

# 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

情報教育講座

## Development of the Interview Reservation System for Programming Practice Classes

Yutaka MATSUNAGA

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

### 1. はじめに

本学, 情報教育課程では, いくつかのプログラミング演習授業を行なっている。授業で扱う言語は主に C++ であるが, 近年では入門編として PEN [ 1 ] を利用しており, また, 応用編として JAVA や PHP [ 2 ] を扱う場合もある。扱う内容も, 文法, 作法, アルゴリズム, 概念, 応用など, 指導範囲も広く充実している。

このように指導範囲が広く, また, プログラミング演習という授業の性格上, 当然, 学生間の習得状況に違いが起きてくる。そこで, 本課程では学生へ個別面談という極めて特徴的な教育方法で学生間の習得状態のバラつきを吸収している。

表 1. プログラミング演習授業で扱う主な内容

項目	扱う主な内容
文法	変数 制御構造 関数 配列 構造体 クラス ファイル など
作法等	インデント コメント 仕様書 チャート など
アルゴリズム 概念等	再帰 リスト オブジェクト指向 など ビジュアルプログラミング
応用	WEB プログラミング など

### 2. 個別面談

プログラミング授業は, 入門, 1, 2 のように, 積み上げ授業として構築されている。当然, 内容はそれぞれの授業で大きく異なるが, 授業の流れはさほど大きく違わない。非常に大雑把には, 以下のようなスケジュールとなる。

1. 半期 (15回) 授業の前半で新規項目等を学ぶ。
2. 後半では各自が最終課題を作成する。
3. 最終課題について個別面談を行い, 面談の結果, 評価を決定する。

このように個別面談 (本課程では面接と呼んでいる) を行なうことにより, 木目細やかな指導が可能になることは言うまでもないが, 個別面談という行為は, 学生, 教員ともに大きな負担を伴うことも事実である。特に双方の都合のよい時間帯を予約することが大変である。以前は, 部屋の前に予定表を貼り付けておき, 学生に書き込みしてもらっていた。ただ, 学生にとってみれば予約を取るために余分に足を運ばせなければならぬため, 特に夏季休暇や冬期休暇にずれ込んだ場合や遠隔地に住む学生には負担を強いていた。また, メール等で予約させるという方法もあるが, ほぼ同時期にメールが届き, なおかつ, 希望する時間帯がバッティングした場合など, その調整が極めて煩雑になることも多い。場合によっては複数回メールを書く必要が生ずるなど, あまり能率がよい方法とはいえない。

そこで, 本研究では WEB 上でスケジュール確認および予約が可能なプログラミング演習授業用面接予約システムの開発を行なった。

### 3. 各授業に特化した項目

前述したとおり, 基本的にはすべてのプログラミン

グ関連授業で面接を行なっているため、単純な仕組みであっても面接予約システムの利用価値は高い。しかしながら、個々の授業の特性を考慮した上でシステムを構築したほうが望ましいことは言うまでもない。

最終課題は入門や1などでは基本的に教員サイドで用意した課題を作らせるが、2などでは学生各自でテーマを決めてもらい、各自で仕様書を作成してから製作に入る。この際、まず、仕様書レベルで合格しなくてはならない。すなわち、仕様書のチェックが入る。

このとき、仕様書を学生間で公開している。公開させる最大の理由は内容のバッティングの防止であるが、学生同士が刺激を受けることも期待できるため、このような方法を採用している。公開方法については、以前は授業時間に発表（プレゼン）してもらっていたが、十分な時間が取れなくなったため、WEB上での公開で対応することにした。当初、WEBへのアップは学生自身に任せていたが、管理上の問題もあり、あまり能率がよいとは言えなかった。

そこで、本研究では、仕様書ページを一括で管理するため、Wikiを使うことにした。Wikiであれば、学生がWEBデータ（仕様書ページ）を書き換えられると同時に学生全員のデータを一箇所で管理できる。これに伴い、面接予約システム自体もWikiをベースにすることにした。具体的にはPukiWiki [ 3 ] を用いることにした。PukiWikiはPHP言語で書かれており、改造も比較的容易に行なうことができることが特徴である。

