

## 1 教育用プログラミング言語の動向

兼宗 進 一橋大学総合情報処理センター

### 教育用プログラミング言語

2006年3月に東京で、「教育用プログラミング言語に関するワークショップ2006」が開催された。このワークショップでは、教育用途で使われているプログラミングについて、表-1の各言語の作者、または国内における第一人者に講演をお願いした。

詳しくは並木実行委員長による報告<sup>1)</sup>を参照されたい。ここでは取り上げられた言語を紹介する。

第1のグループは、実用性を重視した言語グループである。JavaやC/C++は、社会で使われている言語を体験させたいという場合に使われることがある。従来では、FORTRAN、COBOL、PL/Iなどが相当し、現在ではVisualBasicなど、実務で使われる言語もここに含まれる。JavaScriptは、実用的に使われているだけでなく、構文が簡潔なことから、最近では教育用途に使われることが多くなってきた。

第2のグループは、従来から使われてきた簡易的な言語グループである。従来では、Pascalなども相当し、現在ではPerl、Ruby、PHPなどのスクリプト言語もここに含まれる。簡易的な文法を持つ初期のBASICは、黎明期のパーソナルコンピュータに標準で内蔵されていたこともあり、多くの学校で利用された経緯がある。LOGOはLISPの設計思想を土台に教育用に設計された言語で

あり、1980年代を中心に全国で広く利用された。LISPは構文が簡潔であり、Scheme等のよい教科書が書かれたことから、大学の一般教育などで利用された経緯がある。

第3のグループは、教育用途を目的として設計された最近の言語である。SqueakはAlan Kayたちが設計したSmalltalkの処理系であり、Squeak eToy<sup>☆1</sup>はその上に構築された子供向けのプログラミング環境である。言霊は自然な日本語に近い記述を可能にしており、現在は「ことだま on Squeak」<sup>☆2</sup>として、Squeak eToyにその成果を反映している。なでしこ<sup>☆3</sup>は入門用の言語で、初心者が自分のやりたい作業を簡潔に記述できることを目的としている。ドリトル<sup>☆4</sup>は筆者が設計した教育用の言語で、小中学生がプログラミングを通してソフトウェアを学ぶことを目的としている。PEN<sup>☆5</sup>は大学入試センターがセンター試験科目である「情報関係基礎」において出題しているDNCLという言語仕様を実装し、改良したものである。ここで紹介した言語はすべて、日本語での記述が可能になっていることも特徴である。

第4のグループは、特定の用途を意識したプログラミング言語である。HSP<sup>☆6</sup>とTonyu<sup>☆7</sup>は、ゲームを記述するために使われることが多い。Rosettaは美術表現のために設計された言語であり、同様な用途でProcessing<sup>☆8</sup>などが知られている。Viscuit<sup>☆9</sup>は子供を対象とした例示型のプログラミング言語である。ワークショップでは、

グループ	言語の例
1. 実用言語	Java, C/C++, JavaScript
2. 伝統的な教育用言語	BASIC, LOGO, LISP
3. 最近の教育用言語	Squeak eToy, 言霊, なでしこ, ドリトル, PEN
4. 特定用途向け言語	HSP, Tonyu, Rosetta, Viscuit

表-1 紹介された教育用言語の分類

☆1 <http://www.squeakland.org/>  
☆2 <http://www.crew.sfc.keio.ac.jp/squeak/>  
☆3 <http://nadesi.com/>  
☆4 <http://dolittle.eplang.jp/>  
☆5 <http://www.media.osaka-cu.ac.jp/PEN/>  
☆6 <http://www.onionsoft.net/hsp/>  
☆7 <http://tonyu.jp/>  
☆8 <http://processing.org/>  
☆9 <http://www.viscuit.com/>

```
public class HelloWorld {
    public static void main(String[] args){
        System.out.println("Hello World");
    }
}
```

図-1 Java のプログラム例 (Hello World)

