عنوان: **آموزش Code Contracts**
نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۱/۱۱/۲۳ ۸:۲۵
آدرس: www.dotnettips.info
برچسب‌ها: Refactoring, Code Contracts

احتمالا تا حالا شده که می‌خواستید متدهایی بنویسید که داده‌های ورودی رو چک کنند و از درست بودن مقادیر اطمینان حاصل کنید و احتمالا کدهای شما هم مثل نمونه پایین هستش

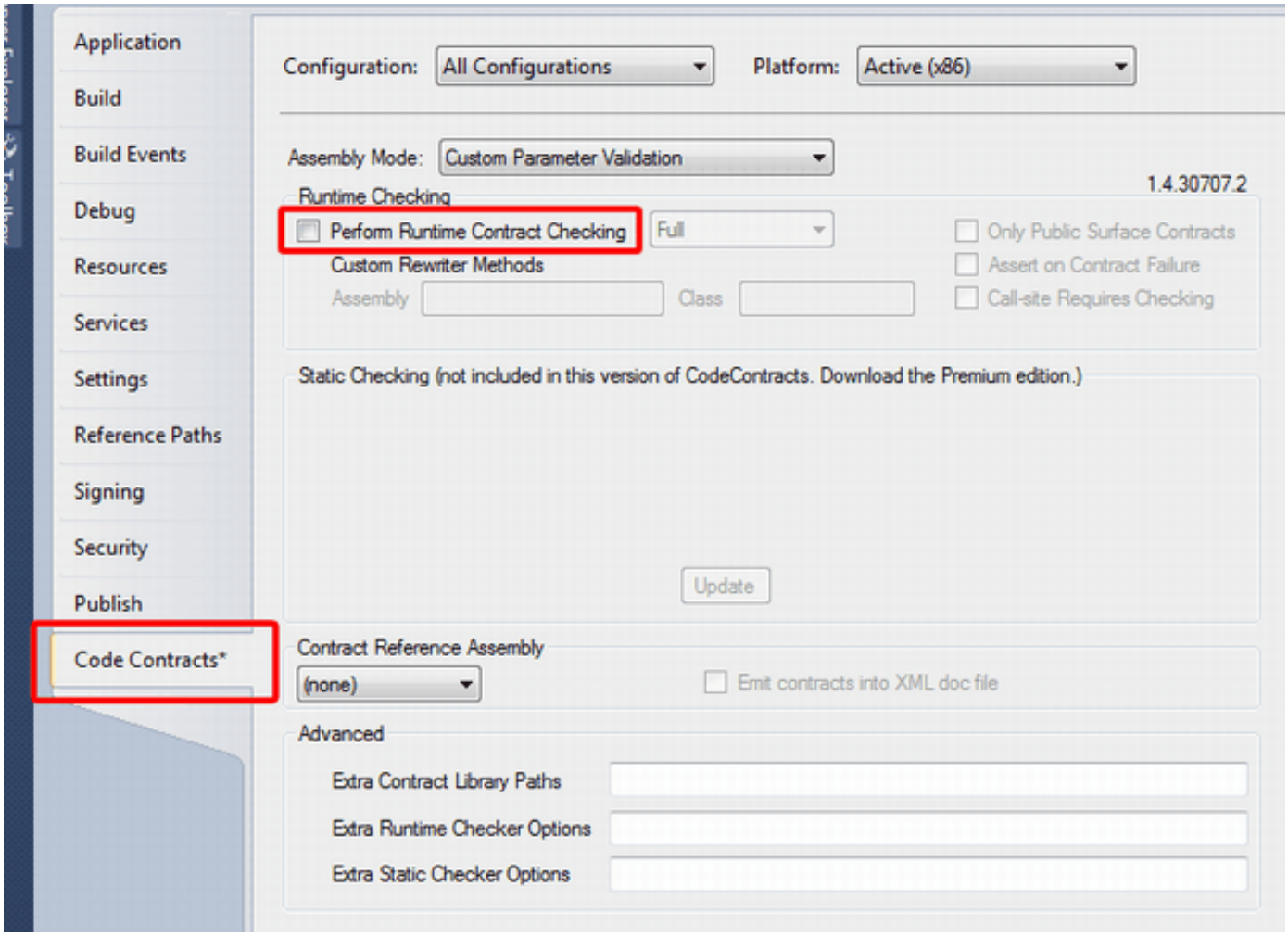
```
public class timeclock
    {
        public void clockin( int32 id, datetime clockdate )
        {
            if ( id < 0 )
            {
                throw new argumentoutofrangeexception( "..." );
            }

            if ( clockdate.date != datetime.now.date )
            {
                throw new argumentexception( "..." );
            }
        }
        public dailyreport getdailyreport( int32 employeeid, datetime fordate )
        {
            var dailyreport = new dailyreport();
            return dailyreport;
        }
    }

    public class dailyreport
    {
        public int32 employeeid { get; set; }
        public int32 hoursworked { get; set; }
    }
```

