

Title	初学者用プログラミング環境 PEN：国際化の実現
Author	陳, 山 / 張, 珏華 / 安倍, 広多 / 石橋, 勇人 / 松浦, 敏雄 / 中村, 亮太 / 西田, 知博
Citation	情報学. 4 卷 2 号
Issue Date	2007
ISSN	1349-4511
Type	Departmental Bulletin Paper
Textversion	Publisher
Publisher	大阪市立大学創造都市研究科情報学専攻
Description	
DOI	

Placed on: Osaka City University

初学者用プログラミング環境 PEN

— 国際化の実現 —

PEN: Programming Environment for Novices

— Implementation of Internationalization —

陳 山[†], 張 珏華[†], 安倍 広多[†], 石橋 勇人[†], 松浦 敏雄[†],
中村 亮太^{††}, 西田 知博^{†††}

Shan Chen, Juehua Zhang, Kota Abe, Hayato Ishibashi, Toshio Matsuura,
Ryota Nakamura, Tomohiro Nishita

概要 初学者用プログラミング環境 PEN は、理解が容易な日本語プログラミング言語を備えており、短時間でプログラミングの基礎を習得できるように工夫されている。また、入力支援機能、プログラム動作の観察を助ける機能など初学者により優しい環境を提供している。一方、中国にはいくつか中国語でプログラムを書けるツールが存在するものの、PEN のような教育に適したものは見当たらない。同じ東洋国家である日本と中国は、プログラミング教育においても、同じような問題を抱えることが多い。また、PEN の優れた機能は日本語に限定しなくても他言語においても通用すると考えられる。従って、PEN を国際化すれば、日本語を母国語としない多くの学生に対するプログラミング教育でも有用であると考えた。本研究は初学者用プログラミング環境 PEN の国際化の実現を研究目的とする。国際化の皮切りとして、まず中国語化することを試みた。中国語化する際に、5 つの実装方法を考え、その中のマルチ言語に同時に対応する実装方法を採用し、実装を行った。それと同時に、他の言語も容易に追加できる仕組みを作り、中国語化できた PEN をベースに、英語、台湾語、韓国語を追加し、動作を確認した。

キーワード：プログラミング入門教育、国際化、中国語プログラミング言語、プログラミング環境、JavaCC

Keyword : Programming Primer, Internationalization, Chinese Programming Language, Programming environment, JavaCC

1 はじめに

高等学校の教科「情報」を始めとして、初等中等教育から大学に至るまで情報教育が行われている。しかし、現状の情報教育の多くはリテラシーに焦点があてられており、「情報処理の理解」を深めるものとはなっていない。しかし、「情報処理の理解」のないままに、利用法のみを学んでもその理解は不完全となり、本質的な誤解や認識不足によって大きなトラブルを引き起こすことも起こりうる^[1]。このような背景から、情報処理学会は「日本の情報教育・情報処理教育に関する提言 2005」^[2]を出し、プログラミングを始めとした『「手順的な自動処理」の理解』の重要性を訴えている。

このような「手順的な自動処理」の理解を助けるための実践的な研究として、ドリトル^[6],

Nigari^[7], PEN (Programming Environment for Novices)^{[3] [4] [9]}などがある。これらは、いずれも日本語のプログラミング言語を提供しており、初学者への配慮がなされている。なかでも PEN は、理解が容易なプログラミング言語とプログラムの入力補助機能を備えており、さらに、ステップ実行機能、スロー実行機能、変数表示機能などにより、初学者に対する支援機能が豊富であり、初学者教育に適している。

一方、中国では、中国語をベースとしたプログラミングツールとして、易語言^[16]、漢語編程^[17]などが存在している。しかし、いずれも実用的なプログラム開発用のものであり、PEN のようにプログラミング教育に適したものではない。日本と中国は同じ東洋の国であり、プログラミング教育において、同じような問題を

[†] 大阪市立大学 Osaka City University

^{††} 大阪府立泉北高校 Semboku Senior High School

^{†††} 大阪学院大学 Osaka Gakuin University

抱えていることが多い。特に英語を母国語としないため、英語をベースとするプログラム言語を理解することがネックになっている。このような問題を解決するために開発されてきた PEN は、そのコンセプトは日本語に限定しなくても他言語においても通用すると考えられる。PEN を国際化すれば、日本語を母国語としない多くの学生に対するプログラミング教育で有用であると考えた。

