

プログラミング演習授業支援システムの開発

松永 豊

情報教育講座

The Development of the Programming Practice Class Support System

Yutaka MATSUNAGA

Department of Information Sciences, Aichi University of Education, Kariya 448-8542, Japan

1. はじめに

本学, 情報教育課程の卒業研究ではプログラミングを必要とする研究が数多く行われている。プログラミングそのものの研究はもちろんのこと, システム開発系の研究を行う学生も多い。

周知のとおり, プログラミング言語は数多く存在し, 与えられた課題や作りたいシステムなどによって得意不得意が存在するため, 研究に際しては適切な言語を選択する必要がある。すなわち, 研究室配属以前に習得しておくことが望ましいプログラミング言語が複数存在すること意味する。

ところで, プログラミング言語の習得はそれほど簡単なものではない。前述のとおり, プログラミング言語は多数存在するので, まず, 言語選びから始めないといけませんが, 一般的に習得が容易なものは応用範囲が狭い場合が多く, 逆に汎用性・応用性が高いものは初心者には難解な場合が多い。

また, 一つの言語を集中的に学ぶべきかどうかとも議論が分かれるところである。『複数の言語を体験しておいたほうがよい』『アルゴリズムさえ理解できれば言語の違いはさほど重要ではない』『ユーザが多い言語が必ずしも初心者に向けた言語とは限らない』など, さまざまな考え方が存在するからである。

さらに, 指導の際にも注意が必要である。プログラミングは数学と国語が融合したような部分があるため(後述), 個々の細かいケアが重要だからである。そのためには事実上の個別指導が必要となるが, 関連してさまざまなデータ収集も必要となり, すべてを手作業で行うと大変である。

そこで, 本研究ではプログラミング演習授業支援システムの開発を行った。また, 実際の授業で本システムの有効性を検証した。

2. 情報教育課程におけるプログラミング授業

開発したシステムの説明をする前に, 本学, 情報教育課程におけるプログラミング授業について説明しておく。前述のポイントを踏まえ, 情報教育課程では以下のようなプログラミング関係の授業を展開している。

表1 プログラミング演習授業

授業名	使用言語
プログラミング1	PEN / C++
プログラミング2	C++
プログラミング3	C++
プログラミング方法論	C#

情報科学コースにおいては, このうち, プログラミング1~3が必修でプログラミング方法論は選択科目である。(ちなみに, 中等情報コースはプログラミング1のみ必修で残りは選択。)

まず, プログラミング1では入門編として主にPEN^[1]を利用し, 後半にC++も扱っている。PENは日本語プログラミング言語の1つで, 入門用としては優れている。ただし, 開発されてからそれほど時間がたっていないこともあり, 普及率はそれほど高くなく, ユーザ数が少ない(つまり, 書籍や解説WEBページ等が少ないことを意味する)。PENを用いた演習においては, 変数, 制御構造などの基本事項と配列の一部は学ぶため, それなりのプログラミングは可能となるが, 応用的なプログラムを作ることは難しい。どちらかといえば、『プログラミングとは何か』『コンピュータに計算させることとは』のような, プログラミングの本質的な理解やアルゴリズムについての理解を中心とした内容となっている。ただし, 中等情報コースの学生にとってはプログラミング1だけが必修科目であり, 選択科目であるプログラミング2以降を受講しない可能性もあるため, マイナーな言語であるPENだけ

では不十分であると考え、後半では最もメジャーな言語の一つである C++ を用いた演習も扱っている。このとき、PEN で行った演習課題を C++ で作らせている（いわゆる移植作業）。PEN と C++ を比較させることで、言語が違っててもアルゴリズムはほぼそのまま利用できることを理解させている。また、プログラミング 2 への引継ぎをスムーズに行わせる狙いもある。

