プログラム自動採点システムF-Java*
～もっともっと演習を！～

板東 慶一郎, 近藤 大己, 長 慎也, 松村 真吾, 水越 拓也†
早稲田大学大学院理工学研究科‡
{ban,dai,cho,matumura, yamanasi}@kake.info.waseda.ac.jp

概要

プログラミング学習の初等教育において，学生に対し，演習問題を解く機会を多くの与えることが重要である。学生に実のある演習を多く行わせるためには，学生の演習に対する解答の評価，合否判定をすばやく伝えることが大事である。そこで，学生の作成したプログラムが問題の出題意図のとおりに動くかどうか自動で採点するシステムを作成した。本システムでは，プログラムの成否判定をメソッドとその戻り値という限定的なインタフェースにしぼり，正しいプログラム結果の形式を指定することができることを利用し自動採点を可能にした。また，出題者は問題の追加も簡単に行うことができ，その問題の内容も様々な単元，難易度に対応して作成することが可能である。生徒に多くの問題を解かせ，結果を即座にフィードバックさせることで有効なプログラミングの経験を積ませようという試みである。また，その上で，インタフェースにFlashを用いることにより，一般的な統合開発環境に近いわかりやすさの実現と簡単な実装を可能にした。

1 はじめに

プログラミング学習の初等教育において，学生に対し，演習問題を解く機会が多くの与えることが重要である。しかし，学生にプログラムのソースコードを提出させ，そのソースをコンパイし，実行してみて，正しい結果が得られるか確かめる，といった従来の採点方法では採点者に膨大な手間が掛かってしまう。さらに，その生徒の数が多くなったり，問題の数が多くなったりすると，その手間は膨大なものとなってしまう。また，採点に手間がかかってしまうということは学生側に採点結果がすぐにフィードバックされないという事にも繋がる。このような問題を自動採点という観点から解決しようと思い，プログラムの自動採点を行うシステムである「F-Java」を開発した。F-Javaは，自動採点の仕組みの導入により採点の手間を大幅に省き，生徒に対しプログラミング演習を行う機会を多く与えることを実現し，個々のプログラミング能力の効果的かつ効率的な向上を図るものである。

*Automatically Programming Marking SystemF-Java
†Keiichiro Bando, Daiki Kondo, Shinya Cho, Shingo Matsumura, Takuya Mizukoshi
‡Waseda University Graduate School of Science & Engineering

2 F-Java について

2.1 機能

Flash を用いた実習環境ページを用意した。それぞれの課題の出題リンクをクリックすると実習用ページが開く。学生はそのページの中でプログラムを作成・編集しすることができる。学生が作成したプログラムはサーバ上に保管され，コンパイルや実行についてもサーバ上で行われる。サーバ上で行われたコンパイル，実行，採点の結果は学生の Flash ページに返される。コンパイル，実行はサーバ上で行われるため，学生はプログラム実行環境を自分の PC にインストールする必要がない。

2.2 Flash によるインタフェース

本システムでは，UI 部分に Web ブラウザ+プラグインの形でリッチクライアントを実現する Flash を用いた。Flash のプラグインである Flash Player は現在のほとんどの PC に標準的に搭載されている。もしインストールされていなかったとしても，ブラウザが自動的にダウンロードを促すダイアログを表示してくれるので，それに従ってインストールを行うだけでよい。これによって，実行環境の構築作業に授業時間を大きく割く必要が無くなった。去年の授業では JDK を初回の授業を使ってインストールさせていたが，F-Java を使った今年度の授業では，Flash が動作するかどうかの確認ページへのアクセスだけで済んだ。また，高い表現力を持つ Flash を UI として利用できるためストレスのないプログラムの編集作業が可能である。

2.2.1 基本画面

画面は大きく分けて「問題文」「エディットボックス」「結果通知欄」「ボタン類」の 4 つの部分に分かれている (図 1)。学習者はこの画面だけで，問題文を読みながらプログラムの記述・コンパイル・実行のすべての作業を行う。保存・コンパイル・実行などの操作は，「ボタン類」に配置されているボタンを押すだけで行うことができるので，コマンドラインで命令を入力する必要はない。

