

# 学校教育用オブジェクト指向言語/環境の構想について

筑波大学大学院 経営・政策科学研究科 企業科学専攻

兼宗 進, 久野 靖

〒112-0012 文京区大塚 3-29-1

TEL 03-3942-6869 Fax 03-3942-6829

e-mail: {kanemune,kuno}@gssm.otsuka.tsukuba.ac.jp

## 概要

本発表では、学校教育で使うことを想定したオブジェクト指向プログラミング言語の設計およびその実行環境の構想について述べる。プログラミングの専門教育を受けていない教員が限られた時間内で授業を行う中で、「プログラムの楽しさを体験する」「簡単なプログラムを完成できる」「生徒の将来に害を与えない(変な癖を付けない、計算機を嫌いにならない)」ことを目標にする。提案する言語および環境は、「学習用の非力な言語でなく、実用プログラムを作るために十分な表現力を持たせる」「共同でプログラムを作れる言語にする」「試行錯誤しながら学べるようにする」「オブジェクト指向を扱えるようにする。ただし、利用者にオブジェクト指向の複雑さを意識させない」「ネットワークに対応する」といった特徴を持つことを想定している。

## 1. はじめに

新しい学習指導要領により、小学校の「総合的な学習の時間」(2002)、中学校の「技術・家庭科」の情報の内容(2002)、高校の普通教科「情報」(2003)において、コンピュータや情報通信ネットワークを活用した教育が開始される [1]。

これらの情報教育の中で、特に中学校技術・家庭科における計測・制御の学習や、高校における情報科においては、プログラミングに関わる教育が行われることが予想される。実際、情報処理学会が作成した高校の情報用教科書の試作案では、JavaScript、流れ図、C++が使用されている[2]。また、これ以外の教科においても、理科や数学のシミュレーションといった目的で、教員や生徒がプログラミングを行うことが考えられる。

## 2. 教育におけるプログラミング言語

大学や企業といった実社会におけるプログラミング教育と比較すると、学校教育におけるプログラミングには次のような特徴があると考えられる。

- 1) 言語の習得が目的ではない。アルゴリズムの理解や作成したプログラムの利用が目的である。よって、プログラミングの学習に費やせる時間は限られている。
- 2) 専門家を養成することが目的ではない。興味や適性によらず、すべての生徒が理解できることが重要である。
- 3) 授業を行う教員がプログラムに習熟しているとは限らない。プログラムの専門教育を受けていない現行の教員が主体となるため、深い専門知識やプログラミングの経験を期待できない。

これらの特徴を考慮した場合、大学や企業で一般的に使用されている C、C++、Pascal、Java といった言語は、次のような点で適切でないと考えられる。

- ・ コンパイラとしての実装が主体であり、対話的な使用に適していない。
- ・ データ型が厳格であったり、ポインタの概念が必要など、プログラムを組む以前に覚えることが多く、初心者にとっての敷居が高い。

教育現場で利用されている言語としては、Basic[3]やLogo[4]のような汎用の言語や、若葉[5]のような教育用に設計された言語が存在する。しかし、これらの言語はオブジェクト指向の概念を取り入れていない。我々は次の理由から、教育用言語にこそオブジェクト指向の導入が望ましいと考えている。

---

An Object-oriented Language and Environment  
for School Education.

S. Kanemune, Y. Kuno

Graduate School of Systems Management,  
University of Tsukuba, Tokyo.

- 1) 近年のソフトウェア開発においては、自分でゼロからコードを作成して行くよりも、既成のオブジェクトを活用することで短時間に高度な機能を持つソフトウェアを組み立てることが多くなっており、教育においてもそのような側面を体験させたい。
- 2) 従来のスタイルの言語ではせっかく苦勞してプログラムを組んでも、その動作のようすが日常親しんでいるソフトウェア製品とくらべてあまりに落差があるため、プログラムをすることに對する動機づけが難しい。
- 3) オブジェクト指向により GUI 部品やイベント駆動を前提としたプログラムを最初から扱うことで、自分達が接しているソフトウェア製品も実習で書くプログラムと同じ原理によっていることが納得してもらいやすくなる。
- 4) 作成したソフトウェアの操作方法が既存のソフトウェア製品群と同様なものとなるため、教師や生徒有志が教材用のソフトウェアを開発することも行ないやすくなる。

