

情報技術科C++言語プログラミングに関する教材の見直し

情報技術科 久保 雅俊

1 はじめに

従前の情報技術科のカリキュラムにおいて、プログラミング言語教育は1年次3Qから2年次2QまでC++言語とJava言語を学生の希望による選択制としていた。しかしながら、平成25年度より当時ソフトウェア開発系企業の多くが開発にJava言語を採用していたことから、1年次3Qに全員がJava言語を学習し、4Qから選択制とするように変更した。その際に、1年次の4Qの授業では、3Q4Qを通じて利用していた教材から抜粋し、基礎的な部分を学び、後半の応用的な部分は進捗状況により学生が個々に進めていく形とした。

ところが、年々の学生の学力低下に伴い、1年次に予定している授業内容の理解が及ばず2年次の授業にも影響が出始め1年次の復習に1か月を要するようになってしまった為、1年次の4Qから2年次の2Qまでの教材をすべて見直し、学生がしっかり理解できるように改定が必要となった。

平成24年度以前			
1年次		2年次	
3Q	4Q	1Q	2Q
C++言語	→	→	→
Java言語	→	→	→

平成25年度以降			
1年次		2年次	
3Q	4Q	1Q	2Q
Java言語	C++言語 →	→	→
	Java言語 →	→	→

図1 情報技術科における言語系選択授業

2 研究の目的

1年次4Qに開講するソフトウェア実習(C++)、2年次1Q2Qに開講する図形処理I(C++)、図形処理II(C++)、図形処理実習(C++)の学習教材を全面改定する。学生が理解しやすくなるよう説明の提示順序や内容を変更し、また学生の理解が及ばない部分に対する補助解説や練習問題を強化した。

更に、PCの入れ換えに伴いソフトウェアの開発環境が変わる為、それに合わせた解説も加えた。これは開発用アプリケーションの操作方法の違いといった簡単なものではなく、開発するソフトウェアのコードの変更にまで言及する必要があった。

具体的には、従前C++98を基準とした文法解説からC++11、C++17の中で必要な機能の追加、またWindowsアプリケーションについてMicrosoft Visual C++が標準サポートしなくなった点を考慮した開発の進め方などの解説などが挙げられる。

3 課題の提示方法の検討

授業内での学生の学力の差は必ずあり、従来その差は問題数で吸収してきた。必須問題とオプション問題にわけ、オプション問題は必須問題が終わった学生が取り組むものと位置付けた。しかしながら、必須問題が終わったにもかかわらず「必須ではないから」と手を付けない学習者がいたり、あるいはその逆で必須問題に手間取るにもかかわらずすべてを解こうとする学習者がいたりして、こちらの思うように授業が進行しないこともあった。

そこで、今回はすべてを必須問題とし、ヒントを段階に分けて提示し、そのヒントの中で小問題を解かせるようにした。最初、ヒントはすべて閉じた状態であり、学習者が必要になったらボタンを押して閲覧するという形にした。

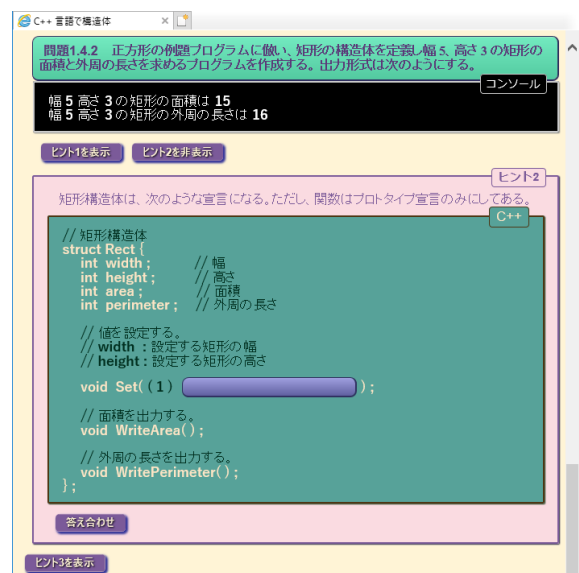


図2 ヒントの提示（ヒント2のみ表示した状態）

4 内容の検討

今回の教材作成にあたり、次のような点に注力した。

4.1 文法の解説範囲の見直し

元々文法の解説は C 言語の学習後を前提とした流れとなっており、それを踏襲し新たに次のような点に手を入れた。

4.1.1 オブジェクト指向の概念

従来もオブジェクト指向の概念を解説しクラスの導入をしてきたが、C 言語の関数型に慣れてしまった学習者には一番大きな壁となっている。今回、C 言語のグローバル変数を用いてそれをメンバー変数と見立ててクラスへ導くように解説を変更した。

