

世界オブジェクトの は海に浮かぶ

.NET Framework
で楽しむ
オブジェクト指向

新連載

オブジェクト指向のおさらい

επιστημη
えびすてーめー

Level				
1	2	3	4	5

Technology Tools
<input checked="" type="checkbox"/> Visual Basic
<input checked="" type="checkbox"/> Visual C#
<input type="checkbox"/> Visual C++
<input type="checkbox"/> SQL Server
<input type="checkbox"/> Oracle
<input type="checkbox"/> Access 2002
<input type="checkbox"/> ASP.NET
<input type="checkbox"/> Other:

Samples
<small>・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.shoeisha.com/mag/windev/からダウンロード可能です。</small>

オブジェクト 指向仕切り直し

僕と翔泳社さんの技術誌でのお付き合いは1993年の夏、「dotNETマガジン」の前身である「VB (Visual Basic) マガジン」のさらに先輩格にあたる「DDJJ (Dr.Dobb's Journal JAPAN: ドクター・ドブズ・ジャーナル日本版)」にC++ネタを寄稿したのが始まりでしょうか。それからほどなくしてDDJJに「επιστημη η η のオブジェクト指向的日常」を連載することになり、それが5年以上も続く長期連載となりました。DDJJに少し遅れて1995年、「VBマガジン」創刊。DDJJは残念ながら1999年の春に幕引きとなりました。連載がぷつぷり切れたのが惜しかったけど、なんせ畑が違うから「VBマガジン」とはご縁がありませんでした。そうこうしているうちにJava、そしてC# (.NET) と「どこからどうみてもC++をお手本にしていると

しか思えない」新言語が台頭して参ります。「dotNETマガジン」と誌名が変わってからも僕の興味対象は依然としてC++一本!で、「dotNETマガジン」という誌名には絡みが浅くて遠慮してました。が、いつまでもC++一辺倒というのもアレなので暇を見つけてはJava/C#にも少しづつ手を染めており、そろそろ潮時(?) かと「dotNETマガジン」に顔を出したという次第。C#および.NET Frameworkによるオブジェクト指向ネタで毎月編集サマにご迷惑をお掛けしておりました(編集サマ、筋金入りの遅筆でホントにごめんなさい)。「.NET」マガジンなもんだからC++に絡むネタを意識的に遠ざけてきました。ホントは同機能のコード片をC++/Java/C#の3つの言語で実装する、なんて類のをやりたかったんですけどね。

そしてこのたび「Windows Developer マガジン」と誌名が改まり、これで守備範囲が広がったことに

なりますか。Windowsベースの開発ってことならば、JavaでもC++でも躊躇することはないわけで、毎月ネタの発掘に四苦八苦していたのが少しは楽になり、ひいては編集部各位の胃痛もいかにばかりか軽減されることとじます（なんてね）。

ともあれ「Windows Developer マガジン」連載第一回ですから、気分も新たに“オブジェクト指向”のおさらいをやっておこうかと。

データの抽象化

人間は考える生き物です。かたや計算機は少なくとも現時点では、考えると言うにはまだ遠くおよびません。考えているかのような振る舞いを見せることもあります、圧倒的な計算速度と確かな記憶力を駆使して「考えているように見えている」のであり、さらにその中身は計算機がある日突然思いついたわけじゃなく、人間が考え抜いたコードを計算機に食わせ、計算機はそのとおりに動いているだけ。プログラムは人間の思考の産物です。昔々のプログラミング言語は計算機寄り、つまり計算機に都合のいい構造のものがほとんどでした。プログラマは計算機に歩み寄り、計算機が簡単に理解できる言語で計算機に語りかけておりました。それが次第に人間にとってわかりやすい、あるいは書きやすい言語へと進化してきました。

ヒトは多くのことを同時に理解し把握するのは苦手です。だから大きな問題は一旦それをいくつかの小さい単位にまとめ、それぞれをひとつずつ解決したほうがずっと楽です。だからプログラミング言語も人間側に歩み寄り、いろんな単位で“まとめる”ことのできる機能を数多く提供してきました。

たとえばとある高校生の定期試験の成績、理科／英語／数学のテストの得点があって、合計と平均を求めるならばC#では、

```
class Exam {
    public static void Main() {
        int sci = ..... // 理科の得点
        int mat = ..... // 数学の得点
```

```
        int eng = ..... // 英語の得点
        int sum = sci + mat + eng; // 合計
        double avg = sum / 3.0; // 平均
        System.Console.WriteLine(
            "合計={0} 平均={1}", sum, avg);
    }
}
```

こんなところでしょうか。高校生が2人いるなら、

```
class Exam {
    public static void Main() {
        // ひとりめ
        int sci0 = ..... // 理科の得点
        int mat0 = ..... // 数学の得点
        int eng0 = ..... // 英語の得点
        int sum0 = sci0 + mat0 + eng0; // 合計
        double avg0 = sum0 / 3.0; // 平均
        System.Console.WriteLine(
            "合計={0} 平均={1}", sum0, avg0);

        // ふたりめ
        int sci1 = ..... // 理科の得点
        int mat1 = ..... // 数学の得点
        int eng1 = ..... // 英語の得点
        int sum1 = sci1 + mat1 + eng1; // 合計
        double avg1 = sum1 / 3.0; // 平均
        System.Console.WriteLine(
            "合計={0} 平均={1}", sum1, avg1);
    }
}
```

どうということはありません。コピー&ペーストで複製を作り、変数名の重複を解消すればいい。しかしこれでは処理すべきデータが増えてくると近いうちに限界が訪れます。3年B組35名の成績をこの延長でやっちゃうとコードは300行に膨れ上がります。全校生徒の集計ともなると……。なので配列と繰り返しによって“まとめる”わけですね。

```
class Exam {
    public static void Main() {
        int N = 35; // 生徒数
        int[] sci = new int[N]; // 理科の得点
        int[] mat = new int[N]; // 数学の得点
        int[] eng = new int[N]; // 英語の得点
        ; // 配列の内容（各人の得点）を埋める
        for (int i = 0; i < N; ++i) {
            int sum = sci[i] + mat[i] + eng[i];
            double avg = sum / 3.0;
            System.Console.WriteLine(
                "合計={0} 平均={1}", sum, avg);
        }
    }
}
```