code contracts در واقع تهیه یک سری قرارداد برای اطمینان از پیاده سازی شروط در برنامه هستش و نکته مهمش اینه که شما را در هنگام کامپایل از این خطاهای احتمالی آگاه میکنه. Microsoft code contract ابزاری برای پیاده سازی این روشه و باید فایلشو دانلود کرده و بر روی visual studio نصب کنید که از این جا می‌تونید دانلودش کنید. بعد از دانلود و نصب، یک قسمت به project properties اضافه می‌شه. اون قسمتی که قرمز رنگ هست برای اجرای قراردادها به صورت runtime هستش و از combo کنار برای تعیین نوع checking استفاده میشه.

نکته: برای استفاده از این روش اگر از net4 به پایین استفاده می‌کنید باید فضای نام microsoft.contracts را به پروژه اضافه کنید ولی برای .net4 به بالا نیازی به این کار نیست. چون کلاس‌های مربوطه در فضای نام system.diagnostics.contracts قرار دارند.

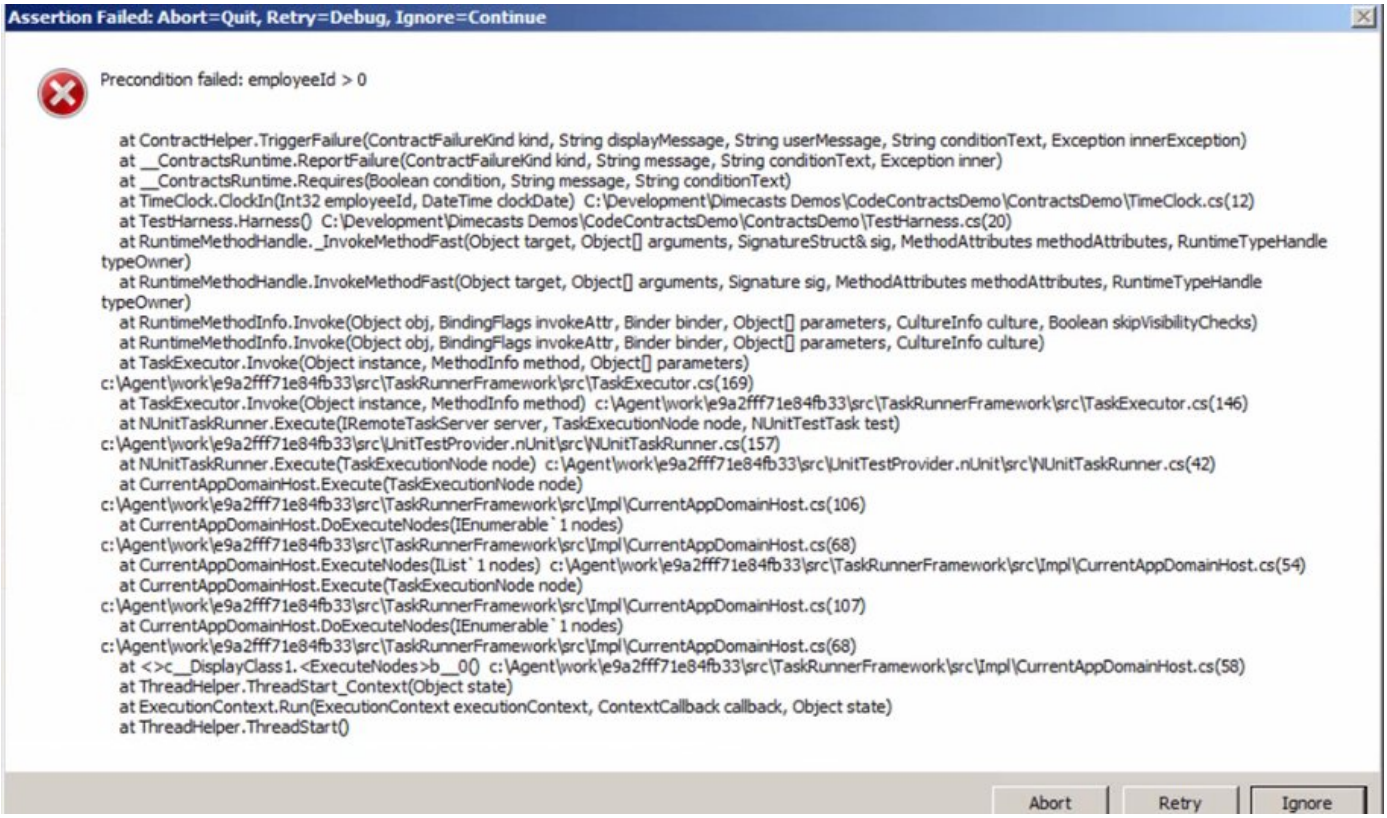


حالا مثال بالا رو به روش زیر پیاده سازی می‌کنیم

```
public void clockin( int32 id, datetime clockdate )
        {
            contract.requires<argumentoutofrangeexception>( id < 0 );
            contract.requires<argumentexception>( clockdate.date != datetime.now.date );
            contract.endcontractblock();
        }
```

