

Visual Studio2008

単体テスト機能



佐野 尚之





本ドキュメントのライセンスについて

この作品は、クリエイティブ・コモンズのAttribution 3.0 Unportedライセンスの下でライセンスされています。

この使用許諾条件を見るには、<http://creativecommons.org/licenses/by/3.0/>をチェックするか、クリエイティブ・コモンズに郵便にてお問い合わせください。

住所は：171 Second Street, Suite 300, San Francisco, California 94105, USA です。



原作者のクレジット（氏名、作品タイトルとURL）を表示することを守れば、改変はもちろん、営利目的での二次利用も許可される最も自由度の高いCCライセンス。

<http://creativecommons.jp/>





変更履歴

■ 第1版

2013/11/10

オープンソースの「LibreOffice 4.0.5.2」を使用して作成



目次

- ・ 動作確認環境 5
- ・ 単体テスト機能について 6
- ・ サンプル 7
- ・ 参考情報 14





動作確認環境

■ 動作確認環境

Acer Aspire 1410

Intel Celeron processor SU2300(1.2GHz, 800MHz FSB)。8GBメモリに変更。HDD250GB。

Windows 7 Home Premium(64bit版) Service Pack1

Visual Studio 2008 Professional Edition・・・昔、MSDNを契約していた事があったので



単体テスト機能について

- Visual Studio 2005では上位エディションであるTeam Developerでのみ利用可能だった単体テスト機能。Visual Studio 2008からは、Professional Editionでも利用可能となっていた。
- Visual Studio 2008 Professional Editionの単体テスト機能は、完全に入力値／出力値にのみに着目した「ブラックボックス・テスト」を行うためのもの。
- 実メソッドから単体テストのコードを生成する機能。Visual Studio 2012からやり方が変わったらしい興味があれば、以下のサイトを参照してください。

単体テストの作り方がVisual Studio 2012から変わった

<http://pgnote.net/?p=1680>

Visual Studio 2012で「単体テストの作成」機能が無くなったけど良い感じ

<http://d.hatena.ne.jp/aoki1210/20120915/p1>

Visual Studio 2012 Express for Windows Desktop で単体テストを試してみる

<http://hara8bunme.blog.so-net.ne.jp/2013-01-20>





サンプル(1/7)

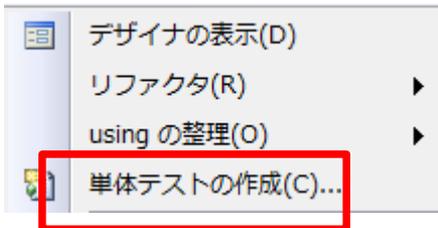
テスト対象のメソッド。

```
private int tashizan(int x, int y)
{
    return x + y;
}
```



サンプル(2/7)

