

επιστημηの

オブジェクト 指向的 夜話

★ C#から見た
★ .NET Framework

第 6 回

電卓を作るのだ

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:

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。

¥STACK
スタックプロジェクト

¥OPERATION
演算とパーサープロジェクト

¥OPERATION_TRIAL
上記2プロジェクトを使用する本体

初恋の相手は

僕が初めてプログラミングの世界に足を踏み入れたのは高校2年生の夏。科学雑誌の広告に釘付けになりました。そこにあったのはHP (Hewlett-Packard) の電卓：HP-33Eです (<http://www.hpmuseum.org/hp33.htm>)。広告にはプログラミングのできる関数電卓とありました。僕はコイツに一目惚れ、小遣い全額はたいて電車で揺られて3時間、HP電卓を取り扱うお店に飛び込んで、

「くださいっ!」

お店のおじさんは“高校生がこんなの買うか?”って顔してました。家にたどり着いたときポケットに30円残っていたのを今でも覚えています。電卓はいまでこそ液晶表示ですけど、このHP-33Eは真っ赤なLEDですごく綺麗でした(ずいぶん前にぶっ壊れちゃったんです

……)。

それからというもの、僕はプログラミングの魅力に取り憑かれました。プログラムといってもたった49ステップで、しかも電源切るとせっかくのコードを忘れてしまうトホホな奴だったけど、自分の思うがままに真っ赤なLEDをチラチラさせながら律儀に仕事をこなす姿に夢中になりました。あの時のドキメキがなかったら僕は今頃何していただろう、と思えるほど運命的な出会いでしたね。それ以来、電卓はHPしか買ったことがありません。33Eを手始めにスペースシャトルにも乗った41C、そして11C、15C、16C、41CX、28S、32Sなどなど……。今でも僕の机の上、カバンの中にはHPの電卓たちが転がっています。

ってなわけで、今月のお題は“電卓”です。

HPの電卓って……

HPの電卓は計算時の入力順が普通の電卓とは違うんです。「1と2を加える」とき、普通の電卓では

1 + 2 =

と入力します。演算子が数の間に挟まるので中置記法 (infix-notation) と呼ぶのだそう。かたやHP電卓では、

1 2 + (1に2を足す)

と入力します。実際には1と2の間に [ENTER] キーで数を区切りますが (これがないと12になっちゃう)。

ポーランドの論理学者Lukasiewicz (ルカシヴィツ/ウカシェーヴィッチ) が考案したポーランド記法では演算子を数の前に置きます。たとえば、

(1 + 2) * 3

を、

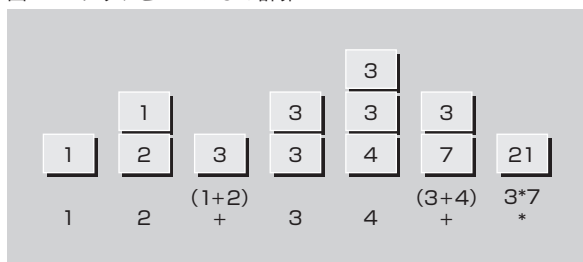
* + 1 2 3

と表記しますが、HP電卓では、

1 2 + 3 *

つまり、「1に2を足し、3を掛ける」となります。ポー

図1：スタックとPRNによる計算



ランド記法とは逆に、演算子が後に来ることからRPN (Reverse Polish Notation : 逆ポーランド記法) と呼ばれます。というか、HP社がポーランド記法の考案者Lukasiewiczに敬意を表して逆ポーランド記法と命名したらしいのですが。

で、RPN電卓には [=] キーがありません。演算子を入力した途端に計算しちゃいますから。さらに、演算子に優先順位がありません。演算子を入力した途端に計算しちゃいますから。だから演算の優先順位を指示するカッコもありません。

(1 + 2) * (3 + 4)

は、

1 2 + 3 4 + *

「1に2を足し、3に4を足し、掛ける」となります。

この計算式入力法は計算機ととっても馴染みがいいのです。スタックひとつで計算機が作れるからです。つまり、

- 数はそのままスタックに積む (push)。
- 単項演算子は数をスタックから取り出し (pop)、演算を施してスタックに積む。
- 二項演算子はスタックから数を2つ取り出し、演算を施してスタックに積む。

これだけのルールで計算できちゃいます。「1 2 + 3 4 + *」を計算するようすを図1に示します (スタックの先頭が下に描かれています)。

というわけで、スタック計算機、C#で作っちゃいましょう。基本操作は数のpushとpopだけ、演算は、たとえば足し算なら、

```
public void add() {
    double x = pop();
    double y = pop();
    push(x+y);
}
```