パターンによるプログラミング授業教材の開発

三　河　佳　紀*

Development of Teaching Materials Based on Patterns for Class in Programming

Yoshinori MIKAWA

要　旨

プログラミングの授業において苦手意識を持つ学生に対し苦手分野の分析を行った。それらを克服させるための補助教材を苦手分野に存在するパターンに着目し，彼らの手助けと成るべく開発を試みた。本報ではその補助教材の操作方法や内部処理について報告する。

Abstract

Programming is one of some student weakest subjects. The author tried to find the main cause for weakest subjects in them. Analysis showed that it contained some problems. The author has developed a teaching materials based on patterns for class in programming. The index in the teaching materials will help student locate the section he want to learn. In this paper some operation for this system, and the inner processes within the system, are reported.

1．はじめに

著者は平成5年度から情報工学科低学年を対象としたプログラミングの授業を担当している。中学から高専に入学し，言語教育のスタート時点において学生は当然のことながら，理解度での差異は皆無である。しかし，学年が進むにつれてその差異は著しくなる傾向がある。その要因は種々考えられるが，分析の結果ある時点からその傾向が顕著に表れることが判明した。その要因を取り除き，少しでも苦手意識を克服させるためにプログラミング授業における補助教材を作成することを試みた。この補助教材は留学生への補習授業にも効果的である。本報ではその開発に関する経緯と，教材の概要を紹介する。

2．低学年における言語教育の変遷

本校情報工学科は，平成2年度に新設された学科である。言語教育については，平成5年度までは低学年ではプログラミング言語にPascal言語を，高学年においてはC言語を用いることにより行われてきた。平成6年度より1学年からプログラミング言語としてC言語を用い，各学年共通してC言語を扱うようになった。これについてはPascal言語でアルゴリズム教育を定着させてから，他の言語に移行させるという考えも当然考慮したが，卒業研究や実験等でC言語，あるいはC++などのオブジェクト指向言語を扱うことが多いことなどから，低学年からC言語の定着化を目指した。アルゴリズム教育等については別の科目で扱うこととした。

PascalからCへ移行した学年については，当初戸惑いもあったものの，プログラミングに必要な基礎的な部分についてはすでに確立されていたため，特に問題無く移行が行われてきた。高学年に進むと実験や卒業研究等で多種多様な言語を扱うようになるが，低学年のこの時期における言語教育は高学年に向けてのプログラミング能力形成の大切な時期にあたる。C言語を扱う実習環境は，平成10年度より一斉にオペレーティングシステムをUNIXに変更して行っている。

平成11年度のプログラミング関係の科目一覧を表1に示す。現在ではカリキュラムも一部改正され，アルゴリズム関係もプログラミングの授業の中で行っている他，プログラム設計関係について

＊　講　師　情報工学科

表1　低学年プログラミング関係科目一覧

学　年	科　目	時間数
1	プログラミング	2
2	プログラミング	2
3	プログラミング	3
3	プログラム設計演習	2

は，3学年のプログラム設計演習という科目で扱っている。表1を見ると，1学年・2学年においてはC言語の基本的な文法関係を学ぶため，通年週2時間の講義がなされ，3学年になると2科目で週5時間の講義が行われている。3学年の学生は2学年までの基礎的技術を更に発展させた，アルゴリズム関係を交えての実習と，プログラム設計の概念と実践について学ぶことになる。

3．苦手分野の分析とパターン化

実習系の科目であるプログラミングの授業では，学生の苦手分野が顕著に現われる場合がある。著者がプログラミング授業を担当している平成5年度からの学生におけるプログラミング技術の習得度状況をまとめたものを，図1に示す。著者の受け持つ学年が，年度により異なる場合があり，必ずしも同一条件とは言えないが，その判断基準として学生の小試験をはじめとする各種試験，課題等における分野ごとの達成度により，評価をA（習得度が高い），B，C，D（習得度低い）に分類し，1年間の平均を取ったものである。ただし平成11年度については前期分のみのデータである。

