زنگ سی‌شارپ – قسمت هجدهم

نوشته‌ی مسعود درویشیان

لینک مستقیم این مطلب در وب‌تارگت

در قسمت قبل با عمل‌گرهای منطقی و بیتی آشنا شدید. همچنین توضیحات مختصری در مورد آرایه یک بعدی داده شد. در این قسمت قصد داریم با مثال‌های زیادی از آرایه یک بعدی آشنا شویم. همچنین دستوراتی که تاکنون توضیح داده شده‌اند در حین انجام تمرین‌ها مرور خواهند شد.

به مثال زیر توجه کنید:

```
using System;
class Example
{
    static void Main()
    {
        string[] daysOfWeek = new string[7];

        daysOfWeek[0] = "Sunday";
        daysOfWeek[1] = "Monday";
        daysOfWeek[2] = "Tuesday";
        daysOfWeek[3] = "Wednesday";
        daysOfWeek[4] = "Thursday";
        daysOfWeek[5] = "Friday";
        daysOfWeek[6] = "Saturday";

        while (true)
        {
            Console.Clear();
            Console.Write("Pick out a number bitween 1 and 7 or press q to quit: ");
            string input = Console.ReadLine();

            if (input == "q" || input == "Q")
                break;

            int index = Convert.ToInt32(input);

            if (index >= 1 && index <= 7)
            {
                Console.WriteLine(daysOfWeek[index - 1]);
            }
            else
            {
                Console.WriteLine("Invalid input!");
            }

            Console.ReadLine();
        }
    }
}
```

در این مثال ابتدا روزهای هفته را در آرایه‌ای از جنس string ذخیره کرده‌ایم سپس در یک حلقه‌ی بی‌نهایت پیغامی را نمایش داده‌ایم تا کاربر با انتخاب یکی از اعداد بین 1 تا 7 روز متناظر با آن را مشاهده کند. دلیل استفاده از حلقه‌ی بی‌نهایت در این‌جا این است که کاربر بعد از وارد کردن عدد و مشاهده روز متانظر با آن، بتواند مجدداً اعداد دیگری را نیز وارد کند و برای این‌که به اجرای برنامه خاتمه دهد کلید q را بفشارد تا برنامه از حلقه‌ی بی‌نهایت خارج شود. اجازه دهید اندکی دقیق‌تر به این برنامه نگاه کنیم، در حلقه‌ی بی‌نهایت، دستور Console.Clear() مطالبی که در صفحه‌ی کنسول در حال نمایش هستند را پاک می‌کند. البته اطلاعاتی که در برنامه‌تان ذخیره می‌کنید توسط این دستور از بین نمی‌رود و در واقع این دستوری است برای از نو نویسی اطلاعات در صفحه‌ی کنسول.

در خطوط بعدی برنامه پیغامی به کاربر نمایش داده می‌شود تا عددی بین 1 تا 7 را انتخاب کند یا کلید q را برای خروج بفشارد. در ادامه ورودی از کاربر دریافت شده و توسط دستور if بررسی می‌شود که اگر کاربر حرف q یا Q را وارد کرده برنامه توسط دستور break از حلقه‌ی بی‌نهایت خارج شود در غیر این صورت ورودی به عدد صحیح تبدیل شده و مجدداً توسط دستور if مورد بررسی قرار می‌گیرد تا بین 1 و 7 باشد.

در صورت برقراری این شرط، daysOfWeek[index - 1] آرایه نمایش داده می‌شود. به‌دلیل این‌که شمارش آرایه از صفر و عدد ورودی کاربر از 1 شروع می‌شود مقدار index را از عدد 1 کم کرده‌ایم.

به مثال بعدی توجه کنید:

```
using System;
class Example
{
    static void Main()
    {
        string[] names = new string[5];

        // Filling array
        for (int i = 0; i < 5; i++)
        {
            Console.Write("Enter name " + i + ": ");
            names[i] = Console.ReadLine();
        }

        Console.WriteLine();
        Console.WriteLine("Array's Data: ");

        // Displaying array's data
        for (int i = 0; i < 5; i++)
        {
            Console.WriteLine(names[i]);
        }
        Console.WriteLine();
    }
}
```

در این مثال ابتدا توسط حلقه‌ی for در هر بار مقداری از کاربر گرفته شده و درون خانه‌های آرایه قرار داده می‌شود. در حلقه‌ی for دوم اطلاعات ذخیره شده نمایش داده می‌شود.