その他、ビジュアルプログラミング演習の課題提出についても一定の仕組みが必要である。ビジュアルプログラミングの場合、一般に1つのプロジェクトで提出すべきファイルが複数存在するからである。そこで、プロジェクトフォルダごと圧縮したものを提出してもらうことにした。本システムでは、同時に圧縮ファイルの中身も解析することにした（後述）。

#### 4. 面接予約システム

以上のことを踏まえ、面接予約システムを開発した。主な特徴は以下の通りである。

- ・ PukiWiki ベース
  - ・ ビジュアルプログラミング演習授業の支援を考慮したシステム
  - ・ 面接予約だけでなく授業支援として利用
- 以下、システムの概要を説明する。

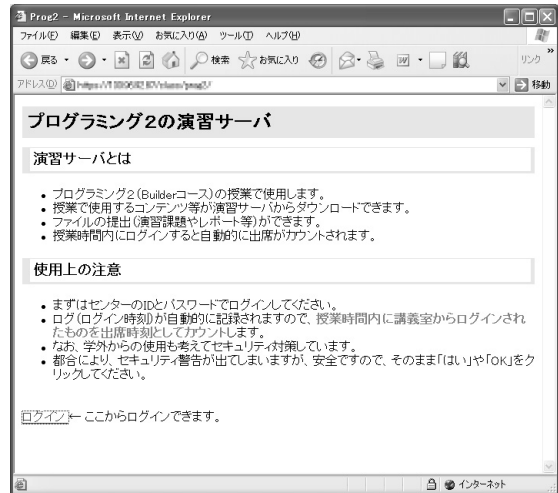


図1. ログインページ（学生、教員、共通）

#### 【学生用】

受講生は、最初に情報処理センターのユーザIDとパスワードでシステムにログインする。ログインに成功すると、授業用のページが表示される。このとき、授業時間内にログインした場合、自動的に出席がとられる。授業用のページでは、補足情報等を得ることができるほか、授業中に使用するコンテンツの配布等も行っている。

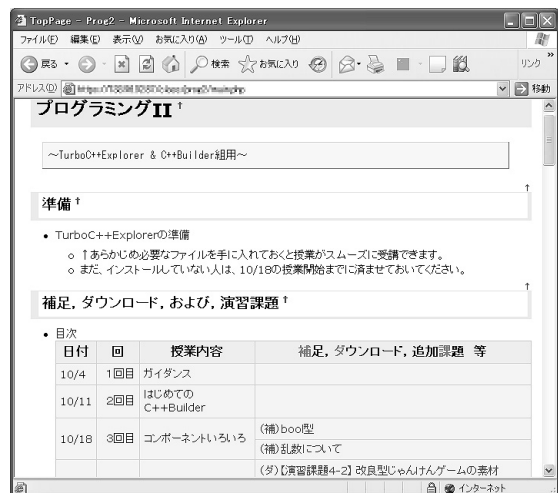


図2. 授業用ページ（学生用）

本システムのベースはWikiであるが、授業用コンテンツ配信のページなどは編集不能である。（つまり、通常のWEBと変わらない）

また、レポート等の提出もここから行うことが可能である。ビジュアルプログラミングにおいては、1つのプロジェクトに対して複数のファイルが使用されるため、提出には工夫が必要である。基本的には、必要不可欠なファイルはすべて提出が必要であるので、プロジェクトフォルダごと提出することが簡単ではあるが、プロジェクトフォルダの中にはコンパイルの過程で生成される中間コードなど不要なものも含まれてい