### 3. 設計方針

本研究では、学校教育において学習や教材作成といった教育利用に活用することが可能なオブジェクト指向言語を設計することを目的とする。この言語の実現により、オブジェクトという、本来親しみやすい自然な概念を使ったプログラミングが可能になる。学習者は当初からオブジェクトに親しむことにより、小さなメソッドを組み合わせたプログラムによる構造化や、オブジェクトを単位としたモジュール化を自然と学ぶことができる[7]。

プログラミングの専門教育を受けていない教員が限られた時間内で授業を行う中で、「プログラムの楽しさを体験する」「簡単なプログラムを完成できる」「生徒の将来に害を与えない(変な癖を付けない、計算機を嫌いにならない)」ことを目標とする。

### 4. システム概要

提案する言語および環境は、次のような特徴を持つことを想定している。これらの機能をプロジェクトの中で段階的に実現していく予定である。

- 1) 学習用の非力な言語でなく、実用プログラムを作るために十分な表現力を持たせる。
- 2) 共同でプログラムを作れる言語にする。教室での活動が個人的な作業だけで終わってはおもしろくない。作品を持ち寄って動かせるようにする。この機能はオブジェクトの保存とネットワークの対応によって可能になる。
- 3) 試行錯誤しながら学べるようにする。対話型の実行の中で、適切なガイドや構文チェック、デバッグ環境を提供する。
- 4) オブジェクト指向を扱えるようにする。ただし、利用者にオブジェクト指向を意識させない。
  - a) 目に見えるオブジェクトを画面に表示する(図 1)。標準的なオブジェクトを用意し、たとえばうさぎはカメの 10 倍の速さで動くなど、それぞれの属性を与えておく。これらのオブジェクトに能力を授けていくことがプログラムの中心になる。そのため、自然とオブジェクトを単位としたプログラムになる。
  - b) プロトタイプベースのオブジェクト指向を採用する[8]。このことにより、「クラスとインスタンス」といった抽象的な概念を使わずに、同等の表現能力を使えるようにする。たとえば画面上的オブジェクトを複製し、それを拡張することで、実質的に継承を含めたクラス階層を実現する。
  - c) オブジェクトの保存を可能にする。これにより、作品としてのオブジェクトを、姿などの属性を含めて配布することが可能になる。
  - d) 画面上で動作が見えるオブジェクトに加えて、プログラムの部品として使うラ

イブラリ的なオブジェクトを扱えるようにする。これによりプログラムの再利用を促進する。

- 5) ネットワークに対応する。
  - a) Web ブラウザ上で実行できるようにする。インストールが不要なため、気軽に実行できる。
  - b) オブジェクトをインターネット上で公開して配布できるようにする。
  - c) LAN 上で、オブジェクトの交換を可能にする。同級生のオブジェクトをコピーして改良したり、先生が生徒のオブジェクトを集めて実行できる。

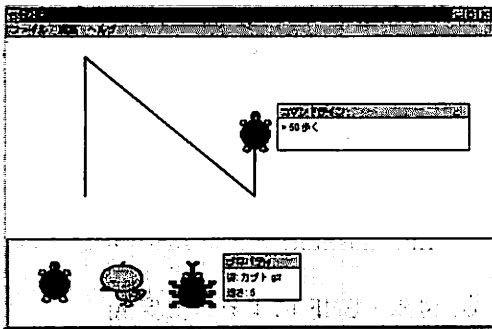


図 1 実行の概念図

## 5. 言語仕様

次のような方針で言語の設計を進めている。

- 1) 少ない原理で一貫していること。覚えることが少なく、類推して使えること。

- 2) 具体的であること。
  - ・ 型やクラスを扱わない。
  - ・ 目に見えるオブジェクトがプログラムを持つ。
  - ・ ライブラリを目に見えるオブジェクトにする。
  - ・ 作ったプログラムを部品として保存できる。
- 3) 初心者には敷居が低いこと。
  - ・ 記号を少なくする。
  - ・ 命令語、変数名、記号などに日本語を許す。
  - ・ 記号の文字種を区別しない。たとえば「,」「,」「,」を同一に扱う。
  - ・ 変数に型がない。
  - ・ 変数や手続きの宣言が不要。
  - ・ シンプルなデータ構造。
  - ・ シンプルな制御構造。