プログラミング 2, 3 では使用言語として引き続き C++ を用いており、より応用性の高いプログラミングを行っている。扱う内容も、変数、制御構造など基本事項はもとより、関数、配列、構造体、クラス、ファイルなど、一通りの文法事項、インデント、コメント、仕様書やチャートの書き方など、作法的なこと、再帰、リスト、オブジェクト指向、アルゴリズムや概念的なこと、など多岐に渡る。

プログラミング方法論ではオブジェクト指向を中心としたビジュアルプログラミングを扱っている。具体的には Visual Studio Express Edition を用いたウィンドウソフトの開発を行っている。使用言語は C# である。Visual Studio には C++ 版も存在するが、以下の 3 点より C# 版を用いて授業を行っている。

- ・ Visual C++, Visual C# で同じプログラムを作成した場合、C++ 版の方ではポイントが多用され煩雑な表現のプログラムになり、その説明に多くの時間を割く必要が生ずるため、本来の目的であるビジュアルプログラミングの本質がぼけてしまう点。
- ・ C++ と C# は言語体系が極めて似ており、多くの文法がそのまま利用できる点。（少なくとも PEN から C++ への移行よりはギャップが少ない）
- ・ 複数の言語を体験しておくこと自体は決して悪いことではない点。

以上が大まかな授業内容であるが、プログラミング 1, 2, 3, 方法論に関しては、実際の授業運営に関してもかなり特徴のある方法を採用している。

まず、各授業の最後に行う個人面談が極めて特徴的である。各授業においては、最終課題としてその単元に見合った課題を作成してもらっている。課題内容は初期段階（具体的にはプログラミング 1 や 2）では教員側が用意した問題を解いてもらっているが、最終的には受講者本人にテーマを選んでもらい作成させている。そのため、受講者全員がバラバラの内容になり、対応も学生ごとにバラバラになる。その際、進度をより正確に把握するために個別面談を行っている。個別面談では教員 1 人、受講生 1 人で面談し、完成したプログラムの説明をしてもらったり、個別のアドバイスをしたりしている。

次に特徴的なのが、リカバリ学生のケアである。プ

ログラミング授業は積み上げ科目として位置づけられており、同時受講も禁止にしている。すなわち、プログラミング 1 が合格しないとプログラミング 2 には進めない。すべての学生がすんなりと上位コースに進んでくれればうれしいのだが、当然、数名は足踏みをすることになる。すべてのプログラミング授業は半期科目であり、当然、連続的に開講されているため、通常の科目のように、前期に落とした授業を次年度の前期に受講しては、半年間ブランクが空くことになってしまう。そこで、プログラミング授業においては、本来の開講時期とは逆側（前期なら後期、後期なら前期）にリカバリ用の授業を用意している。

このように特徴的な授業運営をしているため、教員側の人的リソースの管理も重要となっている。そこで、情報教育講座ではプログラミング担当チームを結成し、これらの授業を有機的に運用している。

なお、情報教育課程においては、この他にもプログラミングを教えている（プログラミング演習を行っている）授業が数多く存在する。無論、プログラミング担当チームが直接管理しているわけではないが、説明中に登場したプログラミング言語以外にもさまざまな言語を扱っている。もちろん、研究室単位で行うゼミの中でも数多くのプログラミング勉強会が存在し、扱っているプログラミング言語も多岐に渡る。その一部を紹介すると、PHP, JAVA, LISP, C, ActionScript などがあげられる。

3. プログラミング演習授業で求められるもの

前述のとおり、プログラミング授業は他と授業とは異なり特殊な授業運営を行っている。また、演習という授業の性格上、多かれ少なかれ、個別指導が重要な授業と考えられる。

周知の通り、プログラミングにおいてはあいまいな表現は一切許されない。人間同士が外国人と会話する場合のように、『文法は間違っているが、多分、こういう意味だろう』と察してくれるなどということは絶対にないし、『前後の文脈から言えば、この計算式は間違っているのではないか？』などという配慮も絶対にない。与えられた命令を忠実に実行するだけであって、（計算式などの）意味が正しいか間違っているかは作成した人間にとっては極めて重要な関心事であるがコンピュータにはまったく関係ない。そのような意味では、プログラミングは論理的、数学的な要素を強く含んでいるといえる。

