

ΕΠΙΣΤΗΜΗ

オブジェクト 指向的 夜話

★ C#から見た
★ .NET Framework

第 4 回

コレクションで遊ぼう

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:
 - Visual C++ .NET
 - 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。

¥FAKE_STL
IListを操作対象とする関数群
¥FAKE_STL_TEST
上記サンプル用テストコード

ちょっと宣伝

ワタクシゴトで恐縮ですが、C#を中心とした、.NETな話題を取り扱うML (mailing list) を立ち上げています。

<http://www.freeml.com/info/csharp@freeml.com>

今年の4月、C#も.NETもまだまだシロートの僕が本誌での連載が決まり、何らかの情報源になりはせんかといういささかヨコシマな目的で慌てて立ち上げたMLです。会員数は約500名、メールの流量はまださほど多くはないのですが、このギョーカイでは名の知れたあの方やこの方が参加してくださっていて、.NETオンチな僕には心強い限り。僕のトーシロな質問に (おそらく) 呆れながらも親切に答えてくださってます。ありがたやありがたや。C++のバックボーン (?) があるか

ら右も左もわからんというほどのことはないのですが、時折見せるC++との決定的な違いに途惑うこともしばしばです。C#あるいは.NETに関する限り、まだまだビギナーに毛が生えた程度の僕ですが、引き続きよろしくお願ひ致します。

ところで 今月のお題は?

で、プログラミングを始めたばかりのホントのビギナーに訊かれることがあります。

「プログラムって何ですか?」

これに答えるのはなかなか難しいように思います。やりたいことを計算機の解るコトバで表現したもの、とでも言いましょか。そこには何をやりたいか (what) からどうやってやらせるか (how) への変換が伴います。それこそがプログラマの大事なお仕事のひとつですわね。で、“どうやってやらせるか”

にはいくつかの定石というかパターンがあって、プログラムは要求と目的に応じて頭の中にある多種多様な多くのパターンの中から最適なものを選び出し、必要に応じてカスタマイズしながらプログラムを組み上げていくわけです。とはいえなんだかんだいいながらもキホンはひとつ、“吸って、溜めて、吐く”の繰り返しです。ファイルやメモリからデータを“吸って”、適当な形で“溜めて”おき、そして画面やらプリンタやらファイルやらに“吐く”、それを繰り返すのがプログラムだ、といささか強引な説明をしています。

今回のお題はその“吸う／溜める／吐く”のうちの“溜める”お話、データの集合を扱うお話です。

配列の限界

C++、Java などなど大抵の言語には配列があらかじめ言語仕様の一部として用意されています。配列はいわば大きさの決まった箱です。あらかじめ要素をいくつ入れかを決めておき、その個数を上限とした箱を作ります。

C/C++だと、こんな。

```
int int_array[5];
```

C#/Javaだと、こう。

```
int[] int_array = new int[5];
```

こうすればintが5つ入る箱、int_arrayが用意されます(実はこの2つの例、そして字面は同じに見えるC#とJavaでの配列、その意味するところはかなり異なるのですが)。

プログラミングに手を染めて間もない頃は、この配列にはさんざん世話になることでしょう。データの集合として、これほどお手軽なものはありませんからね。しかししばらくすると配列ではどうにもならない局面に出くわします。たとえばファイルから単語を吸って、配列に溜めて、アルファベット順に並べ替えて画面に吐く。

そんなプログラムを作るとしましょう。配列は大きさの決まった箱です。一体どれくらいの大きさの箱を用意すればいいのか、ファイルを読んでみないことにはわかりません。だから無理矢理配列でやるには一度ファイルを読んで箱の大きさを見積もり、その大きさの箱を用意して再度読み込むといった処理になります。ファイルであれば何度でも読み出せますが、電話回線やネットワークから流れ込む場合はそうはいきません。アチラから流れてくるのは一度きりですからね。そんな場合は適当な大きさの箱をとりあえず用意し、ファイルから読み込んだ単語を格納します。そして箱が一杯になったらちょっと大きめの新たな箱を作って移し替え……、を繰り返せばいいでしょう。いずれにせよ面倒な処理を行なわなければならないかもしれません。配列は必要な大きさがあらかじめわかっているならお手軽です。が、どれほどの大きさが必要かあらかじめ知ることができないときは、面倒(そして退屈)な処理をプログラマに要求します。メモリの許す限り、要素数の増加に応じて勝手に大きくなってくれる箱が欲しくなります。

System.Collections

.NET Frameworkには、配列には真似のできない“勝手に大きくなってくれる箱”が5種類、名前空間System.Collectionsに用意されています。

- ArrayList
- Stack
- Queue
- Hashtable
- SortedList

これらはすべて、あらゆるクラスの原点となるobjectすなわちSystem.Objectから導出されています。また、名前空間System.Collectionsには要素の集合(つまり箱)が満たさねばならない要件を定めたインターフェイス(ICollection、IEnumerable、IList、IDictionary)が定