زنگ سی‌شارپ – قسمت پانزدهم

نوشته‌ی مسعود درویشیان

لینک مستقیم این مطلب در وب‌تارگت

در این قسمت به حل تمرین‌های ۶ تا ۱۱ می‌پردازیم. هدف از انجام این تمرین‌ها آشنایی بیشتر با حلقه های تو در تو و دستور `switch` است.

حل تمرین شماره ۶: همان‌طور که می‌دانید این تمرین شباهت زیادی با مثال قسمت قبل دارد با این تفاوت که اندازه مثلث از کاربر گرفته می‌شود:

```
using System;
class Solution
{
    static void Main()
    {
        // Solution 06

        Console.Write("Please enter triangle size: ");
        int size = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine();

        for (int i = 0; i < size; i++)
        {
            for (int j = 0; j <= i; j++)
            {
                Console.Write("*");
            }
            Console.WriteLine();
        }

        Console.WriteLine();
    }
}
```

نحوه‌ی دریافت ورودی از کاربر در قسمت چهارم مورد بحث قرار گرفت. تفاوت دریافت ورودی در این تمرین و مثال قسمت چهارم در نوع ورودی آن‌ها است. در مثال قسمت چهارم، ورودی گرفته شده از کاربر از جنس string بود اما در اینجا ورودی دریافت شده باید یک عدد باشد. هنگامی که ورودی را به برنامه می‌دهید، مقدار وارد شده از جنس string است و شما باید آن را به عدد تبدیل کنید. همان‌طور که می‌بینید، مقدار دریافت شده از کاربر توسط دستور Convert.ToInt32() به یک عدد صحیح 32 بیتی علامت‌دار (شامل مقدار مثبت و منفی) تبدیل شده و مستقیماً در متغیر size قرار گرفته است. Convert.ToInt32() متدی است که مقدار string را دریافت کرده و به یک عدد صحیح 32 بیتی

علامت‌دار (32-bit signed integer) تبدیل می‌کند. در ادامه‌ی تمرین بالا، مانند مثال قسمت قبل حلقه‌های تو در تو به جای نمایش اعداد کاراکتر ستاره را نمایش می‌دهند. در قسمت‌های آینده مبحث متدها و کلاس‌ها بطور کامل و دقیق مورد بررسی قرار می‌گیرد پس در این مورد هیچ جای نگرانی نیست که شما کاملاً با شی گرایی آشنا شده و بر آن مسلط می‌شوید.

حل تمرین شماره ۷: با این تمرین راحت کنار آمدید؟ این مسئله بسیار ساده است. کافی است همه مراحل انجام کار را به‌روی کاغذ بیاورید تا به روش حل آن پی ببرید. در ابتدا اجازه دهید جواب را با هم ببینیم و سپس به توضیح و شرح آن بپردازیم.

```
// Solution 07
using System;
class Solution
{
    static void Main()
    {
        // Getting input from the user
        Console.Write("Please enter the size: ");
        int size = Convert.ToInt32(Console.ReadLine());

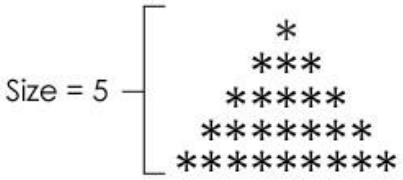
        // Variables declaration
        int rowNumber, j, k;

        // 1st loop that's counting the rows
        for (rowNumber = 1; rowNumber <= size; rowNumber++)
        {
            // 2nd loop that write space character
            for (j = 1; j <= size - rowNumber; j++)
            {
                Console.Write(" ");
            }

            // 3rd loop that write star character
            for (k = 1; k <= (2 * rowNumber) - 1; k++)
            {
                Console.Write("*");
            }

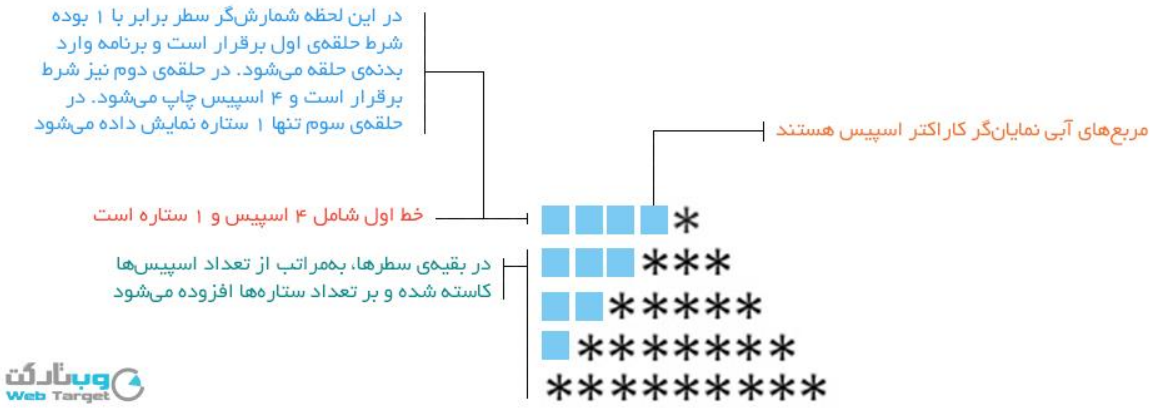
            // finally, this WriteLine() method makes a break.
            Console.WriteLine();
        }
    }
}
```