一方で、プログラミングにおける解答が 1 つではないことも周知の通りである。そもそも、アルゴリズムの段階ですら 1 つとは限らない。そのため、数名のプログラマが単独で同じ課題を解いた場合、完成するプログラムがまったく同じものなることはまずない。そういう意味ではプログラミングは国語的な要素を強く

含んでおり、『プログラムとは一種の小論文のようなものである』という考え方すら存在する。事実、プログラムが得意の人が見ると『A君のプログラムは明らかにB君の真似だな』と分かる場合がある。これは、たとえば以下のような比喩に匹敵するかもしれない。

- ・日本語を学びたての留学生A君が、日本人B君の書いた論文を一部だけ修正してそのまま提出したら写したことがばれてしまった。

とりわけ、プログラムにおいては変数名や関数名、表示させるメッセージなどを単純に書き換えただけで、一見違ったように感じる。だが、素人（学びたての人）には充分書き換えたつもりでも、プロ（既習者）にはバレバレの場合があるということである。これは、変な日本語の、でもどう考えても同じ内容の論文が提出された場合、素人（＝学びたての留学生）には見抜けてなくても、プロ（＝日本人）には一目瞭然であることと似ている。ここで重要なことは、勉強中の学生がそのことにすら気づいていないという事実である。つまり、レポート未提出の学生が要注意なのは当然として、逆に、レポートを提出しているから問題ということもありえるという意味である。場合によっては、『こんな写したことがバレバレのレポートを出すわけにはいかない』と提出を躊躇した学生よりも事態が深刻かもしれない。

では、そのような学生をいち早く見抜き正しく誘導するにはどうしたらよいだろうか。その答えの一つが個別面談である。個別面談でいくつかの質問をするとたいいてはすぐに分かる。このとき、難解な質問をする必要はない。たとえば、『この変数はなぜ必要なのか』『この変数はいかなる理由でこのような変数名を付けたのか』『この部分では何をやっているのか』などで充分である。作者本人であれば即答するが、中には（作ったのは提出した本人であるはずなのに）、『この部分で何をやっているのか分かりません』などの答えをする場合がある。これは英文のレポートを提出させて提出者に日本語ではどういうことを書いたつもりなのか聞いたら『訳せない』と答えているようなものである。

もう一つは、本人、他人問わず、提出されたレポートをすべて保存しておき、必要に応じて見比べることである。他人のプログラムと比べてそっくりなものが出てくれば要注意であることは明白だが、中には、「他大学の学生に作ってもらった」「インターネットからそのまま拾ってきた」などという事例もある。重要なのはインターネットを使って調べたことが問題なのではなく（むしろ、自学習として推奨すべきこと）、他人のプログラムを流用した場合、まったく意味を理解せずに使っている場合があることである。この場合、学生

間の比較では見つけられないが、先ほども数名のプログラマに同じプログラムを作らせてもバラバラになると述べたように、一般的に書き方の癖のようなものが含まれるため、本人の他のプログラムと比較すると分かる場合がある。かなり昔の話になるが、『なぜ、私のプログラムがよそからの流用と気づいたのですか？』との学生からの質問に対し、『いままでゴッホ調だったプログラムがあるとき突然ピカソ調に変われば気づくよ』と指導している先生がいて、『なるほど、うまいこというなあ』と感心したものである。いずれにせよ、早めに気づいてあげることが本人のためにもなると考えている。

4. プログラミング演習授業支援システム

以上のことを踏まえ、本研究では以下のような機能を意識したシステムをPHP^[2]を用いて開発した。

- ・授業内容のアナウンス
- ・自動出欠記録
- ・レポート提出システム
- ・レポートコメント機能
- ・提出レポート保管庫
- ・面接予約システム
- ・電子カルテ
- ・プログラミング授業間の連動
- ・教員間スケジュールの連動