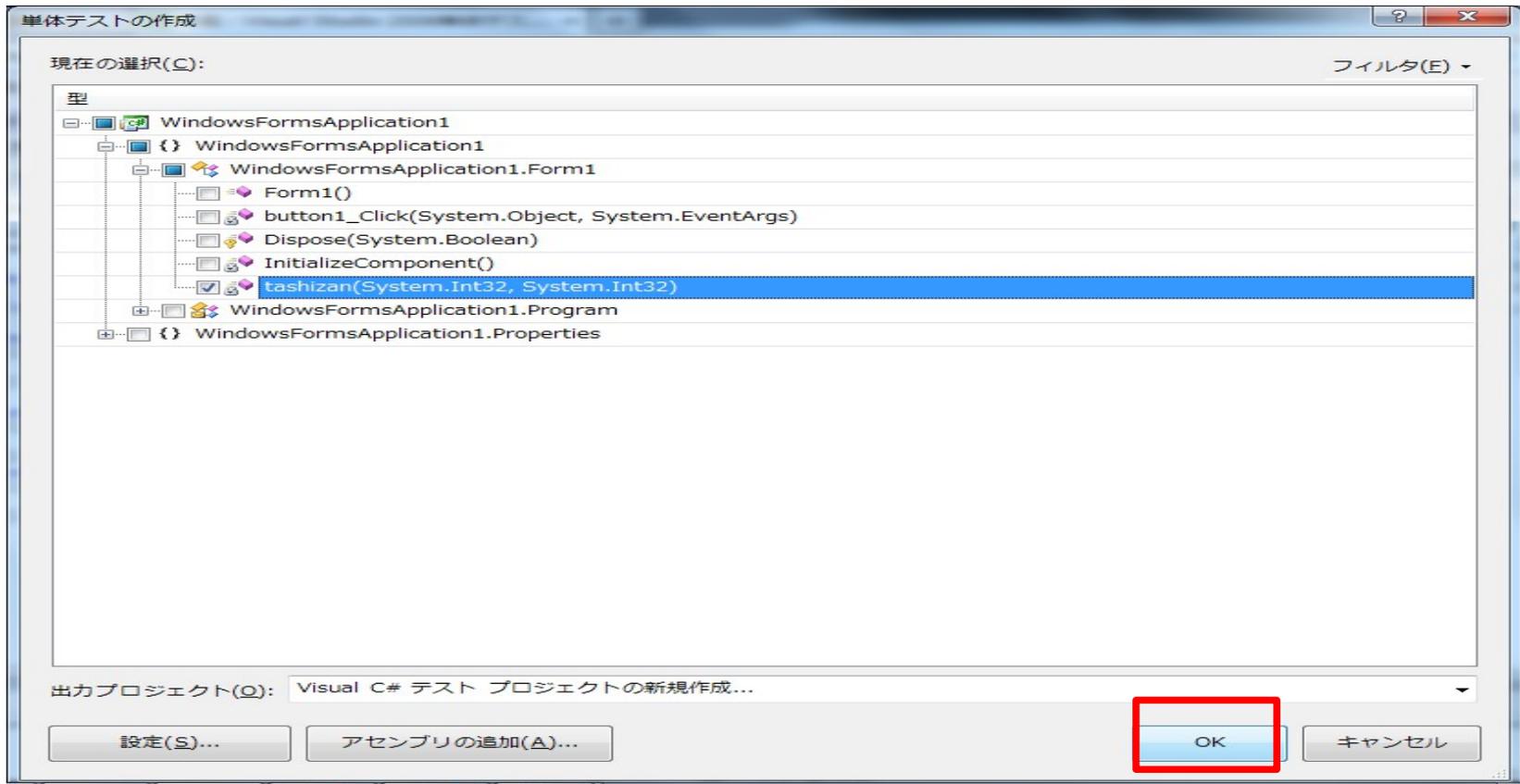
メソッドを選択ー右クリックメニューの「単体テストの作成」を選択。





サンプル(3/7)

「OK」ボタンをクリック。





サンプル(4/7)

プロジェクト名を入力して、「作成」ボタンをクリック。





サンプル(5/7)

```
using WindowsFormsApplication1;
using Microsoft.VisualStudio.TestTools.UnitTesting;
namespace tashizanTest
{
    /// <summary> ...
    [TestClass()]
    public class Form1Test
    {

        private TestContext testContextInstance;

        /// <summary> ...
        public TestContext TestContext...

        追加のテスト属性

        /// <summary> ...
        [TestMethod()]
        [DeploymentItem("WindowsFormsApplication1.exe")]
        public void tashizanTest()
        {
            Form1_Accessor target = new Form1_Accessor(); // TODO: 適切な値に初期化してください
            int x = 0; // TODO: 適切な値に初期化してください
            int y = 0; // TODO: 適切な値に初期化してください
            int expected = 0; // TODO: 適切な値に初期化してください
            int actual;
            actual = target.tashizan(x, y);
            Assert.AreEqual(expected, actual);
            Assert.Inconclusive("このテストメソッドの正確性を確認します。");
        }
    }
}
```

「// TODO」の行をこれから修正します。





サンプル(6/7)

xの値に5、yの値に3、expectedの値を8に変更。

Assert.AreEqualの次の行のコードを削除。

```
/// <summary> ...  
[TestMethod()  
DeploymentItem("WindowsFormsApplication1.exe")]  
public void tashizanTest()  
{  
    Form1_Accessor target = new Form1_Accessor(); // TODO: 適切な値に初期化してください  
    int x = 5; // TODO: 適切な値に初期化してください  
    int y = 3; // TODO: 適切な値に初期化してください  
    int expected = 8; // TODO: 適切な値に初期化してください  
    int actual;  
    actual = target.tashizan(x, y);  
    Assert.AreEqual(expected, actual);  
}
```



サンプル(7/7)

現在のコンテキストのテストを実行をクリック。



以下のように表示されれば成功です。

結果	テスト名	プロジェクト	エラー メッセージ
<input type="checkbox"/> 成功	tashizanTest	tashizanTest	



参考情報

Visual Studio Professional Edition でのテスト ツールの使用

<http://msdn.microsoft.com/ja-jp/library/bb385902%28v=vs.90%29.aspx>

Visual Studio 単体テスト機能大全

<http://codezine.jp/article/corner/355>

改めてコードの品質を見直してみませんか？ (Visual Studio 各エディション機能解説添え)

<http://blogs.msdn.com/b/tomohn/archive/2011/10/28/code-quality-and-evidence.aspx>