本研究は初学者用プログラミング環境 PEN を国際化することを目的とする。PEN の国際化を実現するために、まず、中国語化を行う。その際、中国語以外の言語にも対応しやすいような方法を検討する。

本論文の構成について、第 2 章は国際化における問題について説明する。また、実装方法の選択についても触れる。第 3 章では、中国語化を行う際の作業について詳しく説明する。第 4 章は中国語化が完成したものに、英語をベースとした言語を追加する手順を説明しながら、評価を行う。また、利用者の感想に基づいた評価も行っている。第 5 章はいくつかの問題について考察する。

2 国際化における問題

2.1 国際化に必要な項目

PEN は Java 言語によって記述されている。Java 言語には国際化するための仕組みが用意されている。それによって、ボタンやメニューなどの GUI 部分の国際化は比較的に容易に行える。しかし、PEN を国際化するにはそれだけでは不十分である。GUI 部品に加えて、設定ファイル、入出力ファイル及び対象とするプログラミング言語などを国際化する必要がある。以下では、これらの各項目について概説する。

2.1.1 GUI 部品の国際化

PEN の実行画面の一番左上には「ファイル」、「編集」、「ヘルプ」などのプルダウンメニューがある。その直下にいくつかのボタンが配置されている。また、ボタンの右側に実行速度をコントロールするスライドと実行状態

のラベルがある。さらに、「編集画面」、「コンソール画面」、「変数表示画面」など画面名を表示するラベルが各画面の上に表示されている。これらの各 GUI 部品のラベルについてそれぞれの国の言語に置き換える必要がある。

2.1.2 設定ファイル等の国際化

PEN のユーザーインタフェースの一部は、外部設定ファイルを読み込んで作成するものが存在する。これらを国際化するには、上記の GUI 部品項目と違って、設定ファイルそのものの文字コードが問題となる。

2.1.3 入出力の国際化

PEN には、作成したプログラムの保存と、保存されたプログラムの読み込みの機能が組み込まれている。しかし、既存の PEN は日本語にしか対応していない。つまり、PEN において、プログラムの読み書きする際の文字コードが Shift-JIS か EUC_JP に限定されている。PEN の国際化を図るには、当然日本語以外の文字コードで記述されたプログラムの入出力にも対応する必要がある。

2.1.4 対象とするプログラミング言語の国際化

PEN は日本語でプログラムを書くことができる。その中核部分は日本語の構文を解釈するパーザである。様々な言語でプログラムを書けるためには、言語構文の国際化が不可欠である。この項目が PEN の国際化における最も重要な部分である。

2.2 言語の切り替え方法

PEN の国際化において、対象とするプログラミング言語の違い、及び文字コードの違いを扱う方法として以下のものが考えられる。

2.2.1 言語毎に独立して実装する方法

この実装方法は文字通りに、言語毎に、違うバージョンを用意することである。既存の PEN を日本語バージョンとして、それをベースに、中国語バージョン、英語バージョンなどを作成していくことになる。ソースコードも、

コンパイル後の完成されたものも完全に独立しているため、それぞれのバージョンは他言語のバージョンに全く影響を受けないので比較的実装は容易である。しかし、それぞれのバージョンをすべて独立に管理しなければならないので、共通した修正を加える必要がある場合など、その作業量は大きくなる。

2.2.2 複数の言語を同時に扱う方法

複数の言語を1つのアプリケーションプログラムで扱う方法として以下の方法が考えられる。

(1) 起動時のオプションとして指定する方法

この方法では、起動時に対象とする言語をオプションとして指定する。コンパイルされた時点では、事前に用意された各言語はすべて1つのアプリケーションの中に実装しておく。起動時のオプション指定によって、指定された言語のみを利用できるようにする。また、違う言語を利用する場合は、一旦プログラムを終了させ、再起動して新たにオプションを指定する必要がある。この方法はソースが一体化になっているので、共通した修正を加える場合に、一括で行える。

(2) 実行時にメニューなどで切り替える方法

この方法は、上記の起動時のオプションによる選択方法を拡張するものである。ただし、違うところは、起動時ではなくて、実行時でも簡単に他言語に切り替えることができる。切り替える方法としては、メニューまたはボタンでの選択が考えられる。

(3) OSの言語環境情報により自動的に切り替える方法

この方法は、OSのデフォルトの環境情報によって、自動的に言語を切り替える。例えば、日本語版 Windows XP の場合、自動的に日本語で起動する。また、中国語版 Windows XP の場合は自動的に中国語で起動する。

