophomore");
                break;
            case 3:
                Console.WriteLine("Junior");
                break;
            case 4:
                Console.WriteLine("Senior");
                break;
            default:
                Console.WriteLine("Invalid year");
                break;
```

```
        }
    }
}
```

همان‌طور که می‌بینید، مقدار متغیر year با case های مختلف مقایسه می‌شود و در صورت تطابق، case مربوطه اجرا می‌شود.

برای توضیحات تکمیلی، مثال‌های بیشتر و تمرین قسمت‌های بعدی زنگ سی‌شارپ را دنبال کنید.

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.