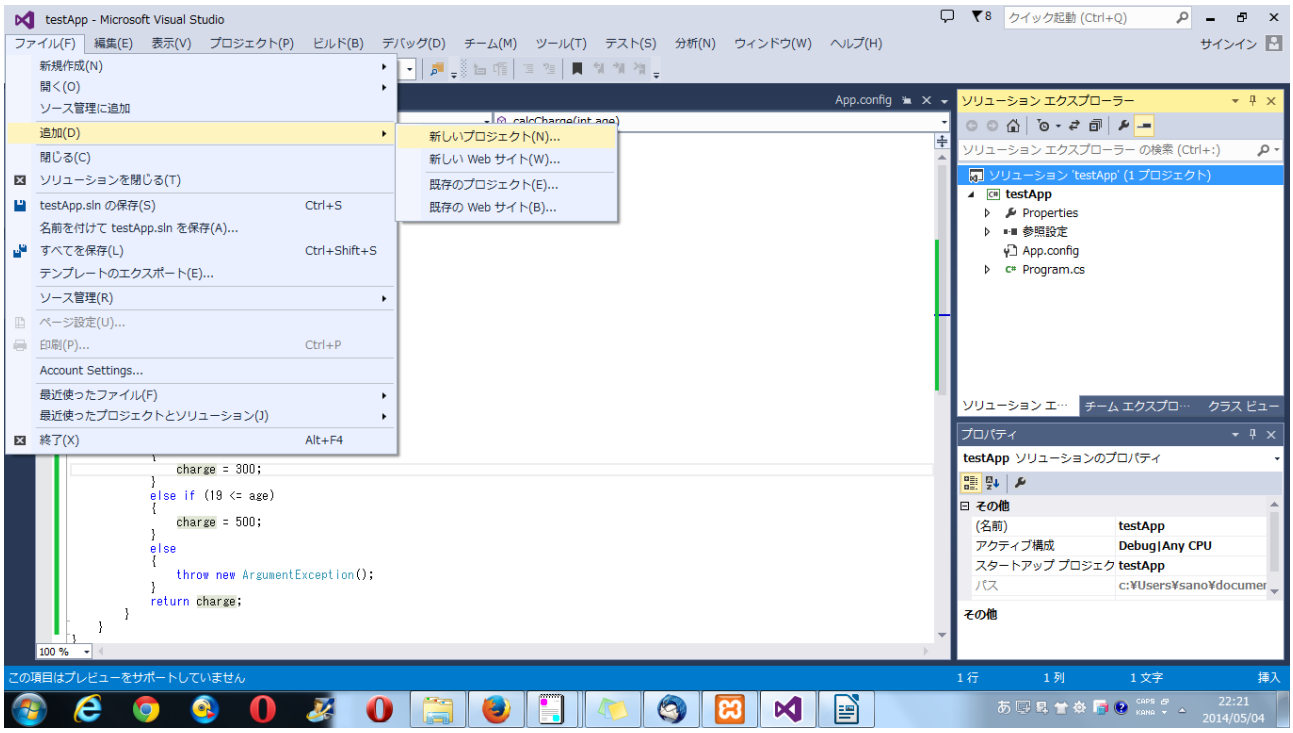
Visual Studio 2013 －単体テスト作成方法－

１、単体テスト対象のクラス作成

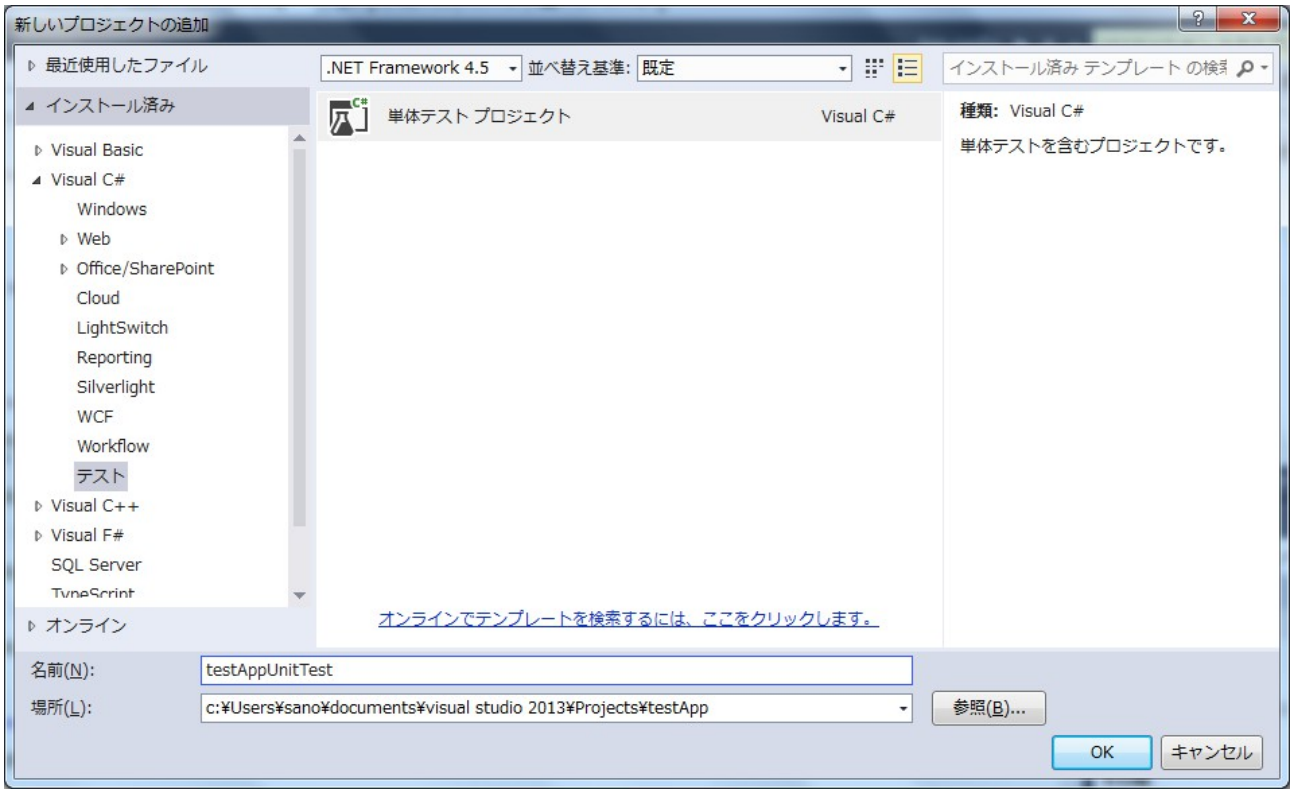
```
1   using System;
2
3   namespace TestApp
4   {
5       public static class Program
6       {
7           static void Main(string[] args)
8           {
9           }
10      }
11      public class Charge
12      {
13          public int calcCharge(int age)
14          {
15              int charge = 0;
16              if (0 <= age && age < 7)
17              {
18                  charge = 0;
19              }
20              else if (7 <= age && age < 19)
21              {
22                  charge = 300;
23              }
24              else if (19 <= age)
25              {
26                  charge = 500;
27              }
28              else
29              {
30                  throw new ArgumentException();
31              }
32              return charge;
33          }
34          private bool haiteiMethod(int p)
35          {
36              if (p == 0)
37              {
38                  return true;
39              }
40              else
41              {
42                  return false;
43              }
44          }
45      }
46  }
47
```

