

επιστημηの

# オブジェクト 指向的 夜話

★ C#から見た  
★ .NET Framework

第 5 回

## 雛形生成 for NUnit

επιστημη  
えびすてーめー



### Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:  
NUnit v2.1

### Level



### Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥YAWAディレクトリに収録しています。

¥COCUPPA\_CONSOLE  
コンソール版

¥COCUPPA\_CORE  
コアクラス

¥COCUPPA\_GUI  
GUI版

### 近況報告などなど

夏休みもそろそろ終わろうという8月25日、“theSpoke東京オフ会”にお邪魔しました。中身はほとんどProfessional版でありながら5,000円を切るという大盤振舞パッケージ「Visual Studio .net theSpoke Premium version 2003」ですが、これには鍵が掛かっている、鍵を外すには学生証のコピーを添えて登録しなきゃいかんだそうです。theSpokeサイトに<sup>[注1]</sup>に登録すると掲示板が使える、自分の掲示板とWebLog空間が貰えます。theSpokeユーザー専用サイトってことですが、おやぢな僕もちゃっかり居座っております、はい。で、theSpokeが福岡/大阪/名古屋/東京/札幌でオフ会を開いたわけで、その東京会場にこのこと顔出してきたと

いう次第。……いやー、若いっていなあ。親子程も歳の差のある若い衆に囲まれ、ちょっと元気のおすそ分けをいただいた気分です。

オフ会も終わり（福岡は台風には逆らえずやむなく中止、残念!）、その余韻に浸っていたら瞬く間に夏休みも終わりです（僕には関係ないけどさ）。そうこうしているうちに連載原稿の締め切りが近づきます。dotNETマガジン最新号が届けられ、無言の圧力を加えてきます。読者のページを眺めると“.NETのテストツール”の話題希望!という投書がありました。……これだ、これでいきましょう。

オフ会は.NETの煽り(?)ムービーやらImageCupのプレゼンやらお土産争奪ビンゴ大会(乱数生成はC#でその場でこしらえたやつ!)の後、ピザとソフトドリンク(学生だもの)とで懇親会となりました。なんだかハイになってた僕は懇親会の席でゴーマンかましましたで

注1) <http://jp.thespoke.net/>

す。この連載でも書いたけど、RADの手軽さに流されるな、見てくれと実体を分離しろよ、と。分離することは、.NETではライブラリ化が簡単なので特に効果的に作用しますからね。

## テストコードの構造

.NET版単体テストフレームワークの代表格ともいえるNUnitでは、テストコードを書き、その中でテスト対象を呼び出して結果が期待通りであるかを検証します。テストコードをコンパイルし、できあがったライブラリ(DLL)をnunit-guiに食わせるとテスト項目を片っ端から実行して、成功なら緑/失敗なら赤のバーがテスト結果を教えてください(図1)。テスト駆動開発では、テスト対象とテストとを組みにして、テストコードを実行することで“正しく動く”状態を維持しながらテスト対象すなわち最終目的物の実装を進めていきます。

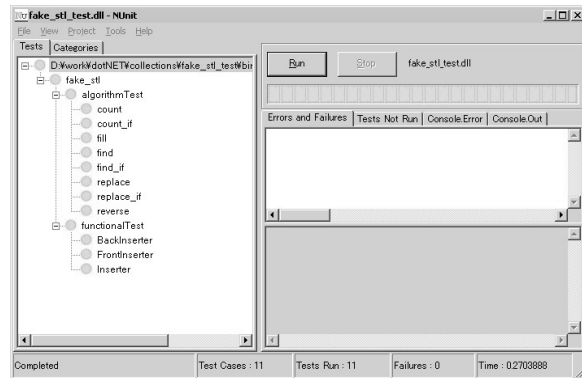
NUnitによるテストコードはお決まりの形式(リスト1)で書かれます。これをコンパイルしNUnit-guiで実行すると、Console.Outタブ内に以下のように出力されます。

```
TestFixtureSetUp
SetUp
Test(1)
TearDown
SetUp
Test(2)
TearDown
SetUp
Test(3)
TearDown
TestFixtureTearDown
```

テストの集合をTestFixtureと呼ぶのですが、NUnitではアトリビュート[TestFixture]の付いたクラスをTestFixtureと認識します。さらに、その中の各メソッドについて、

- [TestFixtureSetUp] / [TestFixtureTearDown]  
それぞれ最初と最後に呼ばれる (TestFixtureの初期

図1：NUnitでテスト中



化と後始末)

- [SetUp] / [TearDown]

各Testの最初と最後に呼ばれる (Testの初期化と後始末)

- [Test]

テスト項目

の各アトリビュートを付けておきます (TestFixtureSetUp/TestFixtureTearDownおよびSetUp/TearDownは必要なければ定義しなくても構いません)。

この雛形 (というよりC#のコードそのもの) はきれいに整った木構造を成しています。つまり、名前空間 (namespace) の中にクラス (class) があり、クラスの中にメソッド (method) があります。このような木構造のデータをプログラムで実現すれば、それに基づいて雛形を吐かせることができそうです。

名前空間 (namespace)、クラス (class) そしてメソッドは、いずれも適当な文字列に続いて「{」ではじまり、「}」で終わります。これをFragment (断片) と呼ぶことにしましょう。すなわちそれぞれ「NamespaceFragment」「ClassFragment」「MethodFragment」です。

その上、Fragmentの{}の内側に、さらにFragmentを内包するものがあります。これをCompositeFragment (合成断片) と名付けましょう。これによって、NamespaceFragmentとClassFragmentがCompositeFragmentとなります。