

C言語プログラミング教育方法

福山職業能力開発短期大学校

白川 浩・平島 隆洋

日浦 悦正・山下 明博

Programming Training Technique using C language

Hiroshi SHIRAKAWA・Takahiro HIRASHIMA
Etsumasa HIURA・Akihiro YAMASHITA

要約

情報システム導入時にオープンシステムを採用する傾向が強まるに従い、C言語によるプログラミングを修得する必要性が高まってきた。こうした背景から、コンピュータ利用の初心者でもC言語によるプログラミングを学習するようになってきた。C言語によるプログラミングを教育する方法は様々であるが、学習者が目標達成という成功体験を積み重ねることにより学習意欲を刺激、増大できるような方法がよい。また、実用を考えると構造化プログラミング技法やシステム設計技法と関連づけて学習するとより効果的である。こうした考えから、単なるC言語文法学習用の課題ではなく、プログラミングに役立ついろいろな技法も合わせて学習できるような課題を作成し、プログラミング教育に使用する方法を提案する。本校に於いて、コンピュータ利用の初心者に対し、本方法を利用してC言語プログラミング教育を行った結果、本教育方法が有効であることが確認できた。

1 はじめに

情報システムの開発費や維持費、そして、ユーザ教育費など必要経費をできるだけ抑え、かつ個人レベルの事務処理作業なども効率よく行えるような情報システムを構築するため、製造業者や機種に依存しないシステム仕様を採れるオープンシステムを採用するユーザが増加している。このようなユーザの大半は、情報システムのOS (Operating System) としてUNIX系OSを採用している。UNIX系OSは、そのほとんどがC言語により記述されている。このため、システム記述言語としてC言語の重要性は、極めて高くなってきている。従って、コンピュータ利用の初心者でも、C言語をよく理解し、システム開発やメンテナンスに利用できるレベルまで到達することが重要である。

ところが、テキストとして市販されているC言語解説書やプログラミング演習書の大部分が、文法解説文書に沿った簡単な例題や課題のみで構成されているという現状である。このようなテキストをもとに教育を

実施した場合でも、例題と同じパターンで処理できるような簡単な問題への応用程度ならば、学習者自身が独力で行えるようになる。また、指導者に要求される能力や使う労力も少なくすむ。

しかし、学習者がより複雑な実際問題への応用を独力で行えるようになるまでにはかなりのセンスと追加学習や経験が必要となる。このため、例題パターンの手直しという単純な作業を繰り返させるだけの指導方法では、多くの学習者が学習後も独力でプログラミングできるようになれないという問題があった。また、これまで入門用言語としてよく利用されてきたBASIC言語によるプログラミングなどと比べても、応用するのが難しいとかあるいは楽しみが少ないといった感想を学習者が抱くことが多かった。

本報告では、この様なC言語の学習者が、学習意欲を向上させつつ効率よく学習できるだけでなく、実用的なシステム開発技法の学習もできるような教育方法を提案する。また、本方法を利用したC言語プログラミング教育の結果についても報告する。

II 方法

これまでの実習方法は、多くのC言語解説書やプログラミング演習書に見られる様なC言語についての文法解説と文法事項確認用の基本プログラミング演習課題とから構成されたテキストを利用し、実習のたびに一定程度の文法解説の講義をした後、それに関する基本的なプログラミング演習課題を少しずつ繰り返して行かせていた。このような実習方法をやめ、文法事項の解説はまとめて最小限の範囲で行う。このため、プログラムのスタイルを重視した概略説明を初めの1コマ程度の時間で行うこととし、あまり細かいことには触れないようにする。この説明の後、プログラミング演習課題をまとめて提示する。以後、学習者は提示された課題の要求に応じたプログラムを完成するべくプログラミングを行う。これを全ての課題が終了するまで繰り返して行う。尚、提示する課題の作成に際しては、学習者が学習意欲を維持向上できるように、課題達成という成功体験を積み重ねることが可能であるような課題とする。このため、以下の指針に従うものとする。

- (1) 単純な文法事項の確認課題とはしない。
- (2) 簡単なものから複雑なものへと段階的にする。
- (3) 視覚的に動作確認ができるようにする。
- (4) 遊びの要素を入れる。
- (5) プログラミングに有効な技法を含める。
- (6) 学習者が応用を考えるようにする。
- (7) エラー処理を含める。

この様にして作成した課題は、各学習者が課題を終了するごとに、指導者が完成度を確認する。その際、指導者は、学習者自身に作成したプログラムについて説明させる。もし完成度が低い場合は、完成度が高まるように指示を与え再考させる。また、エラー処理などで、考慮すべきプログラミング技法や注意事項があれば、課題の進度に応じて補足的に説明する。このほか、プレゼンテーションの技能を向上させるため、学習者にプレゼンテーションの機会を与え、自分のプログラムについて説明させる。