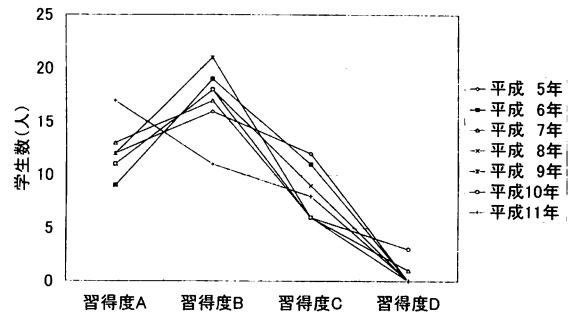

図1　プログラミング技術の修得度

なお，これらの評価は学業成績には無関係である。あくまで独自の方法で分野別達成度基準を設け，それに照らし合わせての評価である。例えば「片方向のリストを作成し…」という課題があれば，学生の作成したプログラムが仕様を満足し，正常に動作したならばAという評価を行うという具合である。

図1より，各年度の傾向はほぼ同様の形態を示していることが，読み取れる。習得度別で見ると，習得度A・Bについてはプログラム作成能力に関して特に問題がない学生である。習得度Cについては，プログラミングに必要な基本的部分の理解度が曖昧な為，自力でのプログラム作成能力に少々難がある学生である。習得度Dについては稀であるが，プログラム作成能力が極端に劣る学生である。習得度が低いCとDの学生については著者の授業では例年多い時で20～25%程度である。C・Dの学生は，いったいどのようなことが要因となってこのような結果になるのだろうかという疑問が起こり，著者は平成8年度からこの点について調査を始めた。調査方法としては，習得度CとDの学生について，小試験や課題における分野毎の達成度の調査，授業アンケートの活用，授業後に質問にくる学生との聞き取りなどである。その結果，表2に示す3学年までに習得すべき項目中，学生の苦手分野についてまとめることが出来た。

表2　低学年プログラミングで扱う分野

学　年	分　野
1	分岐処理・繰り返し処理 配列処理・文字列処理
2	関数処理・ポインタ 構造体・共用体・ファイル処理
3	データ構造・配列のアルゴリズム 探索処理・整列のアルゴリズム リスト処理・木構造・スタック キュー

これらを集計したものが，図2である。ただし，データについては重複しているところもある。調査期間は平成8年度～11年度前期までである。

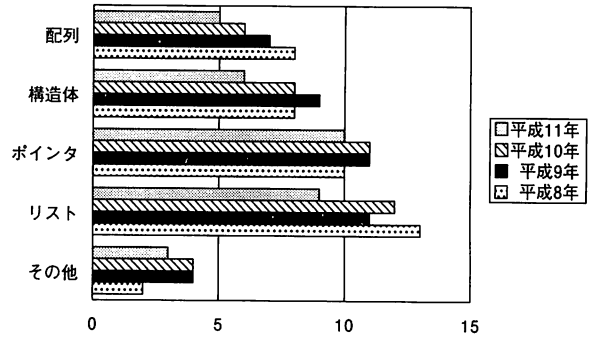

図2　苦手分野のパターン別分類（単位：人）

調査の結果，苦手分野を持つ学生を次のように「配列処理，構造体処理，ポインタ処理，リスト処理，その他」とパターン化することが出来た。

配列処理については，１次元配列における添え字の概念についての理解が不十分なため，配列操作が正常に行えず，その後の多次元配列においては更に理解不能となり，プログラミングについての苦手意識が増幅するケースであり，学生にとっては最初に訪れる１つ目の大きなハードルである。

構造体処理については，異なる型のメンバーを扱うことには違和感がないものの，メンバーを直接参照するのか，間接参照するのかなど，実際の組み立ての際での躓きが見られる。また構造体の入れ子構造，自己参照構造体においては，基本的な構造体についての理解が不充分であればかなり厳しいものがある。

アドレスを扱うポインタ処理については，配列を充分理解している学生や，１学年で行われる科目の情報基礎を理解していれば，何ら苦労することは無い。しかし，アドレスの概念を正しく理解していない学生にとっては，配列に引き続き２つ目の大きなハードルになっている。この時点で苦手意識を強める学生も多い反面，煩わしい添え字を持つ配列よりはポインタを扱う方が得意だという学生も多くなる時期でもある。

動的データ構造であるリスト処理についてはポインタの概念と構造体の理解が必須となり，この時点で足踏みをする学生も多い。しかしながら，これらは避けては通れず，他のデータ構造のプログラミングにおいては基本となる事柄である。