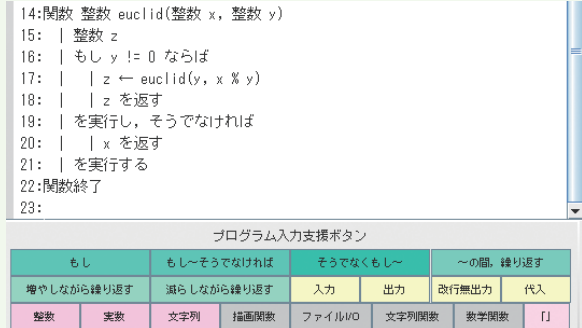


図-2 PEN の構造化エディタ

ワンダーボーグ<sup>☆10</sup>という昆虫型のロボット教材を制御するための言語も紹介された。命令を示すブロックを画面上に配置してプログラムを記述する。

## 求められる性質

教育用のプログラミング言語にはどのような性質が求められるのだろうか。ここでは、実際の言語に見られる特徴を紹介する。

### プログラム記述の工夫

教育用の言語では、プログラム記述について、それぞれの工夫がなされている。ここでは代表的な特徴を紹介する。

#### □ 構文と編集環境

一般に、汎用的なプログラミング言語の構文では多くの定義や宣言が必要であり、構文自体にも多くの記号が使われている。

図-1 は、言語の学習時に最初に学ぶことが多い「文字列を表示するプログラム」を Java で記述したものである。

これを見ると、文字列を出力する「print("Hello World");」以外にも、public, static, void, system.out など、初心者が理解することが困難な構文要素が数多く含まれており、学習を始めてしばらくの間は「意味を理解できないおまじない」としてそれらを記述する必要がある。

また、「{」「}」「;」など、日常使わない記号を一字一句間違えないで入力する必要があるため、初心者にとっては意味不明のエラーが発生する原因となり、それが挫折の原因の1つになっている。

一般に、学習を始めたばかりの初心者は、構文を理解する過程で試行錯誤を繰り返すため、構文エラーが多く発生する傾向がある。このような構文上の難易度を下げするために、多くの教育用の言語では、定義や宣言を省略して記述することを可能にしたり、構文上の記号を減らす工夫をしている。

また、編集時に構文をチェックすることで、実行時の構文エラーを防ぐ工夫も行われている。図-2 は、PEN の構造化エディタである。制御構造を中心に、画面下部のボタンから入力を行うことができる。

その他、Squeak eToy では、制御構造だけでなく命令や引数を含めたすべてを画面から選択的に入力できるようになっており、基本的に構文エラーは発生しない。

#### □ オブジェクト指向

教育用のプログラムでは、画面上のオブジェクトを操作するなど、オブジェクト指向との親和性が高い。しかし、多くの汎用言語で使われている「クラス設計を行い、クラス定義を記述してからオブジェクトを生成する」方式では、動かしながらプログラミングを学んでいく学習を行いにくいという問題がある。

そこで、教育用の言語では、クラスよりもオブジェクトを主体としたオブジェクト指向を採用しているものが多い。たとえば、Squeak eToy, ドリトル, Tonyu では、画面上のオブジェクトにメソッドを定義することで、クラス定義に相当する記述を行っている。

図-3 は、ドリトルのプログラム例である。あらかじめ用意された雛形のオブジェクト(タートル)を複製することで新しいオブジェクトを生成し、他のオブジェクトと衝突したときの動作をメソッドとして定義している。

#### □ 日本語での記述

利用者として小中学生や初心者を想定した場合には、プログラムを母国語で記述することは大きな利点となる。

<sup>☆10</sup> <http://www.roboken.channel.or.jp/borg/>

カメラ=タートル！作る。  
カメラ:衝突=「! 180 右回り」。

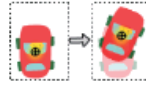
図-3 ドリトルのオブジェクト生成例

「こんにちは」と表示する。

図-4 なでしこのプログラム例

## 車を動かそう

1) プログラムの左右で車がずれて、置いてある。



2) 車をクリックすると、車が動き出す。

3) プログラムの車のずれ方を変えてみよう。車の動き方が変わるよ。バックしたり、右に曲がったり、上手に運転できるかな。

図-5 Viscuit によるアニメーションの説明図

一般に、一度に複数の新しいことを同時に学ぶことは学習者への負荷が大きい。プログラミングをはじめて学ぶ学習者にとって、「プログラミング言語」と「英語という外国語」を扱うことは、格段に敷居が高くなってしまっているのである。

その意味で、教育用プログラミングワークショップ 2006 で取り上げた言語では、命令語に日本語を扱えるものが多かった。

図-4 は、なでしこのプログラム例である。自然言語としての日本語に近い形で記述できるため、親しみが湧き、可読性も高いことが期待できる。

Java や JavaScript でも変数名などに日本語文字を使うことは可能だが、「{」「}」「;」などの記号に 2 バイト文字を使うことはできないし、単語の区切りに 2 バイトの空白文字を使うとエラーになり、発見が難しいバグにつながる。