る。特にデバッグシンボルテーブルファイルなどはサイズが極めて大きいため、サーバリポジトリ、ネットワークトラフィック共に負荷がかかる。

そこで、本システムでは、提出された圧縮フォルダを自動解析し、足りないファイルは無いが、不要なファイルは提出していないか、表示するようにした。

具体的には、学生は以下の手順でプロジェクト（レポート課題等）を提出することになる。

1. プロジェクトの中から、デバッグシンボルテーブルファイルなど、不要なファイルを削除する。
2. プロジェクトフォルダごと、標準の（パスワードなどがかけられていない）zip 圧縮を行なう。
3. 圧縮フォルダファイル（つまり 1つの zip ファイル）を提出サーバに提出する。
4. サーバサイドでは、提出された zip 圧縮フォルダの中身を自動的に解析し、足りないファイル、不要なファイルを検出する。
5. 判定結果が画面に現れるため、学生は必要に応じて再提出する。

この結果、学生は課題を提出した時点で提出不備が無いかなど最低限の確認が可能となるため、以前より提出ミス（ファイル不足等）の数が減少した。

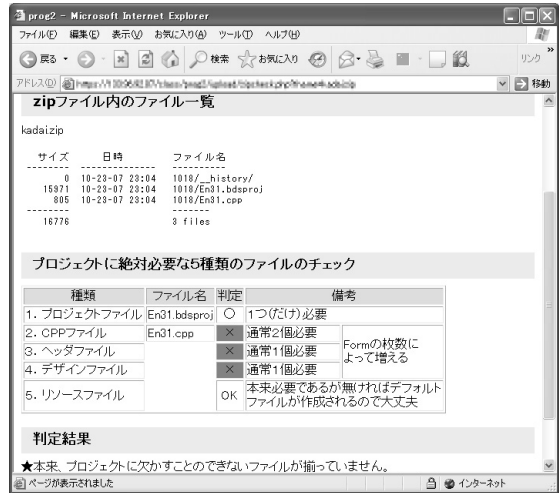


図 4 . 提出ファイルの自動判別（学生用）

次に、仕様書のアップについてであるが、これは Wiki の力を最大限に利用している。具体的な手順は以下の通りである。

- ・まず、教員が、仕様書の雛形ページを各学生の仕様書ページとして作成しておく。
- ・次に学生が、Wiki の書き込みモードで雛形ページを加工していく。
- ・具体的には、タイトルや名前等を書き込み、仕様書を作り上げていけばよい。

