

Java をターゲットにした自国語プログラミングの実験

5M-8

鈴木康彦

野口健一郎

後藤英一

神奈川大学理学部情報科学科

1. はじめに

変換表を与えることにより、キーワード類を自国語表記から正規の Java 表記に変換する変換プロセッサを開発した。これにより、Java のプログラムすべてを自国語で記述することができる。これを日本語に適用して、日本語プログラミングの実験を行った。

2. 研究の動機

情報システムの各国・地域への浸透と、さらにグローバル化の流れから、情報処理技術には次の二つの要求がある。

- ・利用者にとっては、自地域の言語・文化と整合性のよい地域別インターフェースが望まれる
 - ・グローバル・インターオペラビリティなどのためには、共通インターフェースが必要である
- これらの、一見相矛盾する要求を満たす解決策として、次を提案したい。「ひとつの共通インターフェースに対して、地域毎のローカルな表現を持てるようにする。」これを、ソフトウェア技術の最も基本のインターフェースであるプログラミング言語に適用した。

日本語プログラミングについては、さまざまな研究、技術開発が行われてきており、また有効性についても指摘されているが、広く普及するまでには至っていない。[1]

3. 実現方式

3.1 ベースのプログラミング言語の選択

ベースのプログラミング言語として、次の理由から Java(注)言語を選択した。

- ・Java 言語はグローバルなネットワーク環境での利用を狙いとして、国際化機能に配慮が払われている。文字コードに Unicode 2.0 を採用したことにより、ユー

ザ定義の変数名に各国語文字を使うことができる。

(JDK1.2 のコンパイラでは、日本語の出力メッセージもサポートされている。)

3.2 ローカルな言語表現の実現

Java で特定言語（英語）依存の表現は、次の2種類がある。

- ・Java 言語のキーワード類
 - ・Java API（標準クラス）で使用されている名前
- これらについて、自国語に基づいた表記を可能にするために、変換表に基づき変換プロセッサで文字列変換をする方式をとった。(図1) 自国語表記プログラムは正規の Java プログラムへと変換される。この方式は次の利点を持つ。

- (1) 特定の国語に依存していない。広く、いろいろな国・地域語に適用できる。
- (2) 自国語の表現は利用者が設定・変更できる。自国語表現の標準化を待たなくてよい。標準化のための実験ができることにもなる。

実現した変換プロセッサ **rr2Java** の特徴を挙げる。

- (1) 変換対象の文字列は単なる文字テキストのものと、強調文字を含むものとの2種類を受け付ける。後者により、キーワードなどに強調文字を用いることが可能になる。強調文字の表現形式についてはパブリックな文書標準である HTML を採用した。このとき、自国語表記プログラムは、HTML の整形済み(preformatted)

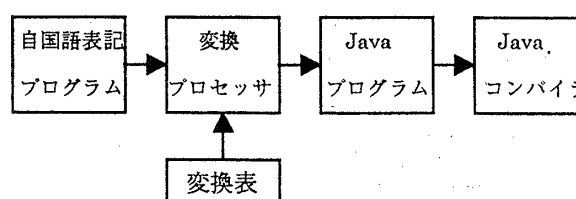


図1 自国語プログラミングの実現方式

テキストとして作成する。

- (2) 言語の語順の違いに対処するため、後置キーワードを可能にするメカニズムを設けた
- (3) Java 言語への依存性も、コメント部分の処理など最小限のものとなっている。(コメントや文字列リテラルは、そのまま表示させる。)

4. 日本語プログラミングへの適用

4.1 方式

日本語表記用の変換表を準備することにより、Java 言語をターゲットとした日本語プログラミングが可能になる。本方式による日本語プログラミングの特徴を挙げる。

- (1) Java プログラムを、キーワードを含めてすべて日本語で表記することが可能になる。(もちろん英文字などの混用もできる。)
- (2) 単語の間はスペース等の区切り記号で区切られる「分かち書き」となる。
- (3) 強調文字を用いることができる。(たとえばキーワードを太字にし、API に関わる名前を斜字にするなど。)

表1に変換表の例を示し、また図2に日本語プログラムの例を示す。

表1 変換表 ja2Java_tbl の例

(強調文字を使用した表とその内部表現)

クラス	class
級	class
公	
私	
長整数	
同期した	
は*	
戻り	
空	
もしも	
.	
.	
<I>文字列</I>	String
<I>システム</I>	System
<I>出力</I>	out
<I>行を印刷</I>	println
<I>処理の糸</I>	Thread
<I>実行</I>	run
.	
.	

```

クラス 口座 {
    公の 文字列 名義人;
    私の 長整数 預金高 = 0;

    公の 口座(名前) {
        名義人 = 名前;
        システム.出力.行を印刷(名義人 + "の口座開設。");
    }

    公の 同期した 長整数 振込み(長整数 金額) {
        預金高 += 金額;
        戻り 預金高;
    }
    // 一部省略
}

クラス 取引は 処理の糸 を拡張 {
    私の 口座 取引口座;
    私の 長整数 取引金額;

    公の 取引(口座 こうざ, 長整数 金額) {
        取引口座 = こうざ;
        取引金額 = 金額;
    }

    公の 空の 実行 {
        もしも (取引金額 > 0) 取引口座.振込み(取引金額);
        // 以下省略
    }
}
    
```

図2 日本語表記 Java プログラムの例

4.2 日本語プログラミングの評価実験

Java 言語の教育を2ヶ月間受けた学生を対象に、同一ロジックの日本語表記プログラムと英語表記プログラムを用いて、内容の読解性のテストを行った。日本語表記プログラムに初めて接したという条件下に関わらず、日本語表記プログラムのほうが若干ではあるが読解性が良い、という結果が得られた。

5. 今後の課題

プログラムの読解性を上げるためのキーワード類の表現については、今後さらに実験心理学的な研究を含めた研究が必要である。

参考文献

[1] 今城哲二：日本語プログラム言語文献ノート，1998 秋のプログラムシンポジウム「日本のプログラミング」報告書，情報処理学会（1999）
 [2] ジェームス・ゴスリン他(村上雅章訳)：The Java™ 言語仕様，アジソン・ウェスレイ・パブリッシャーズ・ジャパン（1997）