

یکی دیگر از کاربردهای Action و Func، امکان حذف و بازنویسی switch statements بسیار حجیم و طولانی به نحوی شکل است؛ و در ادامه این نوع Refactoring را بررسی خواهیم کرد. در ابتدا مثال زیر را که از یک سوئیچ، برای انتخاب نوع حرکت و اعمال آن استفاده می‌کند، در نظر بگیرید:

```
using System;

namespace ActionFuncSamples
{
    public enum Direction
    {
        Up,
        Down,
        Left,
        Right
    }

    public class Sample5
    {
        public void Move(Direction direction)
        {
            switch (direction)
            {
                case Direction.Up:
                    MoveUp();
                    break;

                case Direction.Down:
                    MoveDown();
                    break;

                case Direction.Left:
                    MoveLeft();
                    break;

                case Direction.Right:
                    MoveRight();
                    break;
            }
        }

        private void MoveRight()
        {
            Console.WriteLine("MoveRight");
        }

        private void MoveLeft()
        {
            Console.WriteLine("MoveLeft");
        }

        private void MoveDown()
        {
            Console.WriteLine("MoveDown");
        }

        private void MoveUp()
        {
            Console.WriteLine("MoveUp");
        }
    }
}
```

یک کلاس ساده که بر اساس مقدار enum دریافتی، حرکت به چهار جهت را سبب خواهد شد. اکنون می‌خواهیم با استفاده از Actionها نحوه تعریف متدهای حرکت را به فراخوان واگذار ( [کلاسی بسته برای تغییر اما باز برای توسعه](#) ) و به علاوه switch را نیز کلا حذف کنیم:

```
public interface IMove
```

```

{
    Dictionary<Direction, Action> MoveMap { get; }
}

public class Move : IMove
{
    public Dictionary<Direction, Action> MoveMap
    {
        get
        {
            return new Dictionary<Direction, Action>
            {
                { Direction.Up, MoveUp},
                { Direction.Down, MoveDown},
                { Direction.Left, MoveLeft},
                { Direction.Right, MoveRight}
            };
        }
    }

    private void MoveRight()
    {
        Console.WriteLine("MoveRight");
    }

    private void MoveLeft()
    {
        Console.WriteLine("MoveLeft");
    }

    private void MoveDown()
    {
        Console.WriteLine("MoveDown");
    }

    private void MoveUp()
    {
        Console.WriteLine("MoveUp");
    }
}

public class Sample5
{
    public void NewMove(IMove move, Direction direction)
    {
        foreach (var item in move.MoveMap)
        {
            if (item.Key == direction)
            {
                item.Value(); //run action
                return;
            }
        }
    }
}

```

### توضیحات:

استفاده از Dictionary برای حذف سوئیچ بسیار مرسوم است. خصوصا زمانیکه توسط switch صرفا قرار است یک مقدار ساده بازگشت داده شود. در این حالت می‌توان کل سوئیچ را حذف کرد. case های آن تبدیل به key های یک Dictionary و مقادیری که باید بازگشت دهد، تبدیل به value های متناظر خواهند شد.

اما در اینجا مساله اندکی متفاوت است و خروجی خاصی مد نظر نیست؛ بلکه در هر قسمت قرار است کار مفروضی انجام شود. بنابراین اینبار کلیدهای Dictionary بازهم بر اساس مقادیر case های تعریف شده (در اینجا enum ایی به نام Direction) و مقادیر آن نیز Action تعریف خواهند شد. همچنین برای اینکه بتوان امکان تعریف قالبی (کلاسی) را برای تعاریف متناظر با اعضای enum نیز میسر کرد، این Dictionary را در یک interface قرار داده‌ایم تا کلاس‌های پیاده سازی کننده آن، تعریف متدها را نیز دربر داشته باشند.

همانطور که ملاحظه می‌کنید، اینبار تعاریف متدهای Move از کلاس Sample5 خارج شده‌اند، به علاوه برای دسترسی به آن‌ها نیز switch ایی تعریف نشده و از Action استفاده شده است. نهایتا اینبار متد جدید Move تعریف شده، با اینترفیس IMove کار می‌کند که یک Dictionary حاوی متدهای متناظر با اعضای enum جهت را ارائه می‌دهد.

یک نکته تکمیلی :

متد NewMove را به شکل‌های زیر نیز می‌توان پیاده سازی کرد:

```
// or ...
move.MoveMap[direction]();

// or ...
Action action;
if(move.MoveMap.TryGetValue(direction, out action))
    action();
```

## نظرات خوانندگان

نویسنده: ایمان عبیدی  
تاریخ: ۱۵:۳۸ ۱۳۹۱/۰۶/۱۱

خیلی ممنون ، من که خیلی دوست دارم این سری مقالات رو و دنبال میکنم.

کد این قسمت رو که برای تمرین خودم انجامش دادم ( با یکم تغییرات ناچیز) با اجازه از جناب نصیری اتچش کردم

[Sample5.rar](#)

نویسنده: حسین مرادی نیا  
تاریخ: ۱۷:۲۸ ۱۳۹۱/۰۶/۱۱

مرسی

مطب جالبی هست و من هم دنبال میکنم.