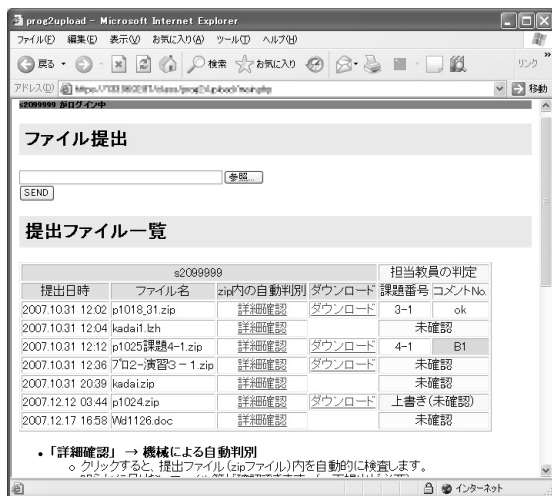


図 3 . プロジェクトの提出（学生用）

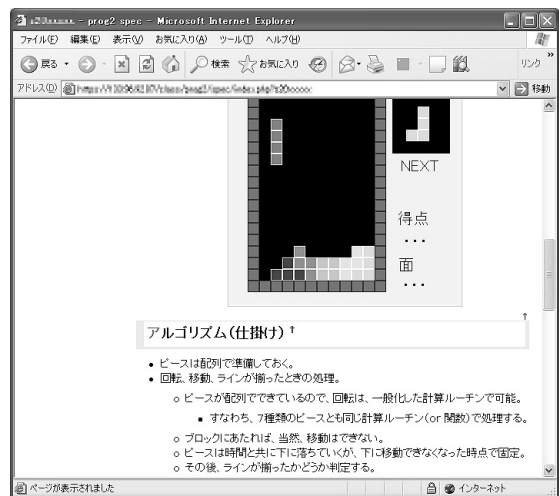


図 5 . 仕様書（学生用）

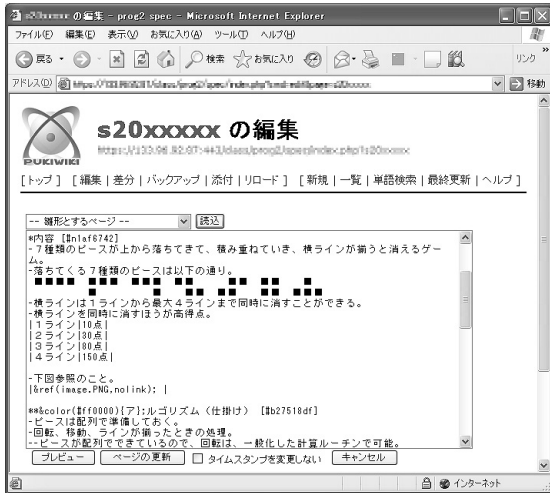


図 6 . 仕様書の編集 ( 学生用 )

続いて、面接予約について説明する。

まず、面接予約に先立って、学生は完成したプログラムを提出しておかなければならない。本システムでは、完成品を提出するまでは、面接予約ができないようにしている。これは、プログラムが完成していないのに都合のいい時間帯をあらかじめ押さえておくこと(いわゆるキープ)を防止するためである。無論、完成品の提出は本システムを用いて行なう。

完成品を提出したら、スケジュール画面まで進むので、予約可能な時間帯を確認して選択し、予約ボタンを押せばよい。無事に予約ができた場合、一覧表に学籍番号が提示されるため、あとは、登録した日時に面接を行なうだけである。

なお、仕組みの上では、ほぼ同時刻に複数の学生が同じ時間帯を希望すると、予約ボタンをあとから押した人が「登録失敗」になるはずである。無論、その旨のメッセージが出るため、失敗した人は、もう一度、別の時間帯で予約する必要があるが、本システムを稼働してから本論文の執筆時点までに、この現象はまだ一度も起こっていない。

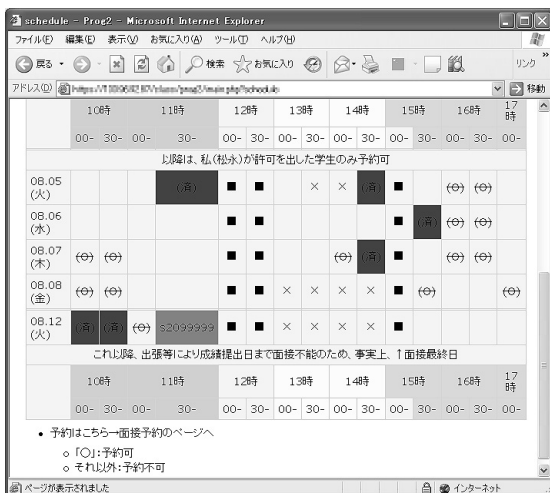


図 7 . スケジュールの確認 ( 学生用 )

なお、面接は一人の学生に対して複数回行なわれるので、面接時に「カルテ」を発行している。これには、学生への指導内容、次回までの課題や指示などが書かれている。特に、別の教員に面接してもらった場合、引継ぎ事項等の意味合いも兼ねているため、学生は再面接時に必ずカルテを持ってこなければならない。

このカルテも以前は紙で発行していたが、本システムでは電子カルテとして発行される。無論、学生も本システムにログインすれば自分のカルテを閲覧可能なので、確認したり、必要に応じて印刷したりできる。