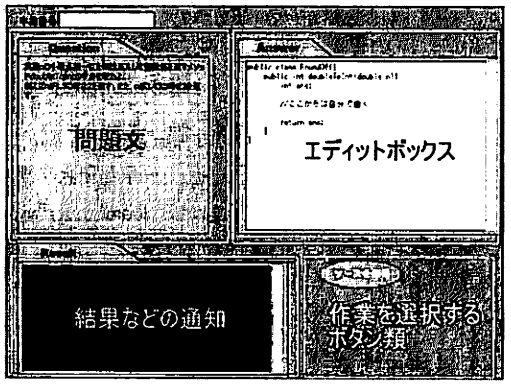

図 1: 基本画面

2.2.2 ポップアップウインドウ

各種作業結果の通知などに Windows ライクなダイアログウインドウをポップアップさせたり，画面内のテキストボックスを書き換えたりすることで画面遷移を 1 画面に留め，ユーザビリティを大幅に向上させる工夫をしている。

図 2 はプログラムのコンパイルに成功したとき，図 3 はコンパイルが失敗したときの画面である。また，図 4 のようにコンパイルや実行の際に起こったエラーに対する簡単なアドバイスを表示できるようにもなっている。

2.3 採点

2.3.1 出題形式

F-Java を用いた問題の出題形式は，入出力の形式が決められたメソッドの中身を書かせるという形をとっている。また，この出題を助けるための仕組みを設けた。出題者は，この出題支援システムを使って，問題文，解答となるプログラムのクラス名，出題対象のメソッド名，メソッドの引数の型と変数名，戻り値の型を入力する。これらの情

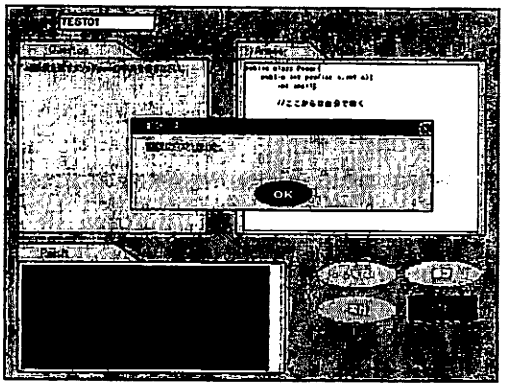

図 2: コンパイル成功時画面

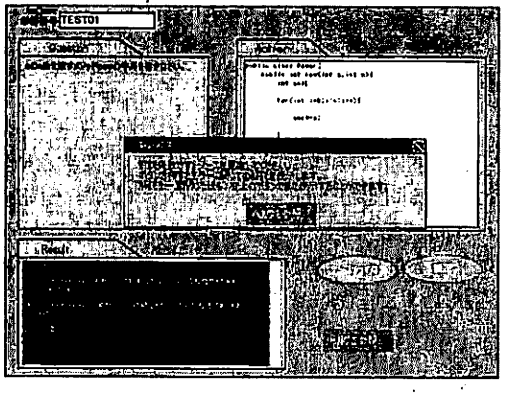

図 3: コンパイル失敗時画面

報を入力すると，問題文，クラス名，メソッドの外枠だけが書かれたソースのテンプレートが出来上がる。出題者は必要に応じてこのテンプレートの中身を書き換える。これらの情報はXMLファイルとして保存され，出題時にFLASHはこのXMLファイルを読み込み，学生側に問題・ソースのテンプレートを表示する。また，出題者はその動作判定を行うための，引数値とそれに対する正しい戻り値を並べて書いたテストケースも入力する。このデータもファイルとして保存され，自動採点時に呼び出される。

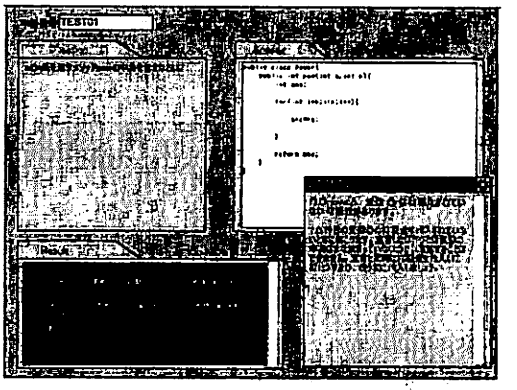

図 4: ヘルプ画面

2.3.2 採点方法

テストケースに用意されている値を引数として，こちらが用意したメソッドを指定して実行する。その戻り値を参照し，正しい答えが返ってきているかどうかを判断する。すべてのテストケースに対し正しい結果を返すメソッドが書かれたプログラムを題意を満たすものとして判断する。メソッドの入出力だけに基づく採点方法では，効率的に動くプログラムであるか，最適のアルゴリズムが適応されているプログラムであるかどうかは採点の判断材料にはならない。しかし，プログラミングの初頭学習において一番重要であるのは，問題を解決するようなプログラムを自分の頭で考え，それを書くということであり，その観点に立てば，メソッドの中身について制限無しに自由に書くことのできるこの採点方法は有効であることが言える。

