```
private static void TestMethod()
{
            //dynamic 仅仅是个占位符而已
            dynamic p1 = new { X = 1, Y = 89 };
            //对dynamic对象p1的X属性访问通过CallSite来实现，并且CallSite可以Cache
            object p1x = p1.X;
            object p1y = p1.Y;

            //dynamic 仅仅是个占位符而已
            dynamic p2 = new { X = 1, Y = 89, Z = 3 };
            object p2x = p2.X;
            object p2y = p2.Y;
            object p2z = p2.Z;

}
```

对于如上所示的含有dynamic调用的C#方法编译后大致的结果如下：

```
using System;
using Microsoft.CSharp.RuntimeBinder;
using System.Runtime.CompilerServices;

private static void TestMethod()
{
    object p1 = new {
        X = 1,
        Y = 0x59
    };

    if (TestMethod_SiteContainer.p_Sitea == null)
    {
        TestMethod_SiteContainer.p_Sitea = CallSite<Func<CallSite, object, object>>.Create
    }
    object p1x = TestMethod_SiteContainer.p_Sitea.Target(TestMethod_SiteContainer.p_Sitea,

    if (TestMethod_SiteContainer.p_Siteb == null)
    {
        TestMethod_SiteContainer.p_Siteb = CallSite<Func<CallSite, object, object>>.Create
    }
    object p1y = TestMethod_SiteContainer.p_Siteb.Target(TestMethod_SiteContainer.p_Siteb,

    object p2 = new {
        X = 1,
        Y = 0x59,
        Z = 3
    };
    if (TestMethod_SiteContainer.p_Sitec == null)
    {
        TestMethod_SiteContainer.p_Sitec = CallSite<Func<CallSite, object, object>>.Create
    }
    object p2x = TestMethod_SiteContainer.p_Sitec.Target(TestMethod_SiteContainer.p_Sitec,

    if (TestMethod_SiteContainer.p_Sited == null)
    {
        TestMethod_SiteContainer.p_Sited = CallSite<Func<CallSite, object, object>>.Create
    }
    object p2y = TestMethod_SiteContainer.p_Sited.Target(TestMethod_SiteContainer.p_Sited,

    if (TestMethod_SiteContainer.p_Sitee == null)
    {
        TestMethod_SiteContainer.p_Sitee = CallSite<Func<CallSite, object, object>>.Create
    }
    object p2z = TestMethod_SiteContainer.p_Sitee.Target(TestMethod_SiteContainer.p_Sitee,
}

[CompilerGenerated]
private static class TestMethod_SiteContainer
{
    // Fields
    public static CallSite<Func<CallSite, object, object>> p_Sitea;
    public static CallSite<Func<CallSite, object, object>> p_Siteb;
    public static CallSite<Func<CallSite, object, object>> p_Sitec;
    public static CallSite<Func<CallSite, object, object>> p_Sited;
    public static CallSite<Func<CallSite, object, object>> p_Sitee;
}
```