فرق این روش با روش قبلی اینه که اگر در برنامه متد clockin رو به روش پایین استفاده کنیم، در هنگام اجرای برنامه با خطای زیر متوقف و رو برو میشیم

```
var timeclock = new timeclock();
  timeclock.clockin( -1, datetime.now );
```



در مطالب بعدی بیشتر به این مورد می‌پردازم

نظرات خوانندگان

نویسنده: داریوش
تاریخ: ۱۳۹۲/۰۸/۱۳ ۱۲:۳۱

مطلب عالی بود. آیا از این راه میشه برای پیاده سازی یک Business Rule Engine استفاده کرد؟

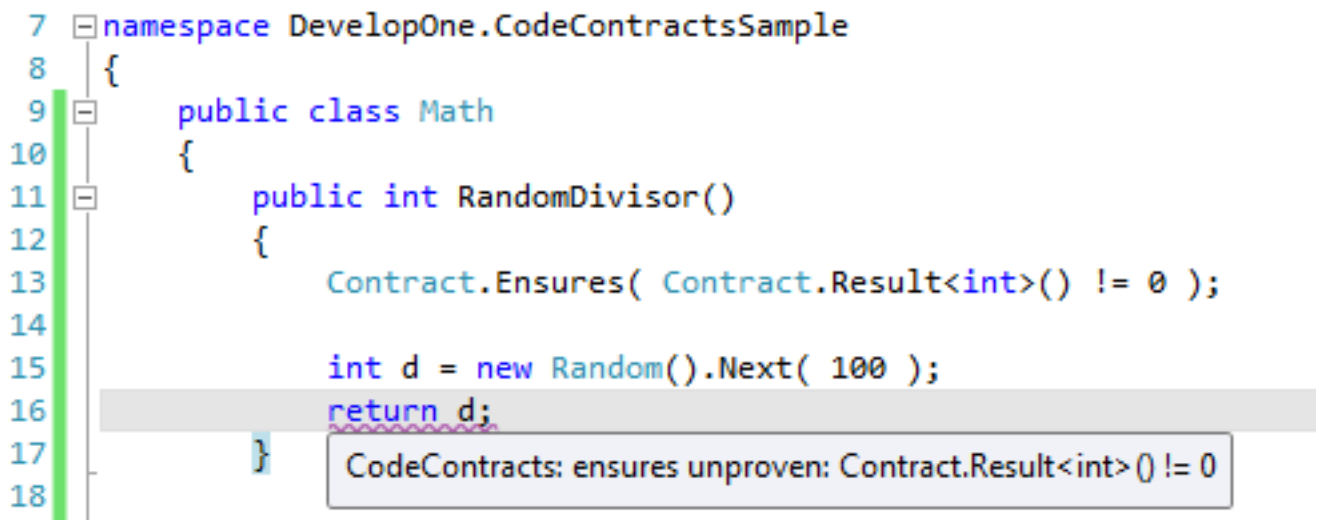
نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۰۸/۱۳ ۱۷:۴

BRE سیستمی است برای تهیه Business Rule توسط شخصی غیر برنامه نویس. در حالی که Code Contract در فاز توسعه نرم افزار مورد استفاده قرار میگرد و فقط به شما در بهتر توسعه دادن سیستم کمک می‌کند. برای مثال:

```
7  namespace DevelopOne.CodeContractsSample
8  {
9      public class Math
10     {
11         public double Divide(int number, int divisor)
12         {
13             Contract.Requires<ArgumentOutOfRangeException>(
14                 divisor != 0,
15                 "Divide by zero is not allowed." );
16
17             return number / divisor;
18         }
19
20         public void Test()
21         {
22             int n = 10;
23             int d = 0;
24
25             var x = Divide( n, d );
26         }
27     }
28 }
29
```

CodeContracts: requires is false: divisor != 0 (Divide by zero is not allowed.)

همان طور که مشاهده می‌کنید با استفاده از تعریف Contract قبل از اجرای برنامه برای ما مشخص خواهد شد مقدار پیش فرض 0 برای متغیر d درست نیست در واقع اصلا این کد کامپایل نمی‌شود.



یا در مثال بالا مشخص شده است که مقدار d ممکن است که برابر صفر باشد و این با Contract تعریف شده مطابقت ندارد. در نتیجه در تهیه یک سیستم BRE کمک خاصی به شما نخواهد کرد.

به این نکته نیز توجه داشته باشید که با تمام مزیت هایی که Code Contracts در اختیار ما قرار می‌دهد، زمان کامپایل پروژه را به شدت افزایش خواهد داد به طوری که در یک Solution نسبتا بزرگ آزار دهنده است.