در مثال بعد فرهنگ لغت کوچکی می‌سازیم:

```
using System;
class Example
{
    static void Main()
    {
        string[] words = new string[5];
        string[] translations = new string[5];
        bool isFound;
        string input;

        words[0] = "book";
        translations[0] = "ketab";

        words[1] = "mobile";
        translations[1] = "telephone hamrah";

        words[2] = "mouse";
        translations[2] = "moosh";

        words[3] = "keyboard";
        translations[3] = "safhe kilid";

        words[4] = "speaker";
        translations[4] = "bolandgoo";

        while (true)
        {
            Console.Clear();
            Console.Write("Enter a word to translate or press q to quit: ");
            input = Console.ReadLine();

            if (input == "q" || input == "Q")
                break;

            isFound = false;
            for (int i = 0; i < words.Length; i++)
            {
                if (input == words[i])
                {
                    isFound = true;
                    Console.WriteLine(words[i] + " Means " + translations[i]);
                    break;
                }
            }

            if (!isFound)
            {
                Console.WriteLine("Not Found!");
            }
            Console.ReadLine();
        }
    }
}
```

در این مثال دو آرایه تعریف کرده‌ایم که اولی برای نگه‌داری کلمات انگلیسی و دومی برای نگه‌داری ترجمه‌ی معادل آن‌هاست. کلمه و ترجمه‌ی معادل آن در آرایه باید در index یکسانی باشند. به‌عنوان مثال اگر کلمه‌ی book در ایندکس صفر است، ترجمه‌ی آن نیز باید در ایندکس صفر باشد. سپس در یک حلقه‌ی بی‌نهایت پیغامی را نمایش داده‌ایم تا ورودی را از کاربر بگیریم. اگر کاربر حرف q یا Q را وارد کند برنامه از حلقه بی‌نهایت خارج شده و اجرا به پایان می‌رسد. در غیر این‌صورت توسط یک حلقه‌ی for تک تک خانه‌های آرایه words را بررسی می‌کنیم تا در صورت پیدا شدن کلمه وارد شده، ترجمه‌ی موبوط به‌آن نمایش داده شود. در این میان از یک متغیر بولین برای این‌که بدانیم کلمه وارد شده در فرهنگ لغت ما وجود دارد یا خیر، کمک گرفته‌ایم. همچنین عبارت words.Length بیان کننده‌ی طول خانه‌های آرایه words است.

در مثال‌های قبلی برای پر کردن خانه‌های آرایه از عبارت‌های جدا استفاده می‌کردیم:

```
arrays[0] = someValue;
arrays[1] = someValue;
.
.
.
```

در حالی‌که این روش صحیح است، روش ساده‌تری نیز وجود دارد که هم‌زمان با تعریف آرایه به آن مقدار می‌دهیم.

فرم کلی این روش به‌شکل زیر است:

```
type[] array-name = { val1, val2, val3, ..., valN };
```

به مثال زیر که از این روش استفاده شده و میانگین اعداد محاسبه می‌شود توجه کنید:

```
using System;
class Example
{
    static void Main()
    {
        int[] nums = { 22, 56, 78, 96, 32, 15, 4 };
        int average = 0;

        for (int i = 0; i < nums.Length; i++)
        {
            average = average + nums[i];
        }

        average = average / nums.Length;

        Console.WriteLine(average);
    }
}
```

در این موارد، سی‌شارپ به‌صورت خودکار اندازه آرایه را با توجه به مقادیری که شما معین کرده‌اید تعیین می‌کند.

رعایت مرز و حدود آرایه‌ها در سی‌شارپ اجباری است. اگر ظرفیت آرایه‌ای 10 است و شما تلاش کنید که تا 11 مقدار را در آن جای دهید با خطای IndexOutOfRangeException مواجه می‌شوید.

اگر سعی در اجرای برنامه زیر کنید خطای نام‌برده را دریافت خواهید کرد:

```
using System;
class Example
{
    static void Main()
    {
        int[] myArray = new int[10];

        for (int i = 0; i < 20; i++)
            myArray[i] = i;
    }
}
```

در این برنامه ظرفیت آرایه ما 10 است و ما در تلاش بودیم تا 20 مقدار را در این آرایه جای دهیم که با خطای IndexOutOfRangeException مواجه شدیم.

در مثال بعدی قصد داریم برنامه فرهنگ لغت را به روش دیگری بنویسیم:

```
using System;
class Example
{
    static void Main()
    {
        string input;
        bool isFound;
        string[] words = {
                            "hello", "salam",
                            "example", "mesal",
                            "revolver", "7tir",
                            "doorway", "rahro"
                        };

        for (; ; )
        {
            Console.Clear();
            Console.ForegroundColor = ConsoleColor.Yellow;
            Console.WriteLine("Words: ");
            Console.ForegroundColor = ConsoleColor.White;
            for (int i = 0; i < words.Length; i += 2)
            {
                if (i == words.Length - 2)
                {
                    Console.Write(words[i]);
                    break;
                }
                else
                    Console.Write(words[i] + " -- ");
            }

            Console.WriteLine();
            Console.WriteLine();
            Console.ForegroundColor = ConsoleColor.Yellow;
            Console.Write("You can pick out a word or press q to quit: ");
```

```
            input = Console.ReadLine();

            if (input == "q" || input == "Q" || input == "quit")
            {
                Console.ForegroundColor = ConsoleColor.White;
                Console.WriteLine("Thank you for coming. We hope to see you again. Bye");
                break;
            }

            isFound = false;
            Console.ForegroundColor = ConsoleColor.Green;
            for (int i = 0; i < words.Length; i += 2)
            {
                if (input == words[i])
                {
                    isFound = true;
                    Console.WriteLine(words[i] + " Means " + words[i + 1]);
                    break;
                }
            }

            if (!isFound)
            {
                Console.ForegroundColor = ConsoleColor.Red;
                Console.WriteLine("Not Found!");
            }

            Console.ReadLine();
        }
    }
}
```