よって、教育用途を考えた場合には、変数名や命令語だけでなく、記号や空白文字などを含めたプログラム全体で、学習者の母国語に対応することは有効である。

また、同じ意味の記号等をグループ化して同等として扱うことも必要になる。たとえばドリトルでは、文の終わりを示す記号として、1 バイト文字と 2 バイト文字の「。」と「。」を同等に扱い、命令語についても「歩く」「あるく」などを同等に扱っている。

### □ テキスト以外のプログラム記述

プログラムの表現として、文字を使わない言語も提案されている。

画面上に命令などを表すブロックを配置し、フローチャートのように連結してプログラムを記述する図形型の

言語は、LEGO MindStorms<sup>☆11</sup> や Baltie<sup>☆12</sup> などで採用され、ロボット制御などに利用されている。

プログラムの動作を記述するのではなく、「このように動作してほしい」という結果を例で示すことにより、計算機の側が必要な動作を推論する例示プログラミングも存在する。例示による言語環境としては、StageCast Creator<sup>☆13</sup> や AgentSheets<sup>☆14</sup> などが知られており、Viscuit はそれらを発展させることでより低年齢からのプログラミングを可能にした。図-5 に、Viscuit の説明画面を示す。

### 適切なフィードバック

初心者のプログラム学習には、適切なリアルタイムのフィードバックが効果的である。構文エラーのフィードバックについては前述したが、意味的なエラーについては、自分の作りたかったものと作成したプログラムの差異を、感覚的に明瞭に確認できることが重要である。

そのためには、単に結果が「3.0」のような文字で出力されるだけでなく、画像としての視覚、現実世界の動作としての触覚、音楽としての聴覚など、さまざまな感覚を通したフィードバックが有効である。

### □ グラフィクス

実行結果が文字だけでなく、視覚的に見えることは重要である。図-6 は LOGO のタートルグラフィクスをドリトルで記述した例である。初期状態が右向きタートルで四角形と三角形を組み合わせる家を描こうとしたときに、三角形を描く左右の向きを間違えると、結果は封

☆11 <http://mindstorms.lego.com/>

☆12 <http://www.baltie.com.pl/>

☆13 <http://www.stagecast.com/>

☆14 <http://www.agentsheets.com/>



カメ太=タートル！作る。  
「カメ太!100 歩く90 右回り」!4回 繰り返す。  
「カメ太!100 歩く120 右回り」!3回 繰り返す。

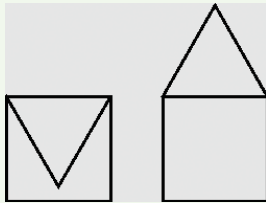


図-6 グラフィクスによるフィードバックの例

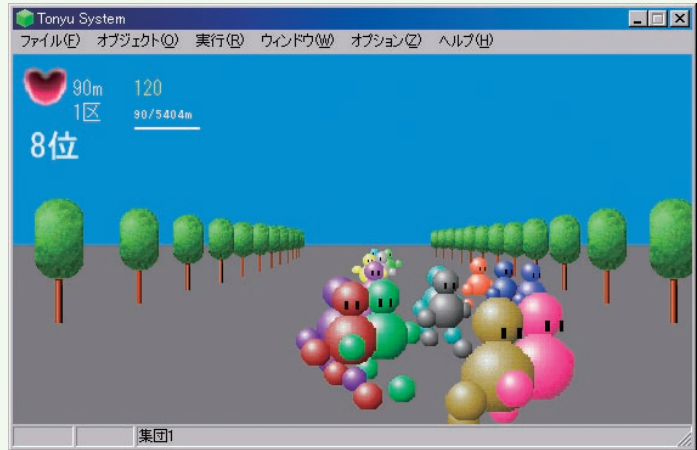


図-7 Tonyu によるゲーム作品の例

筒の絵になってしまう(3行目の「右回り」は「左回り」が正しい)。

自分の記述した内容と望ましいプログラムとの違いが視覚的に一目瞭然であるため、学習者は自発的にプログラムの修正に取り組む。そして完成したときに大きな達成感を感じる。このような、即時の適切なフィードバックは教育用途において効果が大きい。

## □ 制御

特に対象が低年齢の場合には、画面の中の仮想的な世界だけでなく、ロボットのような現実世界のオブジェクトを組み合わせることは有効である。その意味で、画面の外部でロボットなどの実物を動かす制御プログラミングは効果的である。詳しくは本特集の記事を参照されたい。

