

世界はオブジェクトの海に浮かぶ

.NET Framework
で楽しむ
オブジェクト指向

第7回 デバッグしましょ！

ΕΠΙΣΤΗΜΗ
えびすてーめー

夏休み勉強会

この連載でも何度か紹介している、マイクロソフトの学生さん応援大盤振

Level

1 2 3 4 5

Technology Tools

- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access
- ASP.NET
- Other:
↓
NUnit 2.2

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、
<http://www.shoetisha.com/mag/windev/>
からダウンロード可能です。

る舞いパッケージVS.NET theSpoke Premium。このtheSpokeメンバー向けのイベント「夏休み勉強会」が大坂/東京のマイクロソフトセミナールームで開催され、僕は東京会場へ冷やかに行ってきました。

頃は8月下旬夏休みもあとわずか、加えて当日は台風が迫っていてヘタすりゃ帰れないかも……という劣悪なコンディションにもかかわらず十数名のtheSpokeメンバーが集まってくれました。夏休み勉強会でのお勉強のお題は「デバッグ入門」。Visual Studioのデバッグで何が出来るか/どう使うか、のキホンを紹介し、プログラミングに役立ててもらおうという次第です。ブレイクポイントの当て方と条件ブレイク（ここを何回通過したか、とか）、そしてステップ実行（ステップイン/ステップオーバー/ステップアウト）の使い分け、そしてコールスタック/変数のウォッチなど、デバッグの基本的な使い方をデモを交えて解説していただきま

した。うんうん、たしかに「デバッガの使い方」について（ほんの入り口とはいえ）きちんと説明を受けるって機会はなかなかないように思います。夏休み勉強会のお題としてはとてもよかったんじゃないかしら。

さてさてコンパイラはコンパイルしてくれるしリンクはリンクしてくれます。だけど困ったことにデバッガはデバッグしてくれません。デバッガはデバッグのお手伝いをしてくれるものなんですよね。さらに言えば、デバッガはバグを見つけてはくれませんし、「あれ、なんかへんじゃねえ？」と助言してくれるわけでもありません。プログラムは思ったとおりに動きません。作ったとおりに動きます。計算機は作られたものを真面目に忠実に（愚直に）実行しているだけで、僕らが思ったことを知ってるわけじゃない。だから「なんかへんじゃねえ？」——つまり思ったのとは異なる動きであることを助言したくてもできないのは無理からぬことですわね。デバ

ッガは「なんかヘンじゃねえ？」と気づいた“後”に使われるものです。体の異常に気づかない限り病院に行って診察／治療を受けることはないでしょう。手遅れになるまえに、異常に気づくことが大切です。

System.Diagnostics. Debug

夏休み勉強会では.NET FrameworkのSystem.Diagnostics名前空間に用意されたDebugの使い方を少しだけ紹介してくれました。このDebug、ぶっちゃけて言えばその昔ロクなデバッガのない時代によくやってた“プリントデバッグ”をサポートするクラスと思ってもらっていいかと。“プリントデバッグ”とは、コードのあちこちにprint文、.NETでいえばConsole.WriteLineを埋め込んで実行し、「ココを通過した」ことや要素所での変数の値を確認するっていう原始的ながらかなりの効果が望めるデバッグ手法です（僕は今でもしょっちゅうやります）。あちこちにprint文をバラ撒いたプログラムを実行すると当然のことながら埋め込まれたメッセージが画面にぶわーっと出てきます。大量に出力される場合はテキストファイルに落としておいて、終了後にそいつを眺めて怪しい挙動をしていないか検証するわけ。

ではDebugの簡単な使い方をご紹介します。

DebugはSystem.Diagnostics名前空間にあるので、

C#なら

```
using System.Diagnostics;
```

VBなら

```
Imports System.Diagnostics
```

をコードのアタマに書き加えます。あとは簡単、プリントしたい箇所に、

```
Debug.WriteLine(メッセージ)
```

すればいい。改行したくないなら、

```
Debug.Write(メッセージ)
```

です。コンソールアプリケーションならこんな感じ。

```
using System;
using System.Diagnostics;

class Sample {

    [STAThread]
    public static void Main() {
        Debug.WriteLine("START"); // ココでプリント
        Console.WriteLine("Hello, world!");
        Debug.WriteLine("END"); // ココでもプリント
    }
}
```

これをDebugモードでコンパイルし、IDE内でデバッグ実行すると、IDEの「出力」ウィンドウにSTARTとENDが現われるはず（図1）。

変数の値をプリント、たとえば変数*i*を書きたいならば、

```
Debug.WriteLine("i={0}", i);
```

ごめんなさい嘘つきました、書式指定はできません。次のようにstring.Formatを使うことになりそうです。

```
Debug.WriteLine(string.Format("i={0}", i));
```

WriteIf/WiteLineIfメソッドを使えば条件に応じてプリントするか否かを定めることもできます。第1引数で与えた条件がtrueのときに限りプリントします。

```
// 0, 10, 20, ... のときにプリント
for (int i = 0; i < 100; ++i) {
    Debug.WriteLineIf(i % 10 == 0,
```

図1：STARTとENDが現われた！

