

インターフェースとしての言語のデザイン：Ruby

まつもと ゆきひろ
Yukihiro MATSUMOTO

(株)ネットワーク応用通信研究所
〒690-0826 松江市学園南2-12-5 HOYOパークサイドビル2階
E-mail: matz@ruby-lang.org

プログラミング言語とは、プログラムを記述するための記法であり、その目的はコンピュータに行わせる処理の手順を記述することである。そのため、プログラミング言語にはコンピュータが理解(処理)できる程度の単純さと曖昧性のなさが求められる。

しかし、プログラミング言語は人間が意図(やりたいこと)をコンピュータに伝達する手段である。つまり、プログラミング言語は本質的にマン・マシン・インターフェースの一種であると考えることができる。事実、プログラミング言語の歴史はコンピュータ(CPU)が直接処理できるマシン語からよりシンボリックに記述できるアセンブラー、数式を直接記述できるFORTRANとよりコンピュータ側のコストをかけても人間側の開発コストを低減するような方向に進化している。コンピュータ側の性能の向上により、人間側のコストを下げるためにコンピュータの能力を「浪費」することが妥当となってきたことがその背景にある。

しかし、プログラミング言語の評価基準については、いまだに

- その言語から利用できるライブラリなどの資産
- その言語によるプログラムの性能

が重視され、その言語がインターフェースとして優れているかについては、比較的軽視されている部分が見受けられる。本論では、ある言語が優れたインターフェースであるということはどのようなことかについて考察を行い、良い言語インターフェースの評価についての指針をふたつ述べる。

良い言語インターフェースについての最初の基準は「プログラミング開発中に利用者に与えるストレス」である。プログラミングと言う行為そのものが、ある種のストレスの素であることは疑いはないが、開発者が感じるストレスは利用しているプログラミング言語によって差があることは、複数の言語を利用したことのあるプログラマであれば、多くの人が実感していることだろう。

プログラミング言語によるストレスの与えられ方にはいくつかのパターンがある。たとえばアセンブラーのように低レベル過ぎたり記法が非人間的なものは恒常にプログラマにストレスを与える。あるいは、Prologのようにプログラマの普段の発想よりもレベルが高く挙動を理解するのに困難が伴う(時がある)ものにも恒常的なストレスが伴う。一方、CommonLispやC++のように仕様が巨大で全容を把握しきれないものは、普段のストレスはそれほどでもなくとも、ときどき落とし穴に落ちたように高いストレスを与える場合がある。

良いインターフェースとなるプログラミング言語は恒常にストレスを与えないことはもとより、ストレスのピークが低いことも求められる。

良い言語インターフェースについての次の基準は「プログラマが消費する時間」である。現代のようにコンピュータのコストあたりの性能が向上すると、多くの場合においてもっとも重要なファクタはプログラマがあるタスクを達成するための時間であり、プログラマの生産効率である。

プログラマの生産効率を決定するものは、ひとつには資産、すなわち多種多様な機能をいかにスムーズに言語の中に取り込んでいくかという点であり、もうひとつは前述のストレスがいかに低いかという点であろう。

これらの評価基準は一般的のマン・マシン・インターフェースにとっては当然のことであろう。プログラミング言語にも同じ基準が適用できる。

よって、より良いインターフェースとしてのプログラミング言語には、インターフェースの原則が適用されるべきである。そして、その原則は

- 一貫性
- 柔軟性
- 簡潔性

であると考える。筆者が設計したプログラミング言語Rubyは上で述べたみっつの原則に従うように設計したつもりである。