επιστημηの

オブジェクト指向的夜話

C#から見た.NET Framework

最終回

Genericsで線形検索

επιστημη
えぴすてーめー

Technology Tools

- ❑ Visual Basic .NET
- ☑ Visual C# .NET
- ❑ SQL Server 2000
- ❑ Oracle 9i
- ❑ Access 2002
- ❑ ASP.NET
- ❑ Internet Information Services
- ☑ Other:
  .NET Framework 2.0

Level

★★★☆☆

Samples

C#2.0の魅力

僕のマシンには、電源入れるや立ち上げっぱなしのVisual Studio .NET 2003と共にVisual Studio 2005 βがインストールされています。お仕事にはもっぱらVS2003が主流です(そりゃそうよ、VS2005はβなんだもの)。なのでVS2005 βはインストールしてはあるものの、これまでほとんど封印状態でした。MSDNの内容もそのほとんどは英語のまま。そろそろβ2がお目見えとの噂も耳にしています。β2にいきなり触れてあまりの違いにうろたえるのもみっともないので、新しいVisual Studio、というかC#2.0の雰囲気を掴んでおくためにもそろそろ真面目に蓋を開けてもよかろうか。と思ったのが新年を迎えて鏡開きも済ませたころです。それ以前にもC# 2.0には少しは触っていたのですが、ちょいと奥に踏み込んで、2.0だとこんな面白いことができるんだ、というトピックを僕なりに探してみました。

Generics

言語仕様としてのC#1.xと2.0とのもっとも大きな相違はやはり“Generics”でしょう。C++ではtemplateで実現されたGenericsは90年代半ばには主要なコンパイラに実装され実用に供されていましたが、Javaでは昨年リリースされたJDK5.0、C#(.NET)では今年リリースの2.0でお披露目となります。

Genericsとは“差し替え可能な型”です。たとえばスタック(Stack)、要素をいくつか投げ入れ(Push)、そして取り出す(Pop)と、投げ入れたのとは逆順、つまり最後に投げ入れた要素から先に取り出されます。.NET Frameworkの名前空間「System.Collections」に「Stack」

が定義されています。

```
using System.Collections;

class trial {
 public static void Main() {
  Stack s = new Stack();
  s.Push("first");
  s.Push("second");
  s.Push("last");
  while ( s.Count != 0 ) {
   string item = (string)s.Pop();
   System.Console.WriteLine(item);
  }
 }
}
```

便利です。だけどちょっと（かなり）危ういのです。というのも、Stackに積むことのできる要素はobject、つまり何を投げ込もうがエラーにはなりません。

逆にStackから取り出すとき、メソッドPop()の返り値はobjectですから、必要に応じて目的の型にキャストしなければなりません。

```
s.Push("first");
s.Push(2);
s.Push(12.5);
```

とやったって涼しい顔してコンパイルしてくれますが、それに続く、

```
string item = (string)s.Pop();
```

ここで例外（実行時エラー）、

```
ハンドルされていない例外 :
 System.InvalidCastException:
 指定されたキャストは有効ではありません。
at trial.Main()
```

が発生します。こんなエラーはできることなら実行時ではなくコンパイル時に検出してほしい。つまりPush()できる要素はstringに限定したいし、Pop()したらobjectではなくstringが返ってきて欲しい。ならばそんなメンバ関数、void Push(string item)とかstring Pop()とかを定義したstring専用スタック「StackForString」を作ればいい。だけどその調子でint専用スタック「StackForInt」だのfloat専用スタック「StackForFloat」だのといちいち個別に作るのはバカバカしい。XX専用スタック「StackForXX」をひとつだけ作っておいて、XXを好きに差し替えられたらいいのに……。それがGenericsです。

.NET Framework 2.0では従来の名前空間「System.Collections」とは別にGenericsを駆使したコレクションが名前空間「System.Collections.Generic」に用意されています。ここにあるGenericなStackを使えば上記コードは、

```
using System.Collections.Generic;

class trial {
 public static void Main() {
  // stringを要素とするスタック
  Stack<string> s = new Stack<string>();
  // スタックに積んで
  s.Push("first");
  s.Push("second");
  s.Push("last");
  // 逆順に取り出す
  while ( s.Count != 0 ) {
   string item = s.Pop();   // キャスト不要!
   System.Console.WriteLine(item);
  }
 }
}
```

となりますし、うっかりstring以外のオブジェクトをPush()すると、

```
generic_stack_trial.cs(8,12):
 error CS1503: 引数 '1' :
 'int' から 'string' に変換できません。
```

のように、コンパイル時にエラーとしてくれます。

余談ですが.NET Framework 2.0では従来の「System.Collections.Stack」とは別に「System.Collections.Generics.Stack<T>」を提供しています（他のコレクションも同様）が、Javaでは「java.util.Stack<T>」一本となり、代わりに従来のStackがコードに現われたら「Stack<Object>」として扱います。どちらがいいかは議論のわかれるところでしょうけど。