(4) マルチ言語に同時に対応する方法

マルチ言語バージョンも事前に用意された言語はすべて実装するようになっている。他の実装方法との違いは、言語の切り替えが不要になる。実装される言語をすべて一つに統

一することにすることで、切り替えのメカニズムを省くメリットがある。

上記4つの実装方法の中で、「マルチ言語に同時に対応する方法」はすべての言語を統一することによって、切り替えが不要になり、柔軟性と拡張性に優れている。本研究ではこの方式を採用した。

3 中国語化の実現

本研究はまず中国語化を目指す。本章は2.1で述べた各項目毎の中国語化作業について具体的に説明する。

3.1 GUI 部品の中国語化

3.1.1 Java の国際化手法

JAVA 言語には Locale オブジェクトという国際化する手法が用意されている。Locale オブジェクトを使用する際に、地域対応したリソースをプログラムと分離して外部にプロパティファイルとして持ち、それぞれのロケールに合ったリソースにアクセスすることによって国際化されたプログラムを作ることになる。PEN を中国語化する際も、ラベルなど散在している GUI 部品関連の箇所を国・言語毎に一つのプロパティファイルにまとめ、Locale オブジェクトを利用した。

3.1.2 プロパティファイルの作成

PEN を Locale オブジェクトを利用して中国化するため、まず、プロパティファイルを作成し、GUI 部品に関連する箇所をすべてそれらのプロパティファイルに移転する必要がある。そのため、日本語対応の「Locale_ja_JP.properties」と中国語対応の「Locale_zh_CN.properties」を作成する必要がある。「Local_」の後に続く「ja」、「zh」は言語名であり、言語名の後の「_」に続く「JP」、「CN」は国名である。GUI 部品に関連する箇所をプロパティ名をつけてすべてプロパティファイルに移転した。

3.1.3 クラス LocaleProperty の作成

プロパティファイルを作成した後、Locale

オブジェクトを呼び出すために、新たに `LocaleProperty` (図 3.1 参照) というクラスを作成した。これは PEN を起動する際に、自動的に現在使用されている OS のロケール情報を判断するためのクラスである。

```
public final class LocaleProperty {
    //System の言語コード
    public final static String LANGUAGE = System.getProperty("user.language");
    //System の国コード
    public final static String COUNTRY = System.getProperty("user.country");
    //Locale の設定
    public static Locale locale = new Locale(LANGUAGE, COUNTRY);
    //Locale ファイルの取得
    public static ResourceBundle LOCALE_PROPERTY = ResourceBundle.getBundle("Locale/Locale");
}
```

図 3.1 クラス `LocaleProperty` の抜粋

3.1.4 プロパティファイルを参照するための修正

PEN のクラスの中で、参照する必要のある箇所では `rb.getString("プロパティ名")` のように記述すれば、そのプロパティ名に対応するそれぞれの言語の表現に置き換えることができる。表 3.1 はプロパティファイルの日中対応表の一部である。

上記手順のように、`Local` オブジェクトを利用することによって、GUI 部品中国語化を実現した。

表 3.1 プロパティファイルの日中対応表 (一部)

プロパティ名	日本語	中国語
columnNames1	型	型
columnNames2	変数名	変数名
columnNames3	値	値
new_button_label	新規	新建

3.2 設定ファイル等の中国語化

3.2.1 ButtonList.ini

PEN ではプログラムの制御構造の入力を容易にするために入力支援ボタンが用意されており、このボタンはカスタマイズ可能であり、ボタンの定義は `ButtonList.ini` ファイルに記述されている。

「プログラム入力支援ボタン」を中国語にするためには、以下の二つの作業が必要であ

る。一つは、中国語で `ButtonList.ini` を記述することである。もう一つの作業は、PEN の起動時に、指定された文字コードで `ButtonList.ini` を読み込むメカニズムを作ることである。

3.2.2 言語ごとに ButtonList.ini を区別

中国語化だけでなく今後の他の言語の追加も考慮し、`ButtonList.ini` を言語ごとに区別することにする。日本語版のファイル名を `ButtonList_JP.ini` とし、中国語版のファイル名を `ButtonList_CN.ini` とする。どちらのファイルを読み込むかは、`Locale` オブジェクトを利用して `LocaleProperty` の判断に任せることにする。

