

まつもとゆきひろ

The World
of Code

コード

の

世界

スーパー・
プログラマになる
14の思考法

まつもとゆきひろ 著

日経Linux編



日経BP社

ーンの多くは将来の変更に対処するためOCPに従うことができるように設計されています。

* * *

本節ではOCPの観点からデザイン・パターンを眺め直してみました。継承によって実現されたOCPが、デザイン・パターンを使うことでより強力になるのです。変化に強い道具として活用できることがお分かりいただけたと思います。

実は、OCPとデザイン・パターンを組み合わせるというのは、私のオリジナルのアイデアではありません。私の知る限り、国内でこの2つを組み合わせで解説を行ったのは、石井勝氏です。石井さんによる解説はWebページ (<http://www.morijp.com/masarl/homepage3.nifty.com/masarl/article/dp-ocp-2.html>) から読むことができます。

語彙と共通言語の重要性

『デザイン・パターン』を扱った書籍が出た時の最初の印象は「おおげさに話題になっているわりには当たり前の内容だなあ」というものでした。書籍『デザイン・パターン』にカタログされている23のパターンのうち、多くは日常的に使うことが多い珍しくないパターンですし、またC++やJavaのような静的型の言語では有効でも、SmalltalkやRubyのような言語ではたいてい意味のないパターンも多く含まれています。

しかし、しばらくしてから気がつきました。『デザイン・パターン』の本質は今までに使ったこともない新しいパターンを紹介することではなく、しばしば使われる実用的なパターンに適切な名前を提供することによって、デザインにおける語彙を提供することにあるのだと。

人間は思考のために言語を使います。人間は言葉を介在しないと系統だって思考することができませんし、言葉を使わないと他人とコミュニケーションすることもできません。人間の存在と関係にはいつも言葉がかかわっています。

旧約聖書に登場する「バベルの塔」では、人々がお互いに言葉が通じなくなって塔の建設は中断され、人々は散り散りになってしまう様子が描かれていま

す。言葉はそれほど大切なのです。

聞くとところによるとアラスカやカナダに住むアイヌの人たちの言語には雪と氷を表現する単語が80種類もあるそうです。それだけ微妙な違いを区別する機会があるのですね。日本語でも雨を表現する言葉がたくさんありますね。霖雨、五月雨、時雨、夕立...

デザイン・パターンもそれと同じです。ソフトウェア開発では、さまざまな局面で用いることができるクラス構成やイディオムが存在しますが、それらに対して適切な名前がついていなければ、多くの開発者はその存在さえ意識できません。デザイン・パターンとはそのような語彙を集めてカタログ化する試みなのです。ですから、書籍で紹介されていた23のパターンは始まりにすぎません。最終的にはもっともっと多くのパターンを収録したデザイン・パターン・カタログが登場するべきなのです。

残念ながらそのようなカタログが登場する気配はあまりないようですが、もしかするとまだ人類はすでにもっている23のパターンさえ有効に活用できておらず、さらなるパターンを追求できるほどにはまだ進化していないのかもしれない。