

日本語プログラム言語「言霊」

岡田 健†1 中鉢 欣秀†2 鈴木 弘†3 大岩 元†4

†1 慶応義塾大学政策・メディア研究科 †2 慶応義塾大学SFC研究所

†3 東京都立航空工業高等専門学校 †4 慶応義塾大学環境情報学部

e-mail: turkey@crew.sfc.keio.ac.jp

概要

日本語を用いて Java 仮想マシン (以下 JavaVM) 上で動作するプログラムを記述できる日本語プログラム言語を設計・実装した。JavaVM のアセンブリ言語であるバイトコードを日本語化し、アセンブラレベルの日本語開発環境を開発した。更に日本語バイトコードを改良しながら日本語プログラム言語を設計している。

Japanese Programming Language “Kotodama”

Ken Okada†1 Yoshihide Chubachi†2 Hiroshi Suzuki†3 Hajime Ohiwa†4

†1 Graduate School of Media and Governance, Keio University †2 Keio Research Institute at SFC

†3 Tokyo Metropolitan College of Aeronautical Engineering

†4 Department of Environmental Informaion, Keio University

Abstraction

We have designed and made a Programming Language “Kotodama” written in Japanese which runs on a Java Virtual Machine. We have transformed the assembly language of Java Virtual Machine called Bytecode to a Bytecode written in Japanese, and the Japanese programming environment has been archived. We are designing the Japanese Programming environment by enhancing the Japanese Bytecode.

1. はじめに

プログラミング教育で常にネックになるのは、英語を基礎にしたプログラム言語を読むのが難しいことである。既存のプログラム言語は欧米において開発され、英語とその文化・思想の影響を強く受けているため、我々日本人には馴染み難い言語になっているのである。

プログラミング言語の日本語化は、FORTH を日本語化したMIND[1]、識別子や予約語の日本語化を試みたCOBOL[2]、助詞の無い日本語表現を可能にしたドリトル[3]などが存在するが、いずれも自然な日本語表現ではない。

本プログラム言語は、日本語を母国語とする人が自然と感じる表現を用いて、プログラムを記述

できる言語を目指して設計している。まずバイトコードを日本語化し、その上で言霊コンパイラを日本語バイトコードを用いて実装した。

2. バイトコードの日本語化

Java プログラムは、クラスファイルに記述されているバイトコードというアセンブリ言語をJavaVM が解釈することで動作する。このバイトコードをニーモニックに変換したものが図1のバイトコードであり、さらに日本語化したものが図1の日本語バイトコードである。

JavaVM はスタックマシンであり、データをスタックに積むことで処理を行う。図1のプログラムを解説すると、Javaのプログラムは変数 x が0だ

if(x==0) y=1; else y=2; (Java ソース)	
1: iload_1	6: Label_0:
2: ifne Label_0	7: iconst_2
3: iconst_1	8: istore_2
4: istore_2	9: Label_1:
5: goto Label_1	(バイトコード)
1: ローカル変数 1 番目から整数を積む。 2: 0 以外ならば、Label_0 にジャンプする。 3: 整数 1 を積む。 4: ローカル変数 2 番目に整数を格納する。 5: Label_1 にジャンプする。 6: Label_0: 7: 整数 2 を積む。 8: ローカル変数 2 番目に整数を格納する。 9: Label_1: (日本語バイトコード)	

図1 バイトコードの日本語化

ったら変数 y に 1 を、そうでなかったら y に 2 を代入している。バイトコード上では変数 x の値を調べる為に、変数 x をスタックに積む(1行目)、そしてその値を検証し、分岐する(2行目)。代入の手順は、まず代入すべき値をスタックに積んでから変数に代入している(3-4,7-8行目)。

このような日本語バイトコードを利用する為のコンパイラを実装した。このコンパイラはバイトコードから日本語バイトコードへ変換できるし、その逆も可能である。

3. 日本語プログラム言語「言霊」

言霊は、日本語バイトコードの文法を拡張する方向で設計された。拡張した点の一つ目は、代入文の表現である。日本語バイトコードでは変数に値を代入するだけで2行を費やしていた。これを「<値>を<変数>に代入する」と表現できるように拡張すると、図2のようになる。

そこで「1をyに代入する。」などの代入文を日本語バイトコードに変換する言霊コンパイラを、

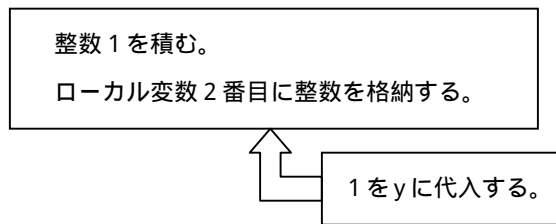


図2 言霊の代入文

日本語バイトコードで実装した。

同様に、日本語バイトコードを拡張して制御構造を使えるようにすると、以下のようなコードになる。

x が 0 ならば、1 を y に代入する。
そうでなければ、2 を代入する。

図3 言霊のコード例

図3のコードを言霊コンパイラで変換すると、図1の日本語バイトコードになる。それをさらに日本語バイトコードコンパイラでクラスファイルに変換するのである。同様に拡張する事で、繰り返し文やメソッド呼び出しなどが使用できる。

4. 考察

言霊は助詞を使うことで自然な日本語表現になるよう設計した。これにより構造化プログラミングレベルの高級化が達成された。今後はオブジェクト指向の機能を日本語で定義する方法を検討中である。

5. 参考文献

[1]片桐明：日本語プログラミング言語MIND基本文法,リギーコーポレーション(1995).
[2]西村恕彦：日本語とCOBOL,情報処理,Vol.8,No.3,pp.157-160(1967).
[3]兼宗進、中谷多哉子、御手洗理英、福井眞呉、久野靖：オブジェクト指向言語「ドリトル」を利用した情報教育について,情報処理学会「情報教育」シンポジウム,Vol.2001,No.9,pp.275-282(2001).