

# 初心者向きアルゴリズム学習環境の構築

中村 亮太<sup>†</sup> 松浦 敏雄<sup>†</sup> 西田 知博<sup>††</sup>

<sup>†</sup>大阪市立大学 大学院創造都市研究科 都市情報学専攻

<sup>††</sup>大阪学院大学 情報学部

mail: m04uc507@ex.media.osaka-cu.ac.jp

## 1. はじめに

中学、高校での情報教育が必修となり、大学における情報教育の見直しについて活発に議論されている。その中で、我々は特にプログラミング教育の重要性に注目している。

ここでいうプログラミング教育は、特定のプログラミング言語の修得を目指すものではなく、コンピュータの仕組みの理解を深めること、および、ルールに従って論理的に記述することを体験することが目標と考えている。

プログラミング初心者がプログラミングを学習する際に、プログラミング言語の細かな文法によるエラーに悩まされることが多い。カンマ1つを忘れたために、膨大なエラーメッセージが表示され、それだけで拒絶反応を示す学習者も少なくない。また、一般に言語仕様が複雑であるので、言語の修得に時間をとられることが多い。

初心者向けのプログラミング学習環境を提供する研究として、ドリトル[1]やNigari[2]などがある。これらの学習環境は、初心者でも分かりやすいようなさまざまな工夫がなされている。また、これらはいずれも、導入時点から、オブジェクト指向の考えでプログラミングの学習を行うものである。オブジェクト指向の考え方は極めて重要ではあるが、メソッドの記述は手続き的に書かざるをえず、オブジェクト指向の考えに至る前段階で手続き的なプログラミングの能力が必要と思われる。

本研究では、初心者が文法エラーや論理エラーに悩まされずに、アルゴリズムの理解を重点的に学習できる、初心者向けのプログラミング学習環境を構築した。

## 2. 学習環境の概要

### 2.1 言語仕様

プログラミング初心者でもプログラム全体の構造を理解できるよう、言語仕様は単純かつ、学習しや

すいものとした。用意する文は、代入文、制御文(条件分岐、繰り返し)、関数呼び出しのみを考える。図1は、プログラミング学習環境の言語を用いて記述した、階乗を求めるプログラムの例である。

関数呼び出しは、画面への出力や、sin 関数など計算に必要なもの、乱数発生のための関数などをライブラリとして用意する。また、プログラムでの関数定義も可能にしておく。変数は、整数、実数、文字の型があるが、型の宣言なく使用することもできるようにする。

```
1: 整数 seisu;  
2: 整数 i;  
3: 整数 fact;  
4: seisu <- 入力; //キーボードからの入力  
5: fact <- seisu; //seisuをfactに代入  
6: i <- seisu - 1;  
7: 条件(i >= 1)が成立している間以下を繰り返す{  
8:     fact <- fact * i;  
9:     i <- i - 1;  
10: }  
11: 出力 (seisu, "の階乗は", fact, "です。");
```

図1 階乗を求めるプログラム例

### 2.2 GUIによるプログラミング

プログラムの組み立て方を重点的に学習するために、制御構造が書かれたブロックを、GUIのドラッグ&ドロップ操作で、プログラムを記述できるようにしている(図2)。これにより、キーボードからの入力を減らせるので、文法エラーの発生を抑えることができる。積み木を組み立てるように、容易にプログラムを作成することが可能なので、プログラミング初心者はプログラムの組み立て方を重点的に学べる。また、GUI操作で組み立てていくだけではなく、キー入力でプログラムを記述することが可能となっている。

## 2.3 言語表現の切り替え

プログラム構造を理解するために、プログラム言語は日本語の表現を用いていたが、段階が進めば、実際のプログラム言語で学習することも考えていかなければならない。そこで、図3のように、言語表現を日本語から、英語の言語表現へ相互に切り替えられるようにする。切り替えは、ボタン操作1つで動的に行えるようにする。表現の切り替えにより、実際のプログラム言語演習への、橋渡しになると考えている。