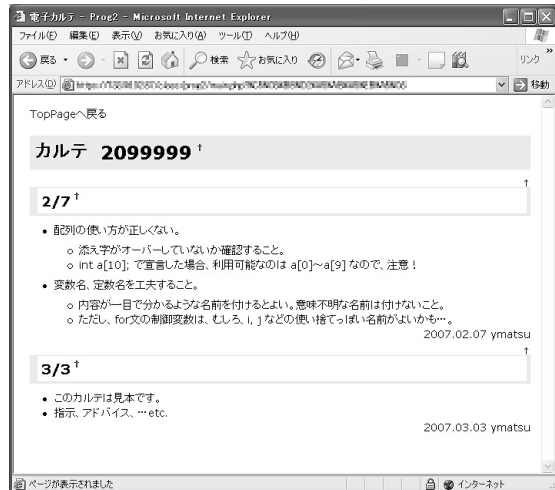


図 8 . 電子カルテ ( 学生用 )

電子カルテの場合、サーバ上に保存されているため、学生がカルテを持ってくるのを忘れて再面接ができない、ということも起こらない。

【教員用】

続いて、教員用の説明に入る。

教員の場合もログインページは学生と共通である。ユーザ ID から教員であることが判別できた場合、学生用のページと教員用管理ページが選択可能となる。教員用管理ページでは通常の Wiki の状態になるため、基本的にはすべてのページが調整可能となる。

管理者が行なうことが可能な主な内容は、以下の通りである。

- ・ 授業ページの更新・修正等
- ・ スケジュール調整
- ・ 電子カルテの生成・編集
- ・ レポート課題のコメント
- ・ 予約リストの確認
- ・ 提出物の確認

## 5. まとめ

以上、本研究では、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を行なった。その結果、面接予約がスムーズになった。また、本研究のシステムは Wiki をベースに作成したため、課題提出、仕様書提出、面接予約、電子カルテを連動して動かすことが可能となり、極めて効率のよい指導が可能となった。さらに、管理者はアナウンス等を簡単に修正できるため、管理者の負担も軽減した。

今後の研究課題としては、複数教員での同時利用、スケジュール登録時のインターフェースの向上、成績評価管理等があげられる。

## 参考文献

- [ 1 ] <http://www.media.osaka-cu.ac.jp/PEN/>
- [ 2 ] <http://www.php.gr.jp/>
- [ 3 ] <http://pukiwiki.sourceforge.jp/>

(2008年9月17日受理)

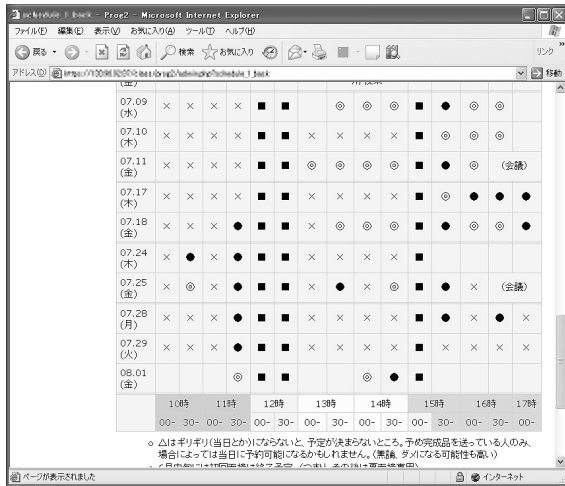


図9. スケジュール調整画面 (教員用)

なお、学生用のスケジュール確認画面においては、スケジュール表には本人分の学籍番号しか表示されないが、教員用のスケジュール確認画面においては、予約者全員の学籍番号が表示され、なおかつ、各学生の仕様書へのリンク、カルテへのリンクが張られている。その結果、カルテや仕様書を確認しながら面接を行なうことができるため便利である。また、カルテの存在の有無に関わらず、リンクだけは張られるようになっている。PukiWiki の性格上、実体のないリンクが張られている場合、リンクをクリックするとページが新規に作られることになる。そのため、初回面接時にはカルテのリンクをクリックすると自動的に電子カルテが生成されることになるので、手間が省けるというメリットもある。



図10. 予約リストの確認 (教員用)