その他については，ファイル処理，あるいは関数処理などが上げられていたが，たとえば関数処理については，配列あるいはポインタでの引数の受け渡し方法など，他のパターンと内容を重ね合わせることが可能なものも多数あった。

4．授業方法について

著者が行っているプログラミングの授業では，通常当日行う予定の分野についての解説を行い，例題を示し，その後実習を行う形態である。この実習では，教科書の問題や，著者が自作した分野毎の演習問題をプリントまたはネットワークを利用して配布を行い（図3），各演習問題のプログラムを作成させ，その後解答例の解説を行うという形式を取っている。プログラム作成中は個々人へアドバイスを与えながらの指導となる。本来なら，問題に対するプログラムの設計を行い，それをフローチャート（PADを使用）で表してから，プログラムを作成するように指導しているが，時間の関係や分野毎の短いプログラムということもあり，理想通りにはいかないのが実情である。

J3 演習問題（７）

（自己参照構造体 No7 二分木）

{ 問題１２ }

人口と都市名を入力し、二分木構造のﾌﾟﾛｸﾞﾗﾑを次の条件で作成せよ。なお、作成にあたっては関数の再帰呼び出しの手法を用いる。

条件１ ﾌﾟﾛｸﾞﾗﾑ作成にあたっては次の型定義を使う。
構造体 toshi を定義し、これを新規データ型 CITY とする

```
struct toshi{
        long  people;
        char  city[20];
        struct toshi *left;
        struct toshi *right;
};
typedef struct toshi CITY;
```

条件２ 都市の人口と都市名をキーボードより入力する。その際、人口の入力として EOF が入力されたら、人口の多い順に人口と都市名を画面へ表示し、ﾌﾟﾛｸﾞﾗﾑを終了する。

条件３ 入力人口と現在の動的変数が示すノードのデータを比較し、入力人口の方が小さければ右分木、大きければ左分木の作成を行う。二分木を作成する時は、malloc() 関数で領域を確保しながら行う。

条件４ 関数は二分木を出力する関数と、二分木を作成する関数の二つを作る。
いずれも再帰呼び出しの手法を用いると便利である。
以下に関数のプロトタイプの例を示すが、この限りではない。

```
void treeprint(CITY *);      出力用
CITY *maketree(CITY *);      作成用
```

(j3en12.c)

図３　演習問題

演習問題は毎週新たなものが用意されるため，この限られた時間内で，演習問題を全て作成出来る学生は全体の30%程度である。そのため，前週以前の演習問題が未完成という学生も多い。これらの学生は自主的に実習室で放課後の時間帯を利用しプログラムを作成している。あるいは自宅または学生寮において，個人所有のパソコンを利用して行っているのが実情である。最近では個人で使うことが出来るUNIX互換環境としてのLinuxなどが普及しているため，学校と同様の環境で，プログラミングの予習・復習を行っている学生も多い。ちなみに，学生のPC所有率は学年が上がるに連れて年々高まっており，アンケートの結果今年度の３学年では，ほぼ全員がPCを所有していた。

一斉授業では，個々人のプログラム作成能力の差が学年進行につれて顕著に現れてくる。授業の進度，演習問題の難易度はどちらのレベルに合わせるかが，非常に難しい点である。かなり高いレベルに位置する学生にとっては，授業は魅力無いものに感じられるであろうが，プログラム作成が苦手な学生も高学年へのステップとして学年末にはある程度以上のレベルまで引き上げなくてはならず，おのずとこのよう状況での授業は進度と難

易度を落とさざるを得ない場合が出てくる。

このような状況を打破するためには，時間外の補講と学生自らの復習が大切になる。しかし時間外の補講はともかくとして，復習については苦手分野を個人で克復するには実習を伴う科目であるプログラミングに関しては簡単ではない。それには疑問点をその都度解決できる手段が必要であろう。幸いなことに著者が担当したクラスの学生は，苦手意識を持つ学生も勉学意欲は旺盛な場合が多く，著者は彼らに何らかの手助けを講じることにより，ある程度上向きな効果が得られると考えた。