以上の方法で、「プログラム入力支援ボタン」を中国語または日本語で記述できるようになった。

3.3 入出力の中国語化

3.3.1 ファイルを扱う際に使う文字コード

PEN では、編集画面に表示されたソースプログラムをファイルに保存できる。保存されたファイルの文字コードは、WindowsXP 日本語版の場合、Shift-JIS であり、日本語版 UNIX の場合、通常 EUC_JP である。

保存されたファイルを読み込む際には、ファイルの文字コードが Shift-JIS または EUC_JP のどちらかを自動的に判断する仕組みを有している。しかし、その他の言語の文字コードには対応していない。

3.3.2 ファイル書き出しの中国語への対応

ソースファイルの読み書きを日本語と中国語両方に対応するために、文字コードを Unicode に限定した。まず、書き出しは、`FileSave` クラスによって定義される。そこには文字コード指定に該当する記述ないので、`FileOutputStream` クラスを利用して、Unicode を指定した。

3.3.3 ファイル保存の中国語への対応

次に、ファイルの読み込みに関する部分を中国語に対応するために、`FileOpen` の Shift-

JIS と EUC_JP の判断メカニズムを削除し、文字コードを Unicode に固定した。

ソースプログラムの読み書きとも文字コードを Unicode にすることによって、中国語を扱えるようになった。

3.4 対象とするプログラミング言語の中国語化

3.4.1 文法定義の仕組み

PEN では対象とするプログラミング言語の文法は IntVParser.jjt というファイルに記述している。このファイルをコンパイラコンパイラである JavaCC が入力し、パーザプログラムを生成する。生成されたパーザは Java のソースコードなので、容易に Java アプリケーションに組み込める。IntVParser.jjt の一部を図 3.2 に示す。

```
TOKEN [IGNORE_CASE] :
{
  < VARINT: "整数">
  :
  | < FOR: "を">
  | < FOR2 : "から">
  | < FOR3 : "まで">
  | < FOR4 : "ずつ">
  | < FORADD: "増やし" ("ながら" | "つつ") <CM> >
  | < FORSUB: "減らし" ("ながら" | "つつ") <CM> >
  :
}
```

図 3.2 IntVParser.jjt の抜粋

3.4.2 キーワードの中国語への対応

キーワード部分を日本語と中国語の両方に対応するために、図 3.2 を図 3.3 のように書き直した。これによって、パーザは日本語と中国語のいずれのキーワードで書いたプログラムも受理できるようになった。

```
TOKEN [IGNORE_CASE] :
{
  < VARINT: "整数">
  :
  | < FOR : "を" | "把">
  | < FOR2 : "から" | "从">
  | < FOR3 : "まで" | "至">
  | < FOR4 : "ずつ" | "以">
  | < FORADD: "増やし" | "ながら" <CM> | "増やし" | "つつ" <CM> | "为基数递增" <CM> >
  | < FORSUB: "減らし" | "ながら" <CM> | "減らし" | "つつ" <CM> | "为基数递减" <CM> >
  :
}
```

図 3.3 キーワードの中国語への対応 (日中混在)

3.4.3 文の順序の中国語への対応

中国語と日本語の文法には違いがある。特に語順について、日本語では動詞は目的語の後ろに置くが、中国語では動詞は目的語の前に置く。文の語順について、IntVParser.jjt には図 3.4 のように記載している。これは「～を(改行あるいは改行なしで) 表示する」という構文に対する定義である。

```
void PutStat() :
{
  {
    PutParam() ( <CM2> PutParam() ) * ( <PUT>
  | <PUTLN> {jstThis.n = "%n";} )
    {jstn000.line_num1 = jj_consume_token(SM).beginLine; }
  }
}
```

図 3.4 文の順序に関する定義 (1)

この中の<PUT>は「を改行」("なし" | "無し") "で" ("表示" | "出力" | "印刷") "する" と定義されている。また、<PUTLN>は「("表示" | "出力" | "印刷") "する"」定義されている。この構文定義によって、プログラムに「a を改行なしで表示する」のように書くことができる。しかし、中国語の場合は、「不換行显示 a」と書かないといけないのに、現状の構文定義でキーワードのみ書き替えただけでは、「a 不換行显示」のようになってしまう。そのために、図 3.4 を図 3.5 のように書き直すことによって、構文定義は中国語の文、つまり、「a 不換行显示」を受理するようにできる。

```
void PutStat() :
{
  {
    ( <PUT > | <PUTLN > {jstThis.n = "%n";} ) PutParam() ( <CM2> PutParam() ) *
    { jstn000.line_num1 = jj_consume_token(SM).beginLine; }
  }
}
```