همان‌طور که می‌دانید خروجی برنامه این چنین است:



در ابتدا سایز مثلث از کاربر گرفته می‌شود. در واقع سایز مثلث همان تعداد سطر برنامه‌ی ما است و با اینکار ما به‌نحوی تعداد سطری که مثلث قرار است داشته باشد را از کاربر دریافت کردیم. این برنامه شامل یک حلقه‌ی خارجی و دو حلقه‌ی داخلی است. حلقه‌ی خارجی که اولین حلقه‌ی ما است این وظیفه را بر عهده دارد که تعداد سطرها را کنترل کند. دومین حلقه وظیفه‌ی چاپ کردن تعداد مشخصی کاراکتر space را در هر سطر بر عهده دارد. سومین حلقه نیز موظف است که در هر سطر تعداد مشخصی کاراکتر ستاره را چاپ کند.

به تصویر زیر دقت کنید:



چرا در قسمت شرطی حلقه ی اول نیاز داریم که متغیر rowNumber کوچک‌تر یا مساوی متغیر size باشد؟ جواب واضح است زیرا سایز مثلث همان تعداد سطر برنامه‌ی ما است. در حلقه ی دوم مشاهده می کنیم که متغیر j باید کوچک‌تر یا مساوی size – rowNumber باشد. علت این امر این است که در هر سطر باید تعدادی کاراکتر اسپیس چاپ شود و این کاراکترها باید با افزایش شماره هر سطر یک واحد کاهش یابند. در قسمت شرطی حلقه ی سوم مشاهده می کنید که متغیر k باید کوچک‌تر یا مساوی `(2 * rowNumber) - 1` باشد. اگر به تعداد ستاره‌ها در هر سطر توجه کنید متوجه می‌شوید که اعداد فرد را در بر می‌گیرند: 1،3، 5، 7 و... بنابراین باید در هر سطر معادل با اعداد فرد، ستاره‌ها نمایش داده شوند. در نهایت `Console.WriteLine()` برنامه را به خط بعد سوق می‌دهد.

