

<C# 강의> 8장. 람다식 (Lamda Expression)

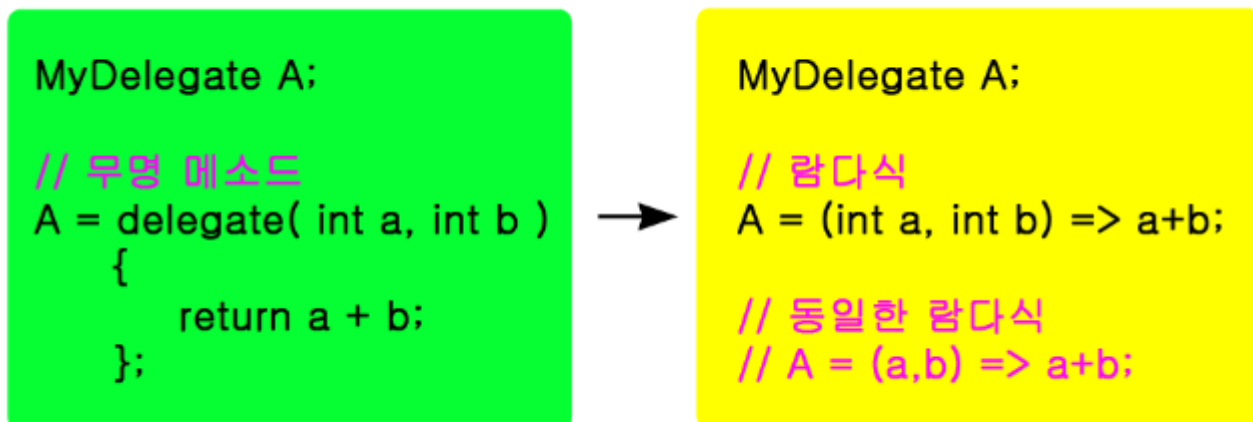
Author : Mr1 Mr1 Date : 2013.12.16 06:00 Category : C샵

1. 람다식

람다식은 **무명 메소드를 단순한 계산식으로 표현한 것** 이다.

메소드는 크게 매개변수와 내부 식, 반환값으로 구성되어 있는데,

이들만 가지고 메소드를 계산식으로 표현한다.



무명 메소드에 비해 상당히 간결하게 표현되는 것을 확인할 수 있다.

람다식은 매개변수로 전해지는 a, b의 타입까지도 생략이 가능하다.

(똑똑한 컴파일러가 좌항의 델리게이트의 타입을 참고해서 타입을 유추하는 것이다.)

그럼 람다식에 대한 다음 예제를 작성해 보자.

```

class MainApp
{
    delegate int MyDelegate(int a, int b);

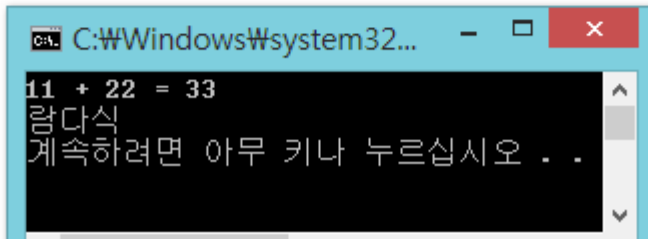
    delegate void MyDelegate2();

    static void Main(string[] args)
    {
        MyDelegate add = (a,b) => a+b;
        MyDelegate2 lamda = () => Console.WriteLine("람다식");

        Console.WriteLine( "11 + 22 = {0}", add(11, 22) );

        lamda();
    }
}

```



2. 문 형식의 람다식

지금까지의 람다식은 단순한 계산식 하나만을 표현하였다.

그렇다면 람다식 내에서 메소드처럼 다양한 처리를 해줄 순 없을까?

당연히 가능하다. 그냥 메소드처럼 중괄호 내에 작성하면 된다.

```

delegate void MyDelegate(int a, int b );

MyDelegate A = ( a, b ) =>
{
    if( a > b )
        Console.WriteLine("{0}가 크다", a );
    else if( a < b )
        Console.WriteLine("{0}가 크다", b );
    else
        Console.WriteLine("{0}, {1}는 같다", a, b );
};

```

그냥 메소드를 간략화 했다고 보면 될 거 같다.

위의 내용을 프로그램으로 작성해서 확인해보자.



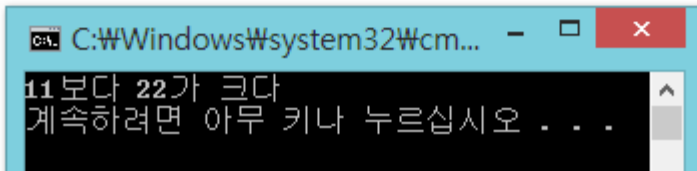
```

class MainApp
{
    delegate void MyDelegate(int a, int b);

    static void Main(string[] args)
    {
        MyDelegate Compare = (a, b) =>
        {
            if (a > b)
                Console.WriteLine("{0}보다 {1}가 크다", b, a);
            else if (a < b)
                Console.WriteLine("{0}보다 {1}가 크다", a, b);
            else
                Console.WriteLine("{0},{1}는 같다", a, b);
        };

        Compare(11, 22);
    }
}

```



공감

Tags C# Lamda, C# 람다식, Lamda, Lamda Expression, 람다식, 문 형식 람다식

Comments 3

Trackbacks 0

소경현 2017.01.31 11:51 신고

감사합니다!

Reply

Edit

궁금 2017.02.27 17:40 신고

궁금한것이 람다식이 델리게이트의 상위호환인가요?

실제 유지보수하며 람다는 많이 보았는데 델리게이트로 구현된건 전혀없네요..

실제업무에서 어떤케이스에서 델리게이트를 쓰는지 좀 궁금한데 글쓴이님의 노하우좀 알려주실수있나요??

-- 아 강의가 너무좋아요 이해가 쏙쏙되네요 감사합니다!

Reply

Edit