このうち、面接予約システムに関しては、以前、詳しく報告した^[3]。これはPukiWiki^[4]をベースとしたシステムであり、本システムも同様のノウハウを利用して構築した。なお、面接予約システムは本システムの一部として組み込まれているが、さらに一部が改良されている。

より、具体的に意識した点について列挙すると大体以下ようになる。

- ★ プログラミング1, 2, 3, 方法論においては、扱うプログラミング言語がPEN, C++, C#と変わるが、共通に使えるシステムを目指した。
- ★ プログラミング授業では複数の先生が担当するが、本システムを共同で使用している限りにおいてはスケジュール等に整合性を持たせるようにした。たとえば、面接予約においては、1度に複数の予約ができないルールになっている。具体的には初回面接と2回目の面接で面接教員が変わる場合、1回目と2回目の両方に予約できてしまうと不具合が生じる。なぜなら、1回目の面接時に2回目までの課題が与えられるので、2回目分としてはプログラムが未完成のは

ずだからである（プログラムが完成していない時点で予約することはルール違反になる）。そこで、面接希望教員を変えた場合、自動的に前の予約はキャンセルになるようにした。なお、（同じ教員間での）面接希望日時の変更に關しては、以前の面接予約システムの時点で既に実現している。

- ★ 複数の先生で使用することを考慮し、各ページの読み書きの権限として、管理者権限、スタッフ権限、学生権限の3種類を用意した。
- ★ 授業で提供しているさまざまなサービスをなるべく一元化する工夫をした。具体的には、通常課題の提出や面接予約時に必要の最終課題の提出先をまとめた。ただし、最終課題に関しては、カルテや面接実施記録などと連動させるため、通常課題とは異なるページで管理した。
- ★ 通常課題のコメントシステムを作成し、レポートのペーパーレス化を促した(図1)。通常課題においては、プログラムソース、実行結果、(その他、場合によっては仕様書など)を印刷させ、表紙とともに通常の(紙の)レポートとして提出させていたが、これはプログラムをチェックして赤を入れて返すためである。デジタルデータでの提出が便利であることは分かっていたのだが、ファイルの形でソースファイルをもらってもチェックを入れることが困難だったため、紙でも提出してもらっていた。そこで、提出プログラムに対してWEB上から簡単にコメントを書き込めるシステムを構築した。この結果、レポートのペーパーレス化が実現できただけでなく、比較的良好に起こるミスを全員にアナウンスできるというメリットも発生した。

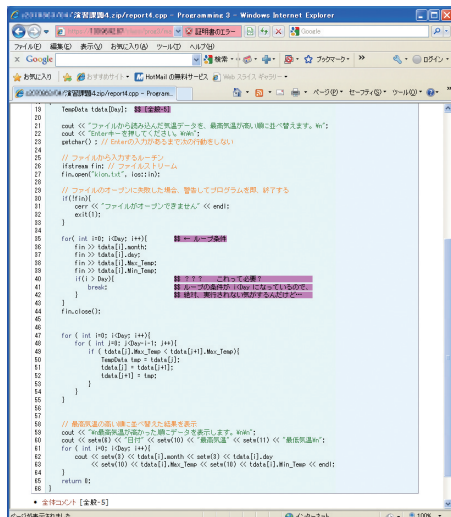


図1 コメントシステム（学生用）

- ★ コメントシステム構築に伴って、教員が書き込みしたデータと提出オリジナルデータの両方が必要になったため、ファイル提出サーバへ提出されたファイルを教員がコメントページへ転送する形にした。これにより、学生による自由な書き換えと提出ファイルの保護の両方を実現した。（詳細は後述）

- ★ 面接予約に関する機能を大幅に向上させた。具体的には、面接予約を行う前に完成品を提出する必要があるが、完成品を提出して初回面接を予約した時点でカルテ等のページが自動的に生成され、スケジュールページから提出物やカルテ等が自動的にリンクされるようにした(図2)。また、同時に実際の面接時の出欠確認もクリック一つでできるようにした。(図3)