次章ではその手段の検討と，補助教材開発への経緯を述べる。

5．補助教材開発の経緯

苦手分野を持つ学生は，基礎的部分で立ち遅れている為，新しい分野に入った場合明らかに新分野への習得意欲と過去からの苦手意識が交錯し，些細な点で躓くと諦めの意識が強くなる傾向がある。躓いた時点で説明を受けるとその時点では確かに理解しているが，その後の応用問題になるとまた更に不明な点が多々出現するという繰り返しでもある。これはある程度，プログラム作成に対する馴れも必要であろうが，根本的には基礎的部分の再確立が必要であろう。

これらの基礎的部分について，全て網羅した形での補助教材を作成するのが理想的であろうが，それでは教科書を全て理解するのと同様であり，あまり意味がない。著者は必要とする補助教材は欠落している分野についてのみ，瞬時に参照できるものであり，その場合卓上で参考書を開くというのではなく，必要な時に瞬時にその分野の項目を検索でき，応用できるものであると考えた。これには簡単なデータベース的なアプリケーションが望ましいと思われる。

現在，低学年に対するプログラミングの授業ではタイピング練習や基礎的なOS環境などの実習は別として，前述したがC言語関係では大まかに表2の様な分野について実施している。

これらの分野から3章での分析・パターン化に基づき必要と思われる項目について自学習可能な補助教材を作成することにした。

6．補助教材の概要

補助教材用のアプリケーションを開発するにあたり，次の点を特に留意して設計を行った。

(1) 希望する分野の項目が瞬時に検索可能であり実用的であること。
(2) 自宅等でも利用可能であり自学習可能なものであること。
(3) 扱う内容は授業で行っている内容を中心に，作成を行うこと
(4) 留学生の補講にも使用可能なように対応させること
(5) 将来的に機能拡張が容易であること

また，開発環境であるが開発はDOS/Vマシンで行い，主要なプログラムはMicrosoft社のVisual C++6.0を用いMFCを用いたWindowsアプリケーション開発の手法に基づき行った。また一部基本データについてはMicrosoft社のAccess97を用いて作成している。

全体の構成であるが，項目として配列処理，リスト処理，構造体処理，ポインタ処理について扱うことにした。更に，授業で行っている演習問題と演習問題の解答例も追加して考えることにした。ただし，解答例については授業で示したものについてのみであり，他はヒント程度に留めた。