上記の方針の下で、次のような言語を設計した(図 2, 図 3)。

- 1) メッセージ送信を基本とする。
  - ・ 「オブジェクト, 引数 引数 … 命令」という構文で記述する。
  - ・ 命令の結果に対して、さらに「引数 引数 … 命令」を後続して記述できる。
  - ・ 引数には数値、文字列、ブロックを記述できる。それ以外は括弧で囲んで記述する。
  - ・ 括弧内には中置記法かメッセージ送信の

```

カメ太 = タートル, 作る;
カメ太.長方形 = [,100 歩く 50 右に歩く 100 戻る 閉じる];
長方形 1 = カメ太, 長方形 "青" 塗る;
長方形 2 = 長方形 1, 複写 50 右に移動 "白" 塗る;
長方形 3 = 長方形 2, 複写 50 右に移動 "赤" 塗る;
  
```



図 2 図形描画の例(フランス国旗)

```

カメ太 = タートル, 作る;
カメ太.斜めの長さ = [|x y| (x * x + y * y), √];
カメ太, 3 4 斜めの長さ;
⇒ 5
  
```

図 3 数式の例(直角三角形の斜辺)

いずれかを記述し、この中で変数を参照できる。

- 2) オブジェクトは既存のオブジェクトを複製して作る。
  - ・ サンプルのオブジェクトを画面下部に用意する。
- 3) オブジェクトはスロットを持つ。
  - ・ スロットには型がなく、定数、ブロック、オブジェクトなどに入れられる。
- 4) オブジェクトはメッセージを受け取る。
  - ・ 相手に呼びかける形にする。  
(例: 「カメ太、100 歩く」)
- 5) タートルグラフィックスの機能を持つ。
  - ・ 描画の開始から「閉じる」までの間を閉領域として扱う。
  - ・ 閉領域がひとつのオブジェクトになる。

## 6. 現状と今後の予定

現在は、検討中の言語仕様を取り入れたプログラム環境を試作中である。

設計の段階から十数人の小中高の教員に意見を聞きながら進めている。今までに、次のような意見が寄せられた。

- ・ 最初は後置記法に違和感を感じたが、説明を聞くと「命令を日本語にして助詞を頭の中で補えばかえってわかりやすい」ことがわかった。
- ・ 数式は学校で使っている記法、または生徒や教師が頭の中で考えている記法に近いことが望ましい。

今後は、寄せられた意見を反映して処理系の改良を進め、クラブ活動や授業において有効性の検証を行う予定である。

## 7. 参考文献

- 1) 文部省: 学習指導要領,  
<http://www.monbu.go.jp/news/00000317/>
- 2) 情報処理学会初中等情報教育委員会 WG,  
高等学校普通教科「情報」試作教科書,  
<http://www.ics.teikyo-u.ac.jp/InformationStudy/>
- 3) 白石和夫: Windows95 で動作する Full

Basic 言語処理系,夏の情報教育シンポジウム(SSS99)予稿集,情報処理学会,1999

- 4) シーモア・パパート:マインドストーム, 未来社,1982
- 5) 吉良ほか: プログラミング入門教育のためのプログラミング言語「若葉」とその処理系,夏の情報教育シンポジウム(SSS99)予稿集,情報処理学会,1999
- 6) 片桐明: 日本語プログラミング言語 Mind, 翔泳社, 1988
- 7) M.Kolling, B.Koch and J.Rosenberg, *Requirements for a First Year Object-oriented Teaching Language*, ACM SIGCSE'95, 1995, pp.173-177
- 8) D.Ungar and R.Smith, *Self: The Power of Simplicity*, ACM OOPSLA'87, 1987, pp.227-242
- 9) A.Goldberg and D.Robson, *Smalltalk-80: The Language and Its Implementation*, Addison-Wesley, 1983

## 8. 付録. BNF による構文定義

```
プログラム ::= (文 ';')…  
文 ::= [変数 '='] 式  
変数 ::= [項 ':'] 名前  
項 ::= 単純式 | 変数  
式 ::= 単純式 | 送信  
送信 ::= [項] ';' 電文  
電文 ::= 単純式… 名前 (';' 単純式… 名前)…  
単純式 ::= 整数 | 文字列 | 括弧 | ブロック  
括弧 ::= '(' 中置式 ')' | '(' 送信 ')'  
中置式 ::= 中置式 演算子 中置式 | 項  
ブロック ::= '[' '[' 名前… '|' 文 (';' 文)… ']'
```

ただし X|Y は X または Y。

- [X] は X があってもなくてもよい。
- X… は X の 0 回以上の繰り返し。
- (X) は X とおなじ(グルーピング)。