حل تمرین شماره ۸: در این تمرین قصد داریم دو عدد را از کاربر دریافت کنیم و عدد اول را به توان عدد دوم برسانیم:

```
// Solution 08
using System;
class Example
{
    static void Main()
    {
        Console.Write("Enter base number: ");
        int baseNo = Convert.ToInt32(Console.ReadLine());
        Console.Write("Enter power number: ");
        int powerNo = Convert.ToInt32(Console.ReadLine());

        int result = 1;

        for (int counter = 1; counter <= powerNo; counter++)
            result = baseNo * result;

        Console.WriteLine();
        Console.WriteLine("The result is " + result);
        Console.WriteLine();
    }
}
```

در ابتدای این مقاله با نحوه ی دریافت ورودی از کاربر و تبدیل آن به عدد صحیح آشنا شدید. در اینجا ما علاوه بر متغیرهای پایه و توان به یک متغیر برای ذخیره ی نتیجه ی نهایی (result) و یک متغیر برای شمارنده ی حلقه (counter) نیازمندیم. حلقه تا زمانیکه کوچک‌تر یا مساوی عدد توان است تکرار می‌شود و عدد پایه را در مقدار result ضرب و حاصل آن را در خود متغیر result ذخیره می‌کند. به این ترتیب عدد پایه به اندازه‌ی عدد توان در خودش ضرب می‌شود.

حل تمرین شماره ۹: در این تمرین قصد داریم با استفاده از دستور switch یک فرهنگ لغت کوچک بسازیم:

```
// Solution 09
using System;
class Example
{
    static void Main()
    {
        Console.Write("Enter a word to translate: ");
        string word = Console.ReadLine();

        switch (word)
        {
            case "window":
                Console.WriteLine("Window means panjereh");
                break;
            case "library":
                Console.WriteLine("Library means ketab khaneh");
                break;
            case "easy":
                Console.WriteLine("Easy means asan");
                break;
            case "memory":
                Console.WriteLine("Memory means hafezeh");
                break;
```

```
            case "software":
                Console.WriteLine("Software means narm afzar");
                break;
            default:
                Console.WriteLine("Oops! Your word doesn't exist in our dictionary.");
                break;
        }
    }
}
```

در قسمت دوازدهم در مورد دستور switch بحث شد. برای ساخت فرهنگ لغت این روش کاملاً غیر استاندارد است پس هیچ‌گاه از این روش برای ساخت فرهنگ لغت استفاده نکنید. این تنها تمرینی برای آشنایی بیشتر با دستور switch بود.

حل تمرین شماره ۱۰: در این تمرین قصد داشتیم توسط حلقه های تو در تو و کاراکتر ▉ یک مستطیل را رسم کنیم:

```
// Solution 10
using System;
class Example
{
    static void Main()
    {
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 14; j++)
            {
                Console.Write("▉");
            }
            Console.WriteLine();
        }
    }
}
```

همان‌طور که می‌دانید حلقه ی اول مشخص می‌کند که عرض مستطیل شامل ۴ کاراکتر ▉ و حلقه ی دوم نشان می‌دهد که طول مستطیل شامل ۱۴ کاراکتر ▉ است و Console.WriteLine() باعث می‌شود برنامه بعد از چاپ هر سطر به خط بعد برود.

حل تمرین شماره ۱۱: ساختن جدول ضرب با طول و عرض مشخص

```
using System;
class Example
{
    static void Main()
    {
        Console.Write("Enter table width: ");
        int tableWidth = Convert.ToInt32(Console.ReadLine());
        Console.Write("Enter table height: ");
        int tableHeight = Convert.ToInt32(Console.ReadLine());
        int result;

        for (int i = 1; i <= tableHeight; i++)
```

```
        {
            result = 0;
            for (int j = 1; j <= tableWidth; j++)
            {
                result = i * j;
                Console.Write(result + "\t");
            }
            Console.WriteLine();
        }
    }
}
```

برای حل این تمرین ابتدا طول و عرض جدول ضرب را از کاربر دریافت کرده‌ایم سپس از طریق دو حلقه‌ی تو در تو متغیر i را در j ضرب کرده و حاصل آن را در متغیر result ذخیره کرده‌ایم و در نهایت آن را نمایش داده‌ایم.

در صورت قابل درک نبودن هر قسمت و هر تمرین می‌توانید سوالات، دیدگاه و پیشنهادات خود را در قسمت نظرات بیان 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.