図 3.5 文の順序に関する定義の中国語化 (中国語のみ)

しかし、目指してる実装方法はマルチ言語なので、一つのパーザで複数の言語に対応しなければならない。要するに、「a を改行なしで表示する」と書いても、「不換行显示 a」と書いても受理するようにしたい。もし、図 3.6 と図 3.5 のように変更してしまうと、構文定義は中

国語文法にしか対応できなくなるので、日本語で「aを改行なしで表示する」と書いても構文エラーと見なされる。

そこで、構文の順序だけでなく、キーワードに関する部分も両国語に対応する必要がある。図 3.6 に関連するキーワード部分の修正箇所を図 3.6 に示す。

```
TOKEN [IGNORE_CASE]:
{
< VARINT: "整数" | "整数">
:
|< PUTLN_SOV: "を表示する" | "を出力する" | "
を印刷する" >
|< PUT_SOV: "を改行なしで表示する" |
"を改行なしで出力する" | "を改行なしで印刷する"
>
|< PUTLN_SVO : "就显示" | "就输出">
|< PUT_SVO: "不换行显示" | "不换行输出">
:
}

```

図 3.6 キーワードの中国語への対応（日中混在）

<PUTLN>と<PUT>は、動詞を目的語の後ろに置く<PUTLN_SOV>,<PUT_SOV>と動詞を目的の前に置く<PUTLN_SVO>,<PUT_SVO>の両方を記述している。このようにキーワードを日中両国語で記述する必要がある。

また、文の順序を定義する場所にも図 3.7 のように変更を行った。これによって、「aを改行なしで表示する」と書いても、「不換行显示 a」と書いても、正しく受理できる。このようにキーワードと順序の定義も二通りにすれば、日中両方の分を受理できるようになる。

```
void PutStat():
{
{
(<PUT_SVO> | <PUTLN_SVO> {jjetThis.n = "
¥n";}) PutParam() (<CM2> PutParam())*
{ jjetn000.line_num1 = jj_consume_token(SM).b
eginLine;}
|
PutParam() (<CM2> PutParam())* (<PUT_S
OV> | <PUTLN_SOV> {jjetThis.n = "¥n";})
{ jjetn000.line_num1 = jj_consume_token(SM).b
eginLine;}
}
}

```

図 3.7 文の順序に関する定義の中国語化（日中混在）

3.4.4 IntVParser. jjt の文字コード問題

前節の修正によって、IntVParser.jjt の中に日本語と中国語が混在するので、Shift_JIS で保存できなくなる。そこで、両国の言語に対応させるため、Unicode で保存することにした。

しかし、JavaCC でコンパイルする時に文字コードを指定するオプションが存在せず、常に OS が仮定しているデフォルトの文字コードでコンパイルするようになっている。そこで、Unicode をデフォルトの文字コードとする Linux (CentOS3.2) 上でコンパイルした。作成されたパーザを PEN の中に組み込むことによって、日本語と中国語のマルチ言語対応が可能になった。

中国語対応版の PEN を実行した画面表示例を図 3.8 に示す。

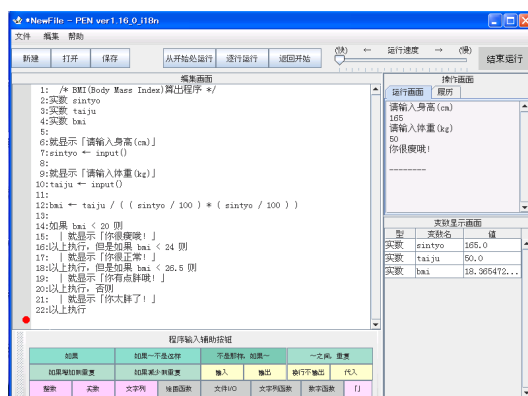


図 3.8 中国語版 PEN の実行例

4 評価

4.1 新たな言語の追加

本節では中国語以外の言語を新たに追加するための手間について考察する。以下では英語の追加を例として、どれだけの手間を要したかについて評価する。

(1) GUI 部品の英語化

英語を追加する場合、中国語のプロパティファイルと同じように、まずは

「Locale_en_US.properties」というファイルを新規に作成する必要がある。プロパティ名に対応する値の部分を英語に翻訳するだけで、この部分の英語化が完了する。(2) 設定ファイル等の英語化

設定ファイル等の英語化もほぼ中国語化と同じように行える。

まず、英語で翻訳されている「ButtonList_US.ini」を用意する必要がある。

次に,Locale_en_US.properties 内に " ButtonList_pass= ButtonList_US.ini " と "file_encoding= ascii "指定しておく必要がある.この指定によって,Locale オブジェクトを利用して Ascii コードで ButtonList_US.ini を呼び出せるようになる.これで設定ファイル等の英語化が完成する.