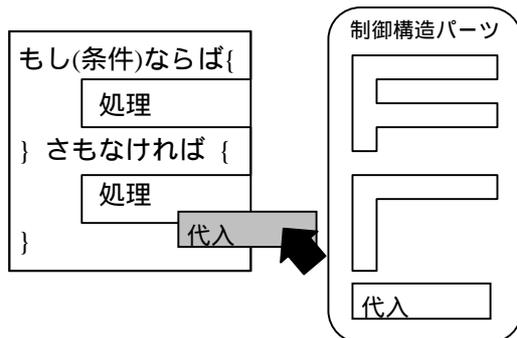


図2 GUIによる制御構造の組み立て

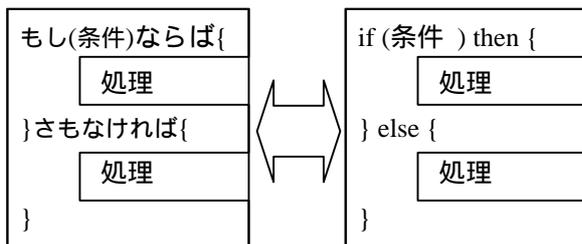


図3 言語表現の切り替え

## 2.4 プログラムの実行

プログラム実行の流れを理解できるよう、1行ずつ実行できるステップ実行や、実行速度を調節し実行できるスロー実行の機能を持つ。実行時には、どの行が実行されているかを表示し、各変数にどのような値が代入されているかを見られるようにしている。このような、トレース機能により、論理エラーの修正を容易に行えるようになる。また、プログラムの流れが追えるので、プログラム全体の構造を理解しやすくなると考える。

## 3. 実装

本システムは Java 言語を用いて作成した。また、日本語と英語の言語表現で記述されたプログラムの解析と実行には、JavaCC[3]と、そのプリプロセッサである JJTree を使用した。JavaCC は Java 向けのコ

ンパイラコンパイラで、BNF 記法による文法定義を入力とし、Java のクラスソースコードに変換するものである。JJTree は JavaCC と組み合わせる使用することにより、構文解析の結果からツリー構造を作るプログラムを自動生成してくれるものである。

図4はシステムの画面表示例である。プログラム編集領域はプログラムのソースの入力・編集および、編集したプログラムの保存・読み込みが可能である。コンソール領域では、キーボードからの入力を受け付け、実行結果の出力が行える。変数表示領域は、実行中の各変数の値を表示する。

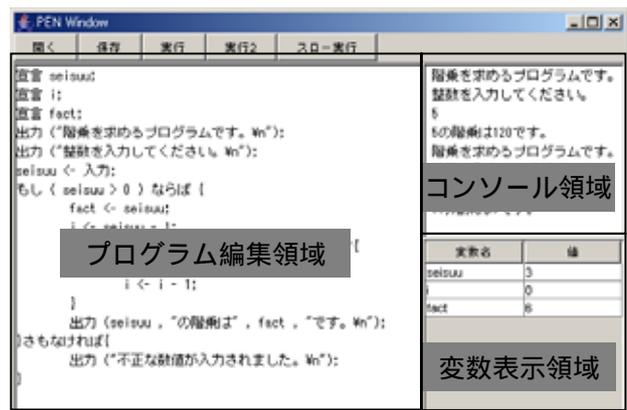


図4 システム実行時の画面表示例

## 4. おわりに

本研究では初心者が使いやすい、プログラミング学習のための環境を開発した。

今後、GUIでのプログラム組み立てを実装し、実際のプログラミング演習での評価を考えている。また、学習者がより一層、興味を持てるような工夫として、プログラムの実行結果を、テキスト出力のみではなく、グラフィックも扱えるようにしたいと考えている。

## 参考文献

- [1] 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.44, No.SIG13, pp58-71 (2003).
- [2] 長慎也、甲斐宗徳、川合晶、日野孝昭、前島真一、筧捷彦:「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「コンピュータと教育」, No.071, pp.13-20 (2003).
- [3] “Java Compiler Compile[tm] - The Java Parse Generator”. (<https://javacc.dev.java.net/>)