

プログラミング教育における

4G-4

制御構造理解のためのイメージ形成の効果

長谷川 聡 山住 富也

名古屋文理短期大学

1. はじめに

手続き型の高級言語によるプログラミング技能の修得を目指す上で、選択・反復などの制御構造の理解は不可欠である。

筆者らは、短期大学の情報処理専門課程において講義と演習によるプログラミング教育を実施している。使用言語はPascalおよびC言語であるが、プログラミング経験の浅い学生が学習の初期の段階でプログラミング技能の修得に挫折する場合、その理由のひとつは、プログラムの制御構造を追うことが出来ないからであると感じている。

プログラムを作成したり既存のプログラムの処理内容を理解したりするには、プログラムの実行時における処理の流れが正しくイメージできる必要があると思われる。

2. プログラミングとイメージについて

筆者らは、プログラミング技能の修得とイメージの形成の関係について、次のような仮説を立てた。

- ①プログラミング技能の修得のためには実行時の処理の流れが何らかの形でイメージできる必要がある。
- ②処理の流れのイメージには、例えば分岐や反復といった制御の種類ごとに、特定の言語の書式に依存しない共通のイメージがあり得る。
- ③イメージの種類、持ち方は、人によって異なり、個々のプログラムの流れを正確になぞるようなものである場合、抽象化された制御構造のイメージである場合、また、抽象的な処理の流れを反映した具体的な例示である場合などがある。

④イメージを持っているか否か、明確なものか漠然としたものかには個人差があるが、より明確なイメージをもっている方が、個々のプログラムの理解能力が高い。

⑤具体的なイメージを強制的に持たせることでプログラミングの学習効果の向上が期待できる。

以上のように、プログラミングにおけるイメージ形成が重要であるとの認識から、プログラミング教育を受けている学生を対象に調査を行った。以下に、調査の結果を報告し、制御構造理解に対するイメージ形成の効果について考察する。

3. プログラムの読み取りテストとアンケート調査

短期大学の情報処理専門課程の学生を対象に、簡単なプログラムの読みとりテストと、制御構造のイメージに関するアンケートを実施した。

[問] 実行後のyの値を答えなさい

```
y:=1;

for k:= 6 down to 3 do
  y := y * k ;
```

図1. プログラム読みとりテストの問題例

プログラムの読みとりテストは、図1のような問題を数問出題し、十分な時間をかけて解答するもので、Pascalの言語仕様をとってはいるが、細かな構文知識は問わないよう配慮している。なお、今回の調査対象の学生はいずれもPascal言語の講義を受講している。

また、イメージに関するアンケートは、制御構造のイメージを持っているかどうかを問うとともに、いくつかの基本的な制御構造について、そのイメージを絵や図で自由に表現してもらうものである。

回答されたイメージ表現は様々であるが、次のようなタイプに大別できた。

- 二分岐(if文)のイメージ
 - ・分かれ道型 ・枝分かれ型 ・二者択一型
- 多分岐(case文)のイメージ
 - ・並列枝分かれ型 ・木構造型 ・二分岐入れ子型
- 指定回数繰り返し(for文)のイメージ
 - ・ループ型(図2) ・階段型(図3) ・例示型
- 前(後)判定型繰り返し(while文, repeat文)
 - ・領域型 ・条件判定反復型

これらの他に、「流れ図型」(フローチャートやPADを使って説明している場合)や「イメージ無し」(白紙または意味不明)を加えて分類した。

アンケートの結果と読み取りテストの成績の関係をみると、二分岐・多分岐の選択構造では、ほとんどの学生が比較的明確なイメージを示すことができテストも高得点でイメージの種類と成績の相関は認められなかったが、反復構造では、表1・表2に示すようにイメージの有無や種類とテストの得点との相関が得られた。

表1. 反復構造(for文)のイメージの有無と読み取りテストの成績(対象102人)

イメージの有無	人数(人)	得点(4点満点平均)
ある	20	2.75
ない	82	2.24
合計	102	2.34

表2. 反復構造(for文)のイメージの種類と読み取りテストの成績(対象47人)

イメージの内容	人数(人)	得点(5点満点平均)
ループ型	6	4.33
階段型	4	3.00
例示型	14	2.43
流れ図型	10	3.30
イメージ無し	13	2.08
合計	47	2.81

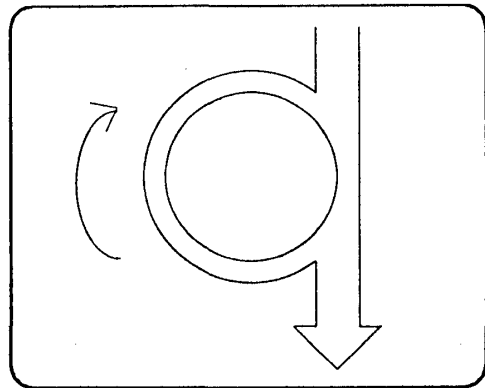


図2. for文のイメージ「ループ型」

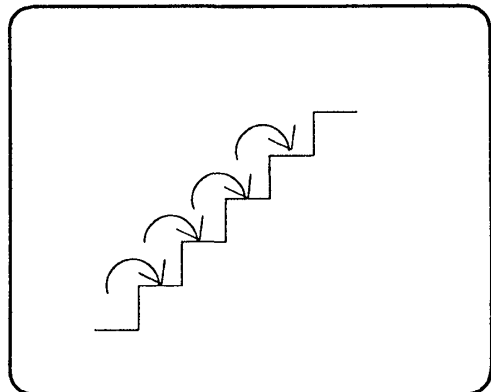


図3. for文のイメージ「階段型」

調査から、調査対象の学生については、反復やその組み合わせのような制御構造では、より明確な、しかも処理の流れを抽象化したタイプのイメージを持っている場合ほど理解度が高いことが分かった。

また、これは、プログラミング能力育成へのイメージ形成の応用の可能性を示唆している。

4. おわりに

初心者を対象にプログラミング教育を行う場合、単にプログラミング言語の文法を教えるのではなく、プログラムの実行時のイメージを持たせることが有効であると思われる。また、持たせるイメージとしては、実行時のプログラムの動作の流れを抽象化したものが良いと思われる。

今後、具体的な、イメージの形成法として、制御の流れをビジュアルに表すアニメーションなどを用いたCAIシステム^{*}の開発・適用を検討したい。

[参考文献]

^{*}長谷川、山住:「情報処理の専門基礎科目におけるCAI導入効果の評価」,情報処理学会第50回全国大会(1994)