## □ 音楽

制御と同様に、音という聴覚を活かしたプログラミングも効果がある。ただし、現在では「曲をデータとして演奏する」、「ゲームのBGMとして音を鳴らす」ことが主流であり、プログラムと音楽を効果的に組み合わせた学習は今後の課題である。

なでしこやドリトルでは、鼻歌的にメロディやリズムを記述できる増井俊之氏の考案した「ストーン表記」<sup>☆15</sup>のアイデアを取り入れている。詳しくは本特集の記事を参照されたい。

## 興味と達成感を持つ題材

プログラミングを通して日常利用するソフトウェアの原理を理解するためには、学習者に身近なソフトウェア

を題材にすることが望ましい。多くの成人にとって、ワードプロセッサや表計算は身近なソフトウェアだが、子供にとっての身近なソフトウェアは若干異なっている。プログラミングの対象に興味のある題材を選ぶことは大切であり、教育用の言語にはそれをサポートする環境が求められる。

## □ 小作品

自分で考えたオリジナルのプログラム作品を作れることは重要である。生徒は作品を作る体験を通して、必死に考えたり達成感を感じながら、世の中のソフトウェアがどのように作られているのかを理解する。

また、「自分でソフトウェアを作れる」感覚を持つことは、一生の宝になることを指摘しておきたい。多くの人にとって、自分が「プログラムを書ける」という可能性に気付くこと自体が難しいのである。

## □ ゲーム

子供たちが自発的にソフトウェアと対話する場面として、ゲームは大きな役割を果たしている。与えられたゲームで遊ぶだけでなく、「いつか自分でも作ってみたい」という夢は、潜在的に多くの子供が持っている。そこで、少ない手間と低い難易度でゲームを作れることは、動機付けの面から有効である。図-7はTonyuによるゲームの作品例である。

## □ Web

子供たちがもう1つ慣れ親しんでいるものは携帯電話であり、その画面でWebを閲覧することも多い。HTMLによりWebページを作成する授業が行われることは行われているが、この中にプログラムを埋め込んで動作を定義することも有効である。このような用途にはクライアントで記述して動かせるJavaScriptが適してい

☆15 <http://pitecan.com/Sutoton/>

るが、適切な環境を用意すればサーバサイドのプログラミングも可能である。

### 教育用プログラミング言語の今後

一般の言語と同様に、教育用の言語も進化を続けている。ユーザインタフェースの改良としては、Squeak eToy はキーボードを使わない構造化エディタを実装し、Viscuit のような例示によるプログラミング環境も公開されている。また、社会でのプログラミングの活用という意味で、組み込みと Web でのプログラミング活用の学習は重要なテーマである。

今後、学校教育でのプログラミングの活用を考えた場合には、教育内容に合わせた機能の整備が重要となる。そのために、カリキュラムと、授業のどの場面でどのように使われるのかという分析が求められる。

また、プログラミングを行う前に、カードなどの教具を用いてアルゴリズム的な思考を体を使って体験することは効果的である。本特集で紹介されている Computer Science Unplugged などの教材と組み合わせた教育カリキュラムについても検討する価値がある<sup>2)</sup>。

国内では、1980年代を中心に多くの教員が BASIC や LOGO を体験した。その経験を活かす意味では、当時の N88-BASIC に代表される言語の雰囲気を残した ActiveBasic<sup>☆16</sup> のような実装は、既存の知識で高機能なプログラムを作成できるという意味で評価される可能性がある。

#### 参考文献

- 1) 並木美太郎ほか:「教育用プログラミング言語に関するワークショップ 2006」の報告, 情報処理学会研究報告「コンピュータと教育研究会」2006-CE-85, pp.17-24 (July 2006).
- 2) 兼宗 進: プログラミングのある生活, 日経パソコン PC オンライン, <http://pc.nikkeibp.co.jp/pc/column/kanemune/> (平成 19 年 4 月 16 日受付)

☆16 <http://www.activebasic.com/>

兼宗 進 (正会員)  
kanemune@acm.org

1985 年千葉大学工学部電子工学科卒業, 1987 年筑波大学大学院理工学研究科修士課程修了, 企業勤務を経て, 2004 年筑波大学大学院ビジネス科学研究科博士課程修了, 同年より一橋大学総合情報処理センター助教授。