(3) 英語の文字コードへの対応

第3章で述べたように,ファイルの保存と読出はすべて Unicode で行える.従って,英語に対応するための手間は不要である.

(4)言語構文への追加

言語構文に英語を追加するには,IntVParser.jjt を編集する必要がある.まず,編集する部分は,キーワードに相当する部分である.図 4.3 に文法定義の一部を示す.

```
|< VARFLOAT: "実数" | "実数" |  
"reloat">  
|< VARREFERENCE: "参照" |  
"reference">  
|< PROCEDURAL: "手続き" | "手続"|  
"procedure">  
|< FUNCTION: "関数" | "函数" |  
"function">
```

図 4.3 IntVparser.jjt 英語キーワードの追加(1)

キーワードの部分は,第3章に説明したように,文法によって分けて設定されたものが存在している.英語の文法は中国語とほぼ同じなので,中国語と同じ場所に対応する語句を追加することになる.

4.2 新たな言語の追加に関する評価

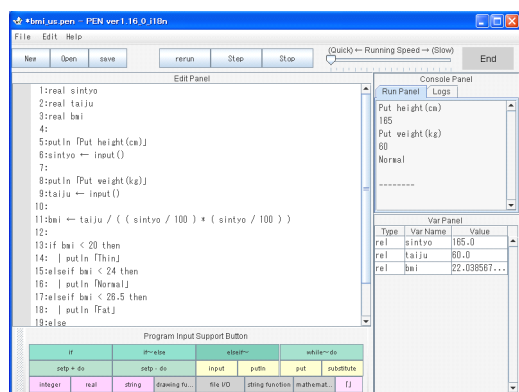


図 4.4 英語化された PEN の実行画面

図 4.4 は BMI プログラムを英語で書いたものを実行している時の実行画面例である.

英語化に要した作業時間はおよそ 20 時間であった.

英語以外に,台湾語と韓国語の追加作業も行った.台湾語は漢字の形以外,中国語とほぼ同じなので,約 10 時間の作業で台湾語対応が実現できた.また,韓国語の文法は日本語とほぼ同じであるため,15 時間で対応できた.

5 考察

5.1 実装方法に関する考察

5.1.1 言語構文に関する考察

今後,さらに言語を追加するようになること,構文に関する記述も膨大していくことが想定できる.膨大化するパーザはアプリケーションの実行速度を低下させる恐れがある.また,保守も大変になる.現時点三種類の言語しか対応していなくてパフォーマンス上に問題がないが,今後,言語毎に言語構文を定義しなければならない可能性もある.

5.1.2 GUI の表示言語に関する考察

利用者は,使っているコンピュータの OS のデフォルト言語だけではなく,別の言語を使用したい可能性が考えられる.また,使用中に起動した時点の言語とは別の言語で使いたいという要望も考えられる.従って,より使いやすくするために,言語判断の仕組みをさらに工夫する必要がある.

5.2 扱う文字コードに関する考察

今回はマルチ言語対応するものを実装した.また,入出力する際の文字コードも Unicode に統一した.一方,日本では Shift-JIS や EUC_JP,中国では GB2312 などの規格化された文字コードが存在している.大部分の利用者にとって,Unicode よりもむしろこれらの文字コードのほうが好ましい.

今後は,利用者によって,文字コードの選択が随時かつ自由に選択できるように PEN を拡張していくことを課題の一つとする.