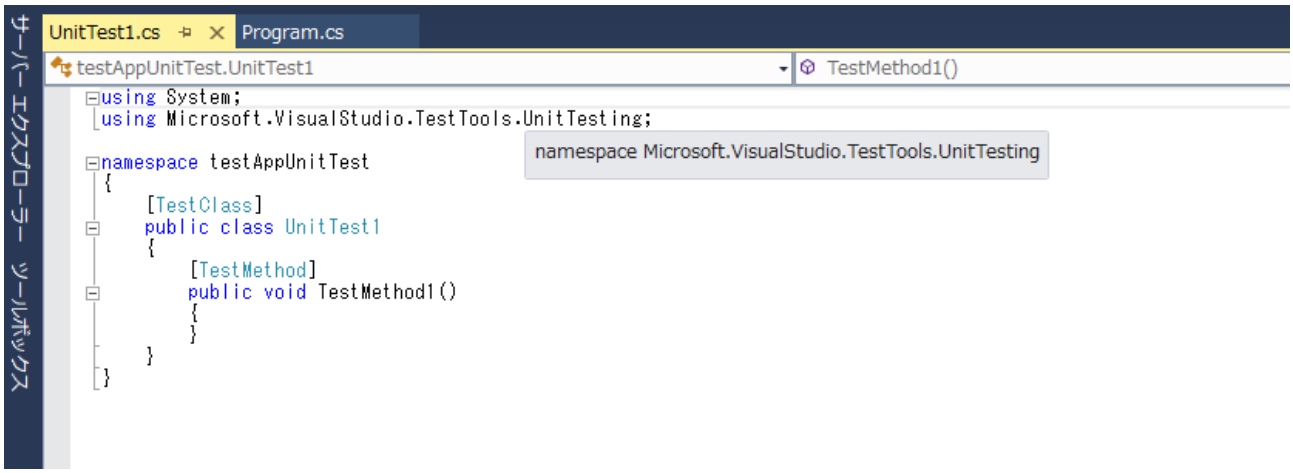
2、テスト対象プロジェクトを開いている状態で、

メニューのファイル→追加→新しいプロジェクト→テスト→単体テストプロジェクトを選択

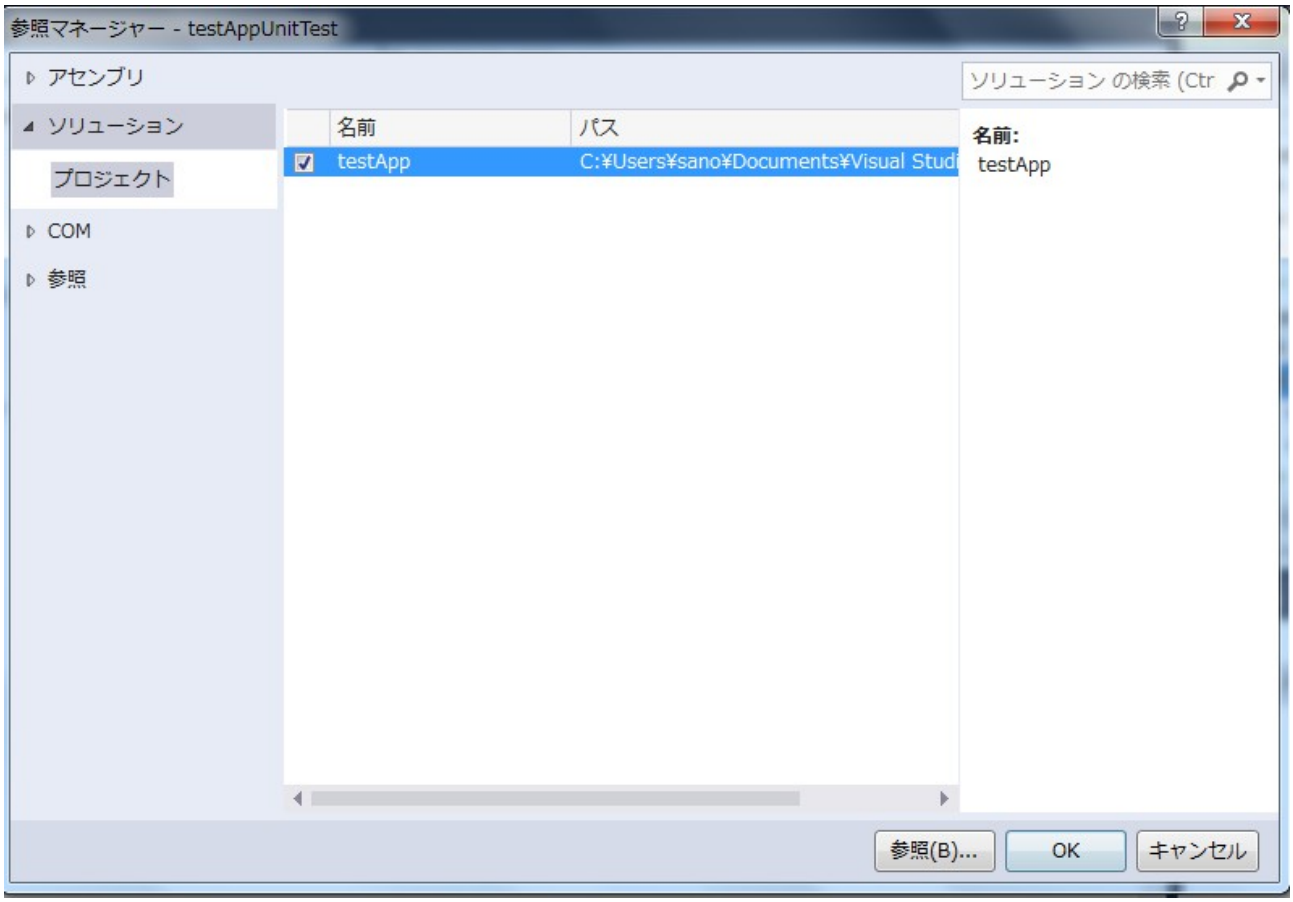


3、名前に「TestAppUnitTest」と入力して OK ボタンをクリック




サーバー エクスプローラー ツールボックス
UnitTest1.cs
Program.cs
testAppUnitTest.UnitTest1
TestMethod1()
using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;
namespace Microsoft.VisualStudio.TestTools.UnitTesting
namespace testAppUnitTest
{
[TestClass]
public class UnitTest1
{
[TestMethod]
public void TestMethod1()
{
}
}
}

5、単体テストクラスの参照設定に「TestApp」を追加

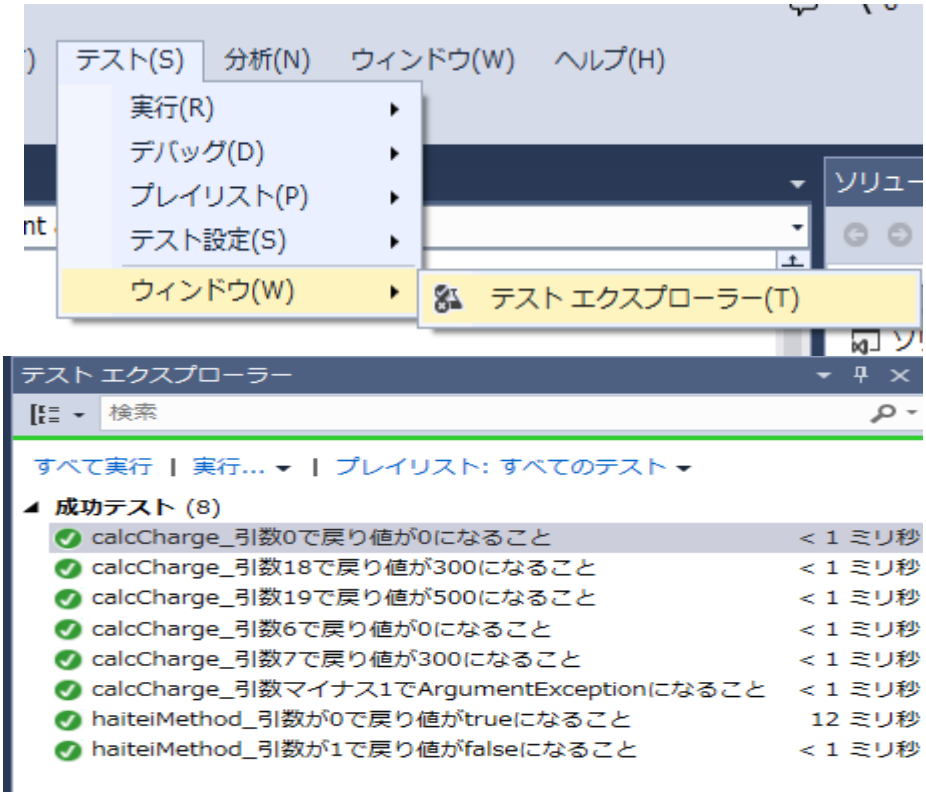


6、単体テストクラスのコード

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System;
using System.Globalization;
using TestApp;

namespace TestAppUnitTest
{
    [TestClass]
    public class UnitTest1
    {
        private CultureInfo cultureInfo;

        private TestContext testContextInstance;
        /// <summary>
        ///Gets or sets the test context which provides
        ///information about and functionality for the current test run.
        ///</summary>
        public TestContext TestContext
        {
            get
            {
                return testContextInstance;
            }
            set
            {
                testContextInstance = value;
            }
        }