図2 スケジュール確認（教員用）

図3 個人データの確認（教員用）

- ★ 面接予約のスケジュール管理に関しては基本的には以前作成した面接予約システムの方法を踏襲したが、複数の教員が使用することを考慮してスケジュール作成の程度の幅を持たせた。具体的にはPukiWikiのテーブル機能を用いてスケジュールを作ることになるが、一定のルールさえ守れば、開始時間、終了時間、表示方法などかなり自由になるようにした。（ただし、面接時間は30分刻みとし、変更はできない）

以上の点を踏まえ、学生と教員による具体的な操作手順は以下の通りである。

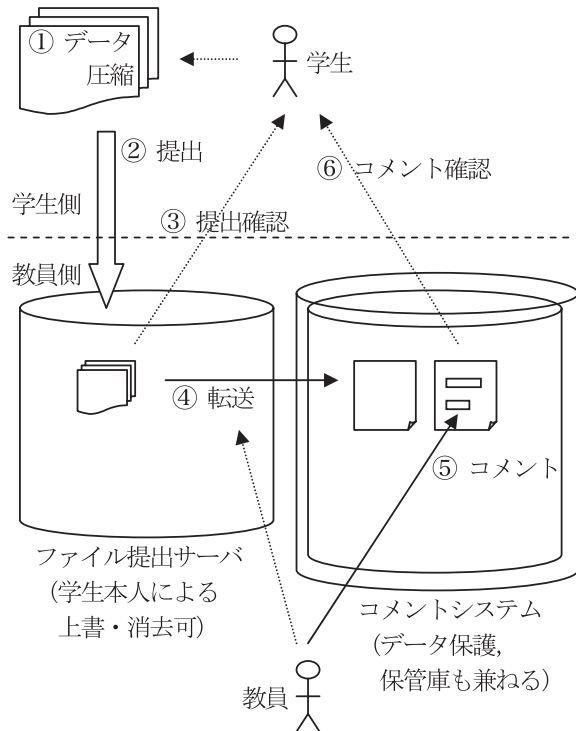


図4 レポート提出とコメント

- ① 学生：提出すべきファイルを zip 圧縮する。このとき、1つの課題において提出すべきファイルが複数ある場合は1つのフォルダにまとめてフォルダ圧縮する。なお、それぞれのファイル名は提出すべきファイルの種類に応じて一定のルールを持たせてある。たとえば、実行結果は *result*.txt (*はワイルドカード) という名前のテキストファイルにするなど。これにより提出漏れの自動チェック（後述）が可能となる。
- ② 学生：ファイル提出システムにより zip ファイルを提出する。
- ③ 学生：提出したファイルを確認する。このとき、課題ごとに提出すべきファイルが異なるが（プログラムソースのみで OK な課題、仕様書＋プログラムソース＋実行結果が必要な課題など）、ファイル提出サーバ自体に zip ファイル内に含まれるファイルを自動的に判別する機能を持たせたため、提出不足や余計なファイルを提出していないかなどがすぐに分かるようになっている（図5）。ファイル提出サーバ上のデータは学生本人ならば上書や消去が可能なので、提出不備が確認できた場合は、必要に応じて再提出可能である。なお、教員が転送（後述）を行った後には上書や消去不能となるため、別名で提出する必要がある。