5.3 マルチ言語の混在問題に関する考察

今回実装したパーザはマルチ言語対応なの

で、「もし `bmi < 20 then`」のようなおかしい文をプログラムの中に書いても、許すことになる。ただ、通常はこのようなプログラムを書くことはないと思われる。また、プログラムの性能に影響がないので、利用者にとって特に困ることがないと思う。

6 おわりに

本稿では、初学者用プログラミング環境 PEN を中国をはじめとする国々の初学者のために、国際化させる仕組みを考案し、中国語版、英語版、韓国語版および台湾語版などを実装し評価を行った。

国際化を行う際、より実装しやすくするため、PEN を GUI 部品、設定ファイル、扱う文字コード、言語構文などいくつかの部分に分けて独立に作業を行った。

今後は、各国のネーティブの方々にも使用してもらい、改良を続けていきたい。

参考文献

- [1] 情報処理学会情報処理教育委員会; 2005 年後半から 2006 年初頭にかけての事件と情報教育の関連に関するコメント, <http://www.ipsj.or.jp/12kyoiku/statement2006.pdf> (2008/01/31 確認)。
- [2] 情報処理学会情報処理教育委員会; 日本の情報教育・情報処理教育に関する提言 2005, <http://www.ipsj.or.jp/12kyoiku/proposal-20051029.pdf> (2008/01/31 確認)。
- [3] 初学者向けプログラミング教育環境 PEN Web ページ; [http://www.media.osaka-cu.ac.jp/PEN/\(2008/01/31](http://www.media.osaka-cu.ac.jp/PEN/(2008/01/31) 確認)。
- [4] 西田知博, 中村亮太, 山本武生, 松浦敏雄; プログラミング環境 PEN --描画とファイル I/O 機能の実装, 情処学情報教育シンポジウム SSS 2006 論文集, pp.69--74 (2006)。
- [5] 軽野宏樹, 木實真一, 上林弥彦; ALAN-K プロジェクト; Squeak を活用した創造的な情報教育の試み, 情報処理学会研究報告 2003-CE-69, pp.1--8(2005)。
- [6] 兼宗進, 中谷多哉子, 御手洗理英, 福井真吾, 久野靖; 初中等教育におけるオブジェクト指

向プログラミングの実践と評価, 情報処理学会論文誌, Vol.4.1, No.SIG13, pp.58--71 (2003)。

[7] 長慎也, 甲斐宗徳, 川合晶, 日野孝昭, 前島真一, 笈捷彦; Nigari - Java 言語へも移行しやすい初心者向けプログラミング言語, 情報処理学会研究報告 2003-CE-71, pp.13--20 (2003)。

[8] Tomohiro Nishida; PEN: Programming Environment for Novices with Programming Language for University Entrance Examinations, Activation of ICT/Informatics Education and Entrance Examination of Universities/Colleges, pp.41--43, (2006-06)。

[9] 西田知博, 中村亮太, 松浦敏雄; 入力支援機能を備えた初学者用プログラミング環境 PEN, 情報処理学会 情報教育シンポジウム SSS2005, pp.155-156 (2005-08)。

[10] 中村亮太, 西田知博, 松浦敏雄; プログラミング入門教育用学習環境 PEN, 情処研報, 2005-CE-81, Vol.2005, No.104, pp.65-71 (2005-10)。

[11] 西田知博, 中村亮太, 松浦敏雄; 初学者用プログラミング環境 PEN, 平成 17 年度 情報処理教育研究集会 講演論文集, pp.-- (2005-11)。

[12] 西田知博, 中村亮太, 松浦敏雄; プログラミング環境 PEN - 描画とファイル I/O 機能の実装, 情報処理学会 情報教育シンポジウム SSS2006, pp.69-74 (2006-08)。

[13] 宮本友介, 原田章, 安留誠吾, 中西通雄, 西田知博; 文系学部を対象としたリテラシー科目におけるプログラミング教育 — PEN を用いた実践例 一, 平成 18 年度情報教育研究集会, pp.201-203 (2006-11)。

[14] 西田知博, 中村亮太, 松浦敏雄; プログラミング学習環境 PEN . 授業実践とカスタマイズ機能, 2007 PC カンファレンス, CD-ROM (2007-08)。

[15] 西田知博, 原田章, 中村亮太, 宮本友介, 松浦敏雄;; 初学者用プログラミング学習環境 PEN の実装と評価', 情報処理学会論文誌, Vol.48, No.8, pp.2736-2747 (2007-08)。

[16] 易語言 Web ページ; [http://www.dywt.com.cn\(2008/01/31](http://www.dywt.com.cn(2008/01/31) 確認)。

[17] 漢語編程 Web ページ; [http://www.hybcw.com\(2008/01/31](http://www.hybcw.com(2008/01/31) 確認)。