4.1.2 参照型の導入

参照型の概念は初期の C++言語からあったが、それは friend 修飾子と併せて演算子のオーバーロードの為に作られた機能と解釈しており、逆に演算子のオーバーロードをしなければ必要ないと考えてきた。

しかしながら、C++11 から右辺値参照が導入され、ムーブと言う概念は実行速度やメモリー効率と言った点で習得が必要と判断した為、遅ればせながら解説することにした。

4.1.3 ラムダ式の導入

参照型と同様になくてもプログラムは作成できるが、使うと便利なラムダ式も解説することにした。ラムダ式は諸刃の剣で省略し過ぎるが故に逆にわからなくなるといったことになり兼ねないので、段階的に省略して最終的にラムダ式にするように解説した。この方法は C#言語の教材において既に試しているが、学習者の理解は今一つであったことを踏まえ、演習問題で理解を補うことにした。

4.2 Windows アプリケーション

Visual Studio 2012 からストアアプリ開発に力が入れられた為、C++言語のプロジェクトに「Windows アプリケーション」がなくなった。また、ユーザーコントロールは別に DLL を作成しそこに登録しなければならないなど色々な制約がある為、それらを解説した。

4.2.1 イベントドリブン

今まで順次制御に代表されるフローチャートで記述できるような処理しか記述してこなかった学習者がイベントドリブンで躓くときは、イベントが発生するボタンなどのリソースではなくイベントで処理するテキストボックスなどのリソースに処理内容を記述し、動かないと嘆く状況が多い。

また、イベントが発生して処理したことで別のイベントが発生するといった連携の仕方が見える学習者があまり多くないので、それらが理解できるような問題を出題した。

5 教材作成に係る補助機能

最近の教材開発はすべて HTML ファイルに記述している。また、学生が記憶媒体にコピーして持ち帰り、自

宅でも学習できるようなファイル構成になっている。これは Web サーバーを使用しないことが前提であることを意味し、クライアントサイドだけで色々な操作ができるようにしなければならない。それは作成者に色々な負担を強いることになるので、次のような工夫を行い、少しでも軽減されるようにした。

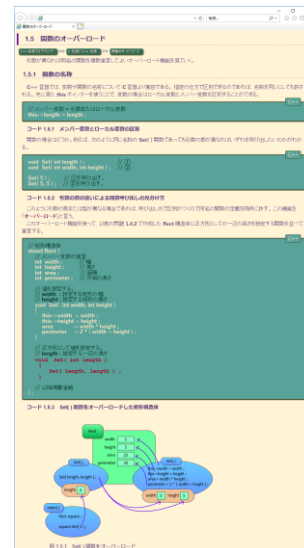


図 3 Web 教材全景

5.1 XML 化

従来の HTML から XML 化することで XSLT の設定により、次のような効果や機能追加ができた。

- ・タグの使用制限によるソース可読性の向上
- ・章番号、問題番号等の割振りを自動化
- ・問題の別解管理を自動化

5.2 JavaScript の補助

従来から JavaScript を使って次のような機能を持たせているが、機能追加及びリファクタリングを行い JavaScript 内のコードの可読性を向上させた。

- ・図の拡大縮小表示
- ・パンくずリストの自動作成
- ・問題の正解判定 (判定方法変更)
- ・全ソースコードの表示非表示 (追加)
- ・ヒントの表示非表示 (追加)

6 おわりに

今年度はソフトウェア実習 (C++) の教材作成のみに留まってしまったので、来年度引き続き作成及び検証し改訂していきたい。

7 参考文献

- (1) 久保雅俊, [実験装置における実験メニューの表現と実現方法], 産業技術短期大学校平成 28 年度講師研究発表会プログラム, 67, (2017), 7.