        #region 追加のテスト属性
        //
        //テストを作成するときに、次の追加属性を使用することができます:
        //
        //クラスの最初のテストを実行する前にコードを実行するには、ClassInitialize を使用
        //[ClassInitialize()]
        //public static void MyClassInitialize(TestContext testContext)
        //{
        //}
        //
        //クラスのすべてのテストを実行した後にコードを実行するには、ClassCleanup を使用
        //[ClassCleanup()]
        //public static void MyClassCleanup()
        //{
        //}
        //
        //各テストを実行する前にコードを実行するには、TestInitialize を使用
        [TestInitialize()]
        public void MyTestInitialize()
        {
            this.cultureInfo = new CultureInfo("ja-JP", true);
        }
        //
        //各テストを実行した後にコードを実行するには、TestCleanup を使用
        //[TestCleanup()]
        //public void MyTestCleanup()
        //{
        //}
        //
        #endregion

        #region haiteiMethodのテスト
        [TestMethod]
        public void haiteiMethod_引数が0で戻り値がtrueになること()
        {
            PrivateObject po = new PrivateObject(typeof(TestApp.Charge), null);
            var actual_ = (bool) po.Invoke("haiteiMethod", new object[]{ 0 }, this.cultureInfo);
```

```
67              Assert.AreEqual(actual, true);
68          }
69
70          [TestMethod]
71          public void haiteiMethod_引数が1で戻り値がfalseになること()
72          {
73              PrivateObject po = new PrivateObject(typeof(TestApp.Charge), null);
74              var actual = (bool)po.Invoke("haiteiMethod", new object[] { 1 }, this.cultureInfo);
75
76              Assert.AreEqual(false, actual);
77          }
78          #endregion
79
80          #region calcChargeのテスト
81          [TestMethod]
82          public void calcCharge_引数0で戻り値が0になること()
83          {
84              var charge = new Charge();
85              Assert.AreEqual(charge.calcCharge(0), 0);
86          }
87
88          [TestMethod]
89          public void calcCharge_引数6で戻り値が0になること()
90          {
91              var charge = new Charge();
92              Assert.AreEqual(charge.calcCharge(6), 0);
93          }
94
95          [TestMethod]
96          public void calcCharge_引数7で戻り値が300になること()
97          {
98              var charge = new Charge();
99              Assert.AreEqual(charge.calcCharge(7), 300);
100         }
101
102         [TestMethod]
103         public void calcCharge_引数18で戻り値が300になること()
104         {
105             var charge = new Charge();
106             Assert.AreEqual(charge.calcCharge(18), 300);
107         }
108
109         [TestMethod]
110         public void calcCharge_引数19で戻り値が500になること()
111         {
112             var charge = new Charge();
113             Assert.AreEqual(charge.calcCharge(19), 500);
114         }
115
116         [TestMethod]
117         public void calcCharge_引数マイナス1でArgumentExceptionになること()
118         {
119             try
120             {
121                 var charge = new Charge();
122                 charge.calcCharge(-1);
123             }
124             catch (ArgumentException ex)
125             {
126                 Assert.IsTrue(true);
127                 return;
128             }
129             Assert.Fail("ArgumentExceptionが発生しませんでした");
130         }
131         #endregion calcCharge
132     }
133 }
```

7、単体テストクラスの実行



<参考情報>

単体テストを使用したコードの検証

http://msdn.microsoft.com/ja-jp/library/dd264975.aspx

[VS2010] Visual Studio の単体テストでプライベートなメソッドやフィールドにアクセスする

http://handcraft.blogsite.org/Memo/Article/Archives/294

C# で private メソッドを呼んでみる

http://normalian.hatenablog.com/entry/20090124/1232782230

Visual Studio で作る単体テスト、UI 操作の自動実行

http://codezine.jp/article/detail/6273

[C#][テスト] Visual Studio で単体テスト

http://d.hatena.ne.jp/superlightbrothers/20081221/1229879705

Visual Studio 2012 での単体テストの作り方

http://gotemba.blogspot.jp/2013/09/visual-studio-2012_30.html

Visual Studio 2012　右クリックで単体テストの追加　コンテキストメニューを表示する。

http://sugalog.sugasaki.com/2013/02/22/visual-studio-2012　右クリックで単体テストの追加　コンテキ/