図4はUML（Unified Modeling Language）の記法によるアプリケーションの検索処理におけるユースケースを表すアクティビティ図である。

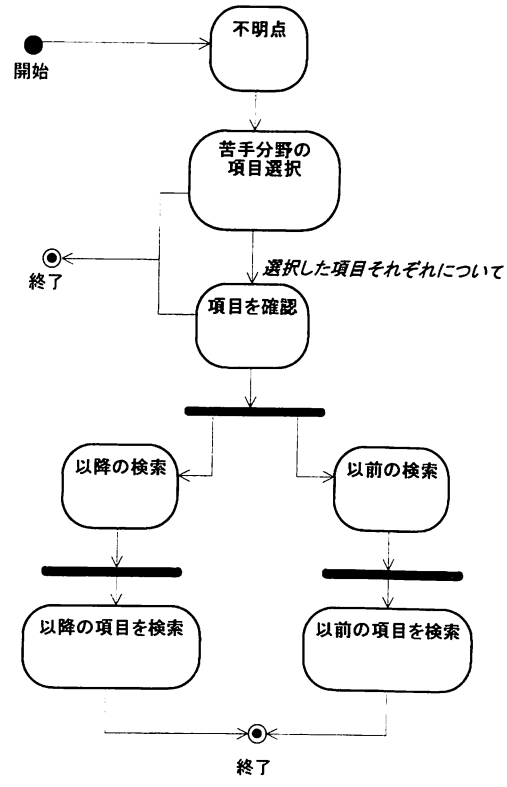

図4　UML記法によるアクティビティ図

図4の構成は次のようになる。利用する学生がプログラム作成中あるいは自学習中に不明点が出現するというアクションが発生した場合，学生がアプリケーションにより苦手分野の項目パターンを選択するというアクションを取る。ここでは作成時の留意点(1)の瞬時に検索可能であるという事を満足しなければならない。またここでの項目パターンは苦手分野＋演習問題関係の6項目にする。次にその項目についての詳細パターンによりその項目に関する情報について希望するものを選択し確認するというアクションに移る。ここでは留意点(5)の機能拡張が容易であるように，新たな項目も容易に追加できるようにする必要がある。さらにそれに関連してその項目以前，または以後の項目も同時に検索し解説等を閲覧することが可能なように，それらのアクションを選択可能にする。これらは，留意点(3)の授業で行っている内容を中心に関連付ける必要がある。その後はそれぞれの項目についての検索等のアクションに移行する。ここでのアクションは留意点(4)の留学生にも対応可能なように，展開しなければいけない。すなわち，日本語に不慣れな留学生に対しても使い勝手の良いアプリケーションでなければいけない。留学生に関しては著者のクラスにも1名在籍しており，学科全体では3名の留学生がいる。留学生の場合は1学年からプログラミングの授業を受けている日本人学生と比べ，国柄の違いもあり情報関係科目に対する教育内容が異なるため，3学年へ編入した時点では情報基礎という特別科目を設けプログラミングを含めた情報関係科目の個別指導を行っている。留学生が3学年後期から始まるソフトウェア実験への手助けとなるような補助教材が望ましい。

図5は実際にアプリケーションを起動した状態である。著者はデータベース的なイメージで使用できることが望ましいと考え，このようなレイアウトにしてある。ここでは6つの検索パターンについては，タブ付きの特殊なダイアログであるプロパティシートを用いた。これらはMFC (Microsoft Foundation Class) のプロパティシート (CPropertySheet) クラスとプロパティページ (CPropertyPage) クラスを使い構築している。

これにより，苦手分野の検索へのアクションはプロパティシートの項目を選択するだけで済む。図5は構造体処理についてのプロパティシートを選択した場合である。

また，ウインドウ内では項目に対する解説と，書式，講義日，実行例などを想定している。講義日については，これは著者のメモ的なものではあるが，学生にとっても学習した時期のノートを読み返すのに役立つはず(?)である。

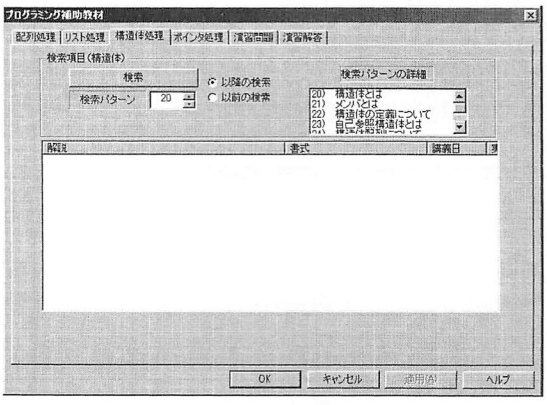

図5　起動直後の例

図6，図7に配列処理とリスト処理の実際の検索画面を示す。図6の一次元配列については，検索パターンの詳細についてリストボックスより選択し検索パターンを入力する。その際，多次元配列とそれ以降の関係する項目を閲覧するため，ラジオボタンにより項目5以降の閲覧を選択している。その結果，検索内容として多次元配列の他，宣言方法，配列関係の使用例などが閲覧可能になっている。

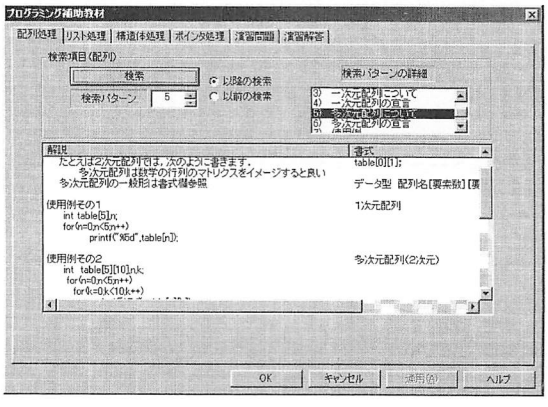

図6　配列処理の例1

留学生のための配慮としては，未だ不充分ではあるが図8のように日本語表記を簡単にし，できる限り専門用語もやさしく解説した。理想としては英語バージョンの開発が考えられるが，現段階では着手していない。

