

برای اجرای متد درون یک کلاس از طریق Reflection ابتدا نوع آن کلاس را به دست می‌آوریم و سپس از طریق کلاس Activator.CreateInstance یک نمونه از آن کلاس را ساخته و در متغیری از نوع object ذخیره کرده و با استفاده از GetMethod اطلاعات متد مورد نظر خود را در متغیری ذخیره کرده و سپس از طریق دستور Invoke آن متد را اجرا می‌کنیم. دستور Invoke سربارگذاری دارد که در یک نوع از آن، متغیر حاوی نمونه کلاس و پارامترهای متد مورد نظر، در قالب یک آرایه از نوع object، به عنوان آرگومان پذیرفته می‌شود. با امضای زیر

```
public Object Invoke(Object obj, Object[] parameters)
```

به مثال زیر که چگونگی این عملیات را شرح می‌دهد، توجه کنید:

```
public class TestMath
{
    public int Squar(int i)
    {
        return i*i;
    }
}

static void Main(string[] args)
{
    Type type = typeof (TestMath); // به دست آوردن نوع کلاس
    object obj = Activator.CreateInstance(type); // ساختن نمونه‌ای از نوع مورد نظر
    MethodInfo methodInfo = type.GetMethod("Squar"); // یافتن اطلاعات متد مورد نظر
    Console.WriteLine(methodInfo.Invoke(obj, new object[] { 100 })); // نمایش نتیجه و object[]
    // ارسال عدد 100 به صورت
    Console.Read();
}
```

توجه کنید که دو متد GetMethod و Invoke در فضای نام System.Reflection قرار دارند.

روش دیگر

در شیوه دیگر برای انجام این کار، نیازی به استفاده از GetMethod و Invoke نیست و فراخوانی متد مورد نظر بسیار شبیه فراخوانی عادی متدهاست و نیازی به ساخت متغیر ویژه‌ای از نوع object[] برای ارسال پارامترها نیست. برای انجام این کار فقط کافیست نوع متغیری که نوع نمونه‌سازی شده را نگه می‌دارد (در اینجا نمونه‌ای از کلاس را نگه می‌دارد) به صورت dynamic باشد:

```
static void Main(string[] args)
{
    Type type = typeof (TestMath);
    dynamic obj = Activator.CreateInstance(type);
    Console.WriteLine(obj.Square(100));
    Console.Read();
}
```

توجه کنید که بعد از تعریف obj، با درج نقطه در کنار آن، منوی Code Insight متد Square را شامل نمی‌شود اما کامپایلر آن را می‌پذیرد.

نظرات خوانندگان

نویسنده: KishIsland

تاریخ: ۱۹:۴۷ ۱۳۹۱/۱۲/۱۵

برای تست کد بالا من یک کلاس بشکل زیر تعریف کردم:

```
class Test
{
    public void func1()
    {
        Console.WriteLine("Hello World!");
    }
}
```

و در قسمت main برنامه فراخوانی رو بشکل زیر نوشتم مطابق روش دوم ولی برنامه Exception داد:

```
Type type = typeof(Test);
dynamic obj = Activator.CreateInstance(type);
Console.WriteLine(obj.func1());
```

علت بروز استثناء void بودن متد هست. می خواستم بدونم برای مواقعی که متد بصورت void تعریف شده چکار باید کرد؟
سپاس

نویسنده: وحید نصیری

تاریخ: ۲۱:۲۰ ۱۳۹۱/۱۲/۱۵

در این حالت بجای

```
Console.WriteLine(obj.func1());
```

فقط کافی هست بنویسید

```
obj.func1();
```