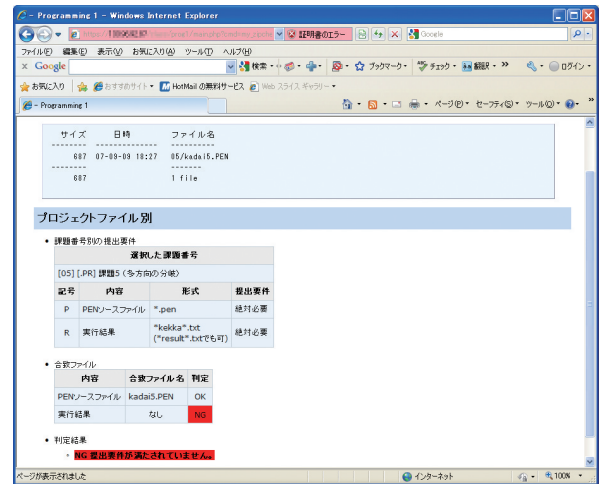


図5 提出物自動判定（学生用）

- ④ 教員：ファイル提出サーバに提出されたファイルをコメントシステムに転送する（基本的にはクリックすればOK）。この時点でファイルは保護され、その後、学生が上書や消去できなくなる。
- ⑤ 教員：コメントシステムに転送されたデータは教員がWEB上から自由にコメントできるようになるため、必要に応じてコメントする。なお、書き込んだコメントが閲覧可能なのは提出者本人だけであるが、必要に応じて「全体的なコメント」というページにコメントを書いてリンクをすることも可能であり、その場合、受講生全員が閲覧可能となる。そのため、よくあるミスに対するコメントなど、何度も同じ内容を書かずに済む。
- ⑥ 学生：学生は教員からのコメントを見ることができ、必要に応じて再提出するなどする。

5. 本システムの運用

本システムを用いて筆者が担当する「プログラミング1」「プログラミング3」で実際運用してみた。ただし、運用時にいくつか改良した部分もある。その結果は以下の通りである。

- ・ペーパーレス化は概ね学生に好評。
- ・授業中、コメントに対する質問が増えたので、コメントシステムが有効に機能していた。
- ・プログラミング1とプログラミング3に対して、まとめてアナウンスしたい内容（両受講生に関係すること）と取って分けてアナウンスしたほうがよい内容の両方があり、その点においてもう少し改良が必要。（特に3の内容を1で公開すると混乱しそうだったので、その部分は分けて作った）
- ・教員間の連携は非常にうまくいった。特に、プログラミング1では必ず複数の先生に面接してもらう必要があるため、電子カルテでの情報共有が有効であった。

- ・スケジュール作成に関してかなり自由度を設けたため、教員間でこだわり（出張、休日出勤、時間外の設定など）があっても対応可能だった。
- ・提出レポートが自動的に整理されて保管されるのは便利。

6. ま と め

本研究では、プログラミング演習授業支援システムの開発を行った。作成したシステムにおいては、複数のプログラミング関連授業で連動して使えるように工夫した。さらに複数の教員が同時に利用可能となっていることが特徴である。実際の授業で運用してみたところ、面接や指導の際に有効であることが分かった。

しかしながら、いくつか改良すべき点も見つかった。一つはインターフェイスの向上である。本システムを利用して授業を行った教員からは、『一通りコメント等は可能なのだが、コメントが書きづらい』『複数のページを同時に閲覧しながら面接を行う際に使いにくい』などの意見をもらった。

もう一つは、プログラミング授業間での連携に関する改良である。運用のところでも述べたとおり、別々の授業に対してまとめてアナウンスしたい内容とバラバラにアナウンスしたい内容があるため、完全に融合してしまうと返って使いづらくなる。逆に、もっと連携度を高めたほうがよい点も見つかった。具体的には、1人の学生をプログラミング1から方法論まで追跡するような仕組みがあるとより利用しやすくなり、学生指導にも有効になると考えられる。これらの改良が今後の課題である。

参考文献

- [1] PEN: <http://www.media.osaka-cu.ac.jp/PEN/>
- [2] PHP: <http://www.php.gr.jp>
- [3] プログラミング演習授業のための面接予約システムの開発, 松永 豊, 愛教大研究報告58輯(教育科学編), 2008. 3
- [4] PukiWiki: <http://pukiwiki.sourceforge.jp>

(2009年9月17日受理)