演習問題と演習問題の解答例については，このプロパティシートでの表示の他，Visual C++ 6.0においてInternet Explorerのユーザインタ

フェイスが活用できるようになっており，標準のHTMLやVBScriptやJscriptなどのスクリプト言語も使用可能となっている。そこで授業用に作成しておいた演習問題および解答例をそのままHTMLに変換し，図9に示すように表示することも可能である。これらについては，すでに従来の演習問題をHTML形式に変換したものがあり，則時対応可能である。

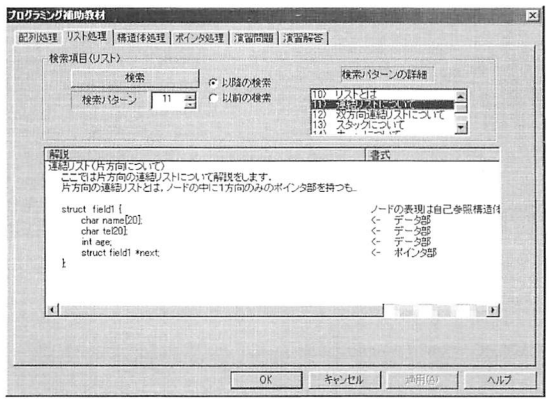

図7　リスト処理の例

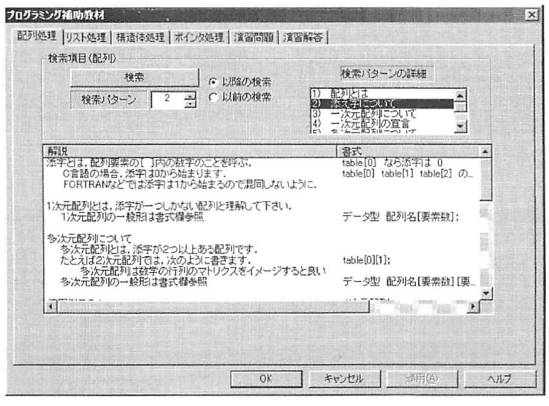

図8　配列処理の例2

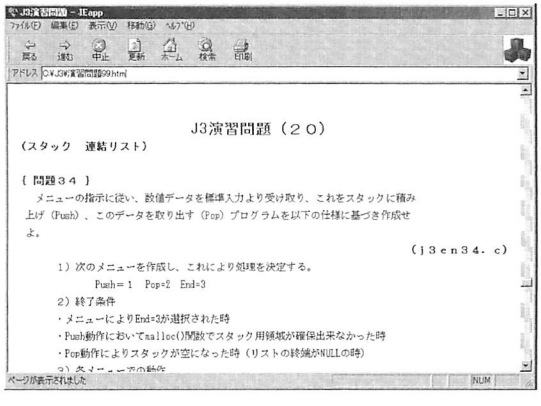

図9　演習問題の表示

7．おわりに

本研究では著者が担当する，低学年プログラミング授業においてプログラミング技術の習得度が低い学生についての要因分析を行った。その結果としてある段階から顕著に苦手意識が現れることが判明した。それらを少しでも克復させるために，プログラミング授業における補助教材の開発を行ってきた。現段階は試行期間で実際の利用統計等は取っていない。今後，授業あるいは授業外で効果を確認し，その結果改良する必要性が出てくるであろう。また次のようなことが当面の課題として考えられる。パターンの新たな項目の追加，詳細パターンの充実，演習問題の充実，留学生への配慮としての，英語バージョンの対応，C言語に偏らず，あらゆる言語についても対応可能にする，などである。また多言語への対応では，特に本研究でも使用したオブジェクト指向言語に対する教材の作成などが上げられる。

本研究で開発した補助教材は，プログラミングに対して苦手意識を持った学生・あるいは留学生の補講等に大いに役立つと考えられる。

参考文献

1）桜田幸嗣・田口景介：Visual　C++6.0プログラミング入門，アスキー出版局，1999
2）H・M・ダイテル・P・J・ダイテル：C言語プログラミング，プレンティスホール，1998
3）Martin Fowler・Kendall Scott：UMLモデリングのエッセンス，アジソン・ウェスレイ・パブリッシャーズ・ジャパン，1998

（平成11年11月29日受理）