本指針に従って作成した課題のうち、過去2年間(1991年度と1992年度)に渡り本校情報システム系1年生(情報処理科と情報技術科、各科約20名)が受講した構造化プログラミング実習(8単位)のなかで使用した課題について、課題別に技術要素を分類した表を表1に示す。また、実際に使用した課題の例として、1992年度前期課題LB7を図1に示す。尚、1993

年度は教官の担当教科分担の関係から担当教官を変更せざるをえず、従来の方法で実習を行った。このため、本方法に従った課題の作成は行わなかった。

表1 課題別の技術要素分類

(◎: 初出要素 ○: 含まれる要素 - : 含まれない要素)

技術要素	LB1	LB2	LB3	LB4	LB5	LB6	LB7	LB8	LB9	LB10
表示	◎	○	○	○	○	○	○	○	○	○
数字入力	◎	○	○	-	○	-	-	○	○	○
例外処理	-	◎	○	○	○	○	○	○	○	○
繰り返し	-	◎	○	○	○	○	○	○	○	○
メニュー	-	-	◎	-	○	-	-	-	-	○
文字入力	-	-	-	◎	-	○	○	○	○	○
ゲーム判定	-	-	-	-	◎	○	○	-	-	○
直線移動表示	-	-	-	-	-	◎	○	-	-	○
らせん移動表示	-	-	-	-	-	-	◎	-	-	○
重なり表示	-	-	-	-	-	-	◎	-	-	○
構造体	-	-	-	-	-	-	-	◎	○	○
ファイル入力	-	-	-	-	-	-	-	-	◎	○
ファイル出力	-	-	-	-	-	-	-	-	◎	○
グラフィック	-	-	-	-	-	-	-	-	-	◎
図形移動	-	-	-	-	-	-	-	-	-	◎
プロジェクト企画	-	-	-	-	-	-	-	-	-	◎

[LB7] (らせんコース)

キーボードから、最大4人分のプレイヤーの名前を入力し、各プレイヤーにそれぞれ、1から4の数字を対応させて表示することにする。入力後、乱数で発生させたいころの目の数(1から6)の分だけ、らせん状にコースを移動する。50右に移動したら、下、左、上の順に移動し、最終的にらせんコースのまん中にあるゴールまで移動したら、その数字はそこで停止させる。移動中は、現在位置を、スコアとして各プレイヤーの右に表示する。複数のプレイヤーが同じ位置に移動したときは、数字の表示は、数字の合計で示すこととする。また、らせんコースの途中で特別な点を6点乱数で決め、プレイヤーがその点に停止したときは、色を変え、再度さいころを振って、その目の数だけ、コースを移動する。最後に4つの数字が停止したら、4人のプレイヤーがどの順番でゴールしたかを画面上に表示する。

<プログラム実行例>

```
Input player(1)'s name ==> SHIRAKAWA
Input player(2)'s name ==> HIURA
Input player(3)'s name ==> YAMASHITA
Input player(4)'s name ==> HIRASHIMA
```

```
NAME      Score .....*.....*.....*.....*.....*.....*.....*
SHIRAKAWA 92 .....*.....*.....*.....*.....*.....*.....*
HIURA     86 .....*.....*.....*.....*.....*.....*.....*
YAMASHITA 80 .....GOAL*.....*.....*.....*.....*.....*.....*
HIRASHIMA 80 .....*.....*.....*.....*.....*.....*.....*
*.....*.....*.....1*.....2*.....7*.....*.....*
```

図1 1992年度前期課題LB7

課題数は前期分(4月より9月まで)が7課題、後期分(10月より2月)が3課題である。最終課題はミニプロジェクトとし、指導者の規定した条件を満たす題材を、学生自身が考えるような課題とした。課題実施に先がけて、C言語の文法とC言語プログラミング書式(構造化プログラミング)について約60分の簡単

な説明をした。その後、数人の学生が課題LB3を実施する頃、エラー処理や配列・ポインタについて40分程度の補足説明をした。そしてミニプロジェクトの実施に際しては、構造化システム分析・設計の手法(SA・SD)について約30分程度の補足説明をした。

実習はパソコン上でBORLAND社のTurbo C (V2.0)を使用して行った。課題LB1から課題LB9の完成度確認は、エラー処理を含むプログラムの記述内容確認とプログラムの動作確認およびプログラムの処理内容をHCP^①(Hierarchical and Compact description chart)で記述させた報告書の確認とで行った。この確認作業は、実習時間中に個人ごとの進度に合わせて課題別に細かく実施した。また、課題LB10(最終課題)実施に際しては、プログラム作成のほか提案書、外部仕様書、内部仕様書および取り扱い説明書の作成も併せて行わせた。これらドキュメント類の作成に加えて、提案についてとプログラムについての2回に渡るプレゼンテーションを行わせた。尚、本校での情報システム系1年生に対する実習の指導は、学生に対する指導を細かくできるように、教官2名体制で行っており学生個人ごとの進捗状況の管理がかなりの確にできた。