روش انجام این مثال تا حد زیادی با روش قبل مشابه است. در این‌جا از `for ( ; ; )` به‌عنوان حلقه‌ی بی‌نهایت استفاده کرده‌ایم. همچنین از دستور `Console.ForegroundColor` برای رنگ متون بهره برده‌ایم. تفاوت اصلی که مد نظر ما بود نحوه دیگر ایجاد آرایه و مقدار دهی به‌آن بود که در این مثال به‌کار بردیم.

در مثال بعدی قصد داریم درفترچه تلفن ساده و کوچکی بسازیم:

```
using System;
class Example
{
    static void Main()
    {
        string[] names = new string[10];
        long[] numbers = new long[10];
        int counter = 0;

        string[] menu = {
                            "#### Simple Phonebook ####",
                            "",
                            "1. Add Contact",
                            "2. Search Contact",
                            "3. List All Contact",
                            "4. Exit",
                            "-------------------------"
                        };

        while (true)
```

```
        {
            Console.Clear();
            for (int i = 0; i < menu.Length; i++)
            {
                Console.WriteLine(menu[i]);
            }
            Console.Write("Pick out a number between 1 and 4: ");
            string userChoice = Console.ReadLine();

            switch (userChoice)
            {
                case "1":
                    Console.Write("Enter Name: ");
                    string name = Console.ReadLine();
                    Console.Write("Enter Number: ");
                    long number = Convert.ToInt64(Console.ReadLine());
                    names[counter] = name;
                    numbers[counter] = number;
                    counter++;
                    Console.WriteLine("Your Contact added succesfully.");
                    break;
                case "2":
                    Console.Write("Enter you contact name: ");
                    string nameForSearch = Console.ReadLine();
                    for (int i = 0; i < counter; i++)
                    {
                        if (nameForSearch == names[i])
                        {
                            Console.WriteLine("Name: {0} - Number: {1}", names[i], numbers[i]);
                            break;
                        }
                    }
                    break;
                case "3":
                    for (int i = 0; i < counter; i++)
                    {
                        Console.WriteLine("Name: {0} - Telephone: {1}", names[i], numbers[i]);
                    }
                    break;
                case "4":
                    Environment.Exit(0);
                    break;
                default:
                    Console.WriteLine("Invalid Choise");
                    break;
            }
            Console.ReadLine();
        }
    }
}
```

در این مثال از طریق دستور switch، حلقه‌ها و آرایه، دفترچه تلفنی ساده را برنامه‌نویسی کردیم. در این‌جا منوی برنامه را درون یک آرایه قرار داده‌ایم و از طریق حلقه‌ی for آن را به نمایش گذاشته‌ایم. درون حلقه‌ی بی‌نهایت از طریق دستور switch گزینه‌ای را که کاربر انتخاب کرده، گرفته شده و عملیات مربوط به هر مورد انجام می‌شود. دستور `Environment.Exit(0)` موجب می‌شود برنامه شما در همان‌جا به پایان برسد.

مثال‌ها و تمرین‌هایی که در این قسمت انجام شد فوق‌العاده از اهمیت زیادی برای یادگیری سی‌شارپ برخوردار هستند. همچنین برای دنبال کردن ادامه‌ی مقالات زنگ سی‌شارپ نیاز است تا این مثال‌ها را بسیار تمرین کرده، درک کنید و بر آن‌ها مسلط شوید. پس خوب مطالب را برای خودتان تجزیه و تحلیل کنید و در صورت نامفهموم بودن هر مثال، سوال‌های خود را در قسمت نظرات بیان کنید تا در اسرع وقت توضیحات تکمیلی و راهنمایی برای فهم و درک بهتر مطالب به شما عزیزان داده شود.

تمرین

تمرین شماره ۱۲: برنامه‌ی دفترچه تلفنی بنویسید که قابلیت اضافه کردن مخاطب جدید، جستجو بر اساس نام و شماره تلفن، ویرایش مخاطبان ذخیره شده و پاک کردن را داشته باشد.