

假设我们有如下的一个整数列表{4,8,14}，我们想要获取这个列表中的每个数值的阶乘的话，我们可能我这样写代码：

```
class Program
{
    static void Main(string[] args)
    {
        List<long> list = new List<long> { 4, 8, 14 };
        var factoriallist = list.Select(n => Factorial(n)).ToList();
    }

    private static long Factorial(long n)
    {
        return n == 1 ? n : n * Factorial(n - 1);
    }
}
```

那么我们有没有办法把上面Factorial函数中的递归代码直接写在Select方法的Lambda表达式中呢？关键就是我们要在Lambda表达式中得到自己，然后再调用自己，这点可以使用MethodBase.GetCurrentMethod()轻松办到，具体代码如下：

```
static void Main(string[] args)
{
    List<long> list = new List<long> { 4, 8, 14 };

    var factoriallist = list.Select(
        n =>
        {
            return n == 1 ? n
                : n * (long)MethodBase.GetCurrentMethod().Invoke(this, new object[] { n - 1 });
        }).ToList();
}
```