III 結果

1991年度と1992年度に本方法を利用してC言語プログラミング教育を実施した結果、構造化プログラミング実習を受講した情報システム系1年生は、全員、期限内(4月中旬から翌年2月中旬まで)に全課題を終了することができた。

1991年度と1992年度に実施した実習の最終課題(LB10)に対する解答として得られたプログラムの分類を図2に示す。図2からわかるように、この課題を実施するに際して学生が選んだ題材は、大きく4つに分類できたが、個人ごとに内容やレベルが違っておりそれぞれ特徴あるものとなった。また、本教育方法の適用により、C言語に関する習熟度がどの程度増加したかを評価するため1992年度学生(本方法)と1993年度学生(従来方法)に対し、文法力、読解力、作成力について簡単な確認試験を行った。試験は、同一問題を使用し、同一解答時間(90分)で解答してもらった。確認試験の内容別正解率を、図3に示す。図3から明らかかなように、本方法の利用により習熟度が増加している。

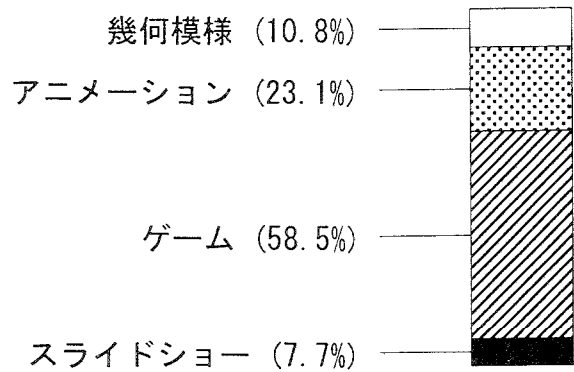


図2 プログラムの分類

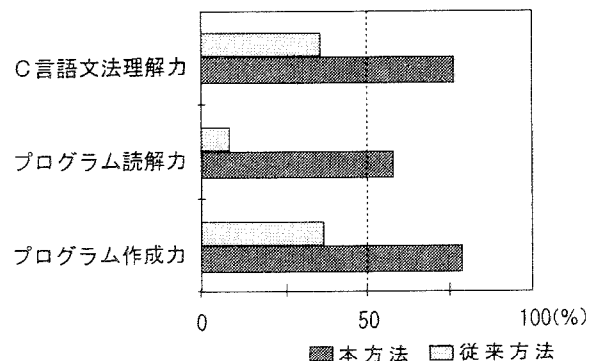


図3 試験内容別正解率

IV むすび

本教育方法の利用により、C言語やプログラミングに関する習熟度が増加することが確認できた。初期段階では、多少の困難をとまなう学習者も一部いたが、学習者が課題達成の充実感を持つようになるため旺盛な学習意欲を維持することができ、慣れるに従って順調に課題を消化することができた。このため指導者は、初期段階での落ちこぼれを出さないように、よく注意することが大切である。

最終課題に対する解答として得られたプログラムの分類でゲームの割合が高かったのは、取り組みやすさや興味深さのほか、今回作成した課題で簡単なゲームを作成するものが多かったためと考えられる。

今後、今回のように、初学者向けにゲームを中心としたプログラム作成を行う課題だけでなく、他言語の経験者向けに、回路計算など工学的な応用を中心としたプログラム作成を行う課題も作成する予定である。また、オブジェクト指向言語の概念を効率よく学習していくための課題も作成していく予定である。

参考文献

- (1) NTTソフトウェア研究所ソフトウェア開発技術
研究部発行：“HCPチャート記述ガイドライン”。
- (2) B.W. Kernighan and D.M. Ritchie: “The C
programming language”, PRENTICE-HALL,
(1981).
- (3) SO/IEC9899: “1990 Programming language
C”, pp178 - 195.
- (4) “Turbo C reference (V2.0)”, BORLAND,
(1989).
- (5) Roger T.Stevens and Christopher D.Watkins:
“ADVANCED GRAPHICS PROGRAMMING
IN C AND C++”, M&T Publishing, (1991).
- (6) 河西朝雄：“C言語”，ナツメ社，(1990)。
- (7) 中山 茂：“Turbo C, Turbo C++グラフィック
スプログラミング入門”，技報堂出版，(1992)。
- (8) Allen B.Tucker and Bruce H.Barnes: “Flexible
Design: A Summary of Computing Curricula
1991”, IEEE COMPUTER, (Nov.1991).
- (9) 日本学術会議 情報工学研究連絡委員会：“情報
工学の体系化に向けて”，電子情報通信学会誌，
Vol.75 No.5, pp443 - 449 (May 1992).
- (10) 大学等における情報処理教育検討委員会CS分
科会：“大学等における情報系専門教育の改善へ
の提言”，情報処理学会誌，Vol.32 No.10, pp1079
- 1091 (Oct.1991).
- (11) 都倉信樹：“情報処理教育における実験・演習”，
情報処理学会誌，Vol.32 No.10, pp1101 - 1108
(Oct.1991).