زنگ سی‌شارپ – قسمت هفتم

نوشته‌ی مسعود درویشیان

لینک مستقیم این مطلب در وب‌تارگت

در قسمت قبل بحث در مورد Program Control Statements بود و توضیحاتی در مورد این سرفصل و دستور if داده شد. در این قسمت به توضیحات تکمیلی‌تر این مبحث می‌پردازیم.

همان‌طور که در قسمت ششم بیان شد، دستور if و if-else یکی از رایج‌ترین راه‌ها برای تصمیم‌گیری در سی‌شارپ هستند. شما دستور if را برای یک تصمیم‌گیری به‌کار می‌برید. به‌عبارت دیگر، شما از دستور if برای این‌که مشخص کنید یک عمل اتفاق بیفتد یا خیر، استفاده می‌کنید. هم‌چنین دانستید که در قسمت condition دستور if یک مقدار بولین قرار دارد که اگر true باشد statement اجرا می‌شود و اگر false باشد statement اجرا نخواهد شد. در بعضی از زبان‌های برنامه‌نویسی مثل C++ اعداد غیرصفر به‌عنوان true تلقی می‌شوند و عدد صفر بیان‌کننده‌ی مقدار false است. در سی‌شارپ تنها عبارت‌های بولی و مقایسه‌هایی که مقدار بولین را برمی‌گردانند برای نشان دادن مقدار true و false استفاده می‌شوند.

همان‌طور که در قسمت قبل گفته شد، هنگامی‌که چندین statement داشته باشید، باید آن‌ها را در یک بلوک قرار دهید. حتی اگر تنها یک statement هم داشته باشید می‌توانید آن‌را در یک بلوک قرار دهید. قرار دادن یک statement در بلوک، یک تکنیک مفید است که باعث جلوگیری از خطاهای احتمالی در آینده می‌شود زیرا اگر بخواهید بعداً در یک برنامه تغییراتی را اعمال کنید و برای مثال چندین statement را در یک دستور if اضافه کنید، این احتمال خیلی زیاد است که از براکت باز و بسته غافل شوید.

همچنین می‌توانید یک بلوک خالی داشته باشید که هیچ statement ای در آن نیست. این‌کار معمولاً زمانی‌که یک برنامه‌نویس شروع به نوشتن یک برنامه می‌کند و برای این‌که به خودش یادآوری کند بعداً statement را در بلوک قرار دهد، انجام می‌شود. بعضی از برنامه‌نویسان ترجیح می‌دهند براکت باز را برای بلوک‌ها در همان خطی قرار دهند که عبارت if هم قرار دارد (به‌جای این‌که براکت باز در یک خط جدا باشد). این سبک بلوک‌گذاری K&R نام دارد و این نام‌گذاری به احترام Brian Kernighan و Dennis Ritchie است که مخترعان زبان C هستند و در اولین کتاب زبان برنامه‌نویسی C از این روش استفاده کرده بودند.

مثالی به سبک K&R:

```
static void Main()
```

```
{
    int i = 6;
    if (i < 8) {
        Console.WriteLine("A sample of K&R style...");
    }
    else {
        Console.WriteLine("A sample of K&R style...");
    }
}
```

Nested ifs (if های تودرتو)

if تودرتو، if ای است که در قسمت statement یک if یا else دیگر قرار دارد.

به نمونه‌ی زیر دقت کنید:

```
using System;
class NestedIf
{
    static void Main()
    {
        int i = 6;

        if (i < 8)
        {
            if (i == 6)
            {
                Console.WriteLine("This is a nested if.");
            }
        }
    }
}
```

همان‌طور که می‌بینید، یک if در درون if دیگری واقع شده است. اگر if اول شامل مقدار true باشد، کامپایلر وارد بلوک آن شده که با if دیگری مواجه می‌شود. اگر if دوم هم شامل مقدار true باشد، استرینگ مربوطه نمایش داده می‌شود. در غیر این‌صورت هیچ خروجی چاپ نمی‌شود.

نکته‌ی اصلی که در مورد Nested ifs باید به‌خاطر بسپارید این است که، عبارت else همیشه مربوط به نزدیک‌ترین if بالای خودش است که هردوشان در یک بلوک هستند.

به نمونه‌ی زیر دقت کنید:

```
if (i == 10)
{
```

```
    if (j < 20) a = b;
    if (k > 100) c = d;
    else a = c; // this else refers to if(k > 100)
}
else a = d; // this else refers to if(i == 10)
```

در مثال بالا، آخرین else مربوط به if(j < 20) نیست به‌دلیل این‌که این‌ها در یک بلوک یکسان نیستند (هرچند که این نزدیک‌ترین if بدون else است). بلکه آخرین else مربوط به if(i == 10) است. else داخلی مربوط به if(k > 100) است زیرا این نزدیک‌ترین if به آن است و هردوی آن‌ها در یک بلوک هستند.

The if-else-if Ladder

اگر شما به یک مقایسه‌ی پیچیده‌تر نیاز داشته باشید، می‌توانید چندین شرط را چک کنید. این روش که به روش نردبانی معروف است، یکی از رایج‌ترین روش‌ها در برنامه‌نویسی است و فرم کلی آن به شکل زیر است:

```
if(condition)
  statement;
else if(condition)
  statement;
else if(condition)
  statement;
.
.
.
else
  statement;
```

در این مدل، condition ها از بالا به پایین چک می‌شوند و به‌محض این‌که یک condition با مقدار true پیدا شد، statement مربوط به آن اجرا می‌شود و بقیه‌ی این ساختار نردبانی نادیده گرفته می‌شود. اگر هیچ‌یک از condition ها درست نبود، عبارت else پایانی اجرا می‌شود. else پایانی غالباً به‌عنوان condition پیش‌فرض عمل می‌کند و هرگاه که همه‌ی condition ها شامل مقدار false باشند، else پایانی اجرا می‌شود. اگر else پایانی را قرار ندهید و همه‌ی condition ها مقدار false داشته باشند، آن‌گاه هیچ اتفاقی نمی‌افتد.

به مثال جالب زیر توجه کنید:

```
//if-else-if structure
using System;
class ifElseIfLadder
{
    static void Main()
    {
        string fullname = "Webtarget";
```

```
        if (fullname == "Bill Gates")
        {
            Console.WriteLine("C# is pretty cool");
        }
        else if (fullname == "James Gosling")
        {
            Console.WriteLine("Java is pretty cool");
        }
        else
        {
            Console.WriteLine("Webtarget.ir is pretty cool!");
        }
        //end if-else-if structure
    }
}
```

خروجی:

```
Webtarget.Ir is pretty cool!
```

همان‌طور که می‌بینید، وب‌تارگت انتخاب شد زیرا Bill Gates و James Gosling شرایط لازم را نداشتند!

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.