2.3.3 アドバイス機能

出題者はテストケース作成時に，学生に対し，何故そのテストケースで引っ掛ってしまったのかということを伝えるアドバイスを入力することができる。このアドバイスは，採点時にそのテストケースで間違えると学生に表示される。(図 5) これにより，学生は間違ったテストケースに対して

的確なアドバイスを受けることができ，そのアドバイスを元に自分のプログラムを見直すことができる。

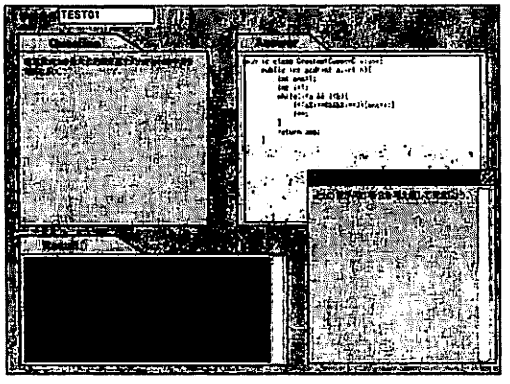

図 5: アドバイス画面

2.3.4 メソッド実行

学生に対し，実際に自分が中身を埋めたメソッドがどのような挙動をするかを確かめる仕組みであるメソッド実行機能を用意した。メソッド実行機能では，学生自らが引数の値を決めてメソッドを実行し，どのような値が返ってくるのかを見ることができる。(図 6) このメソッド実行機能により，提出の前に，プログラムがある入力に対し正しい結果返すかどうかを確かめることができる。

3 システムの評価

3.1 授業への導入

今回は，大学 1 年生前期のプログラミングの授業の後半において，学生数が約 100 人のクラスに対し，1ヶ月間自習教材という形で本自動採点システムを実際に導入した。各単言ごとに用意された問題を，学生は F-Java を使ってその問題を解いていく。学生はテストボタンを押すことにより，その場ですぐにそのプログラムが正しい結果を返すかどうかを知ることができる。学生の書いたプログラムのソース，コンパイルや実行，自動採点の結果は全て逐次データベースに格納するようになっている。

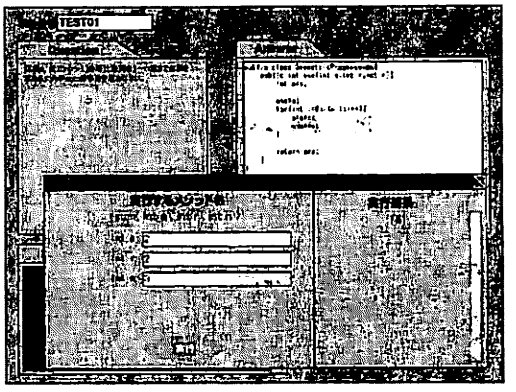

図 6: メソッド実行時画面

自習問題は全 15 問あり，「四則演算，変数と型」に関する問題が 2 問，「制御構造文」に関する問題が 3 問，「配列」に関する問題が 5 問，「メソッド」に関する問題が 4 問，「オブジェクト」に関する問題が 1 問という内容になっている。

3.2 学習効果

授業の後半においてプログラミングに関する模擬試験を行った。それまでに自習問題に取り組んだ問題数と試験の点数との関係をグラフにしたものが図 7 である。図 7 より自習問題に多く取り組んだ学生ほど成績が良いことが分かる。逆に，自習問題に取り組んではいるが，その問題数が少ない場合，全体的に成績が低くなっている。また，自習問題に取り組んだ学生に対しする自習問題の取り組み問題数と試験の点数の相関係数は 0.55 となり，やや弱くはあるが正の相関関係が確認できた。つまり，自動採点システムにより，多くの問題を提供することができるので，その取り組み量から各学生のプログラミングに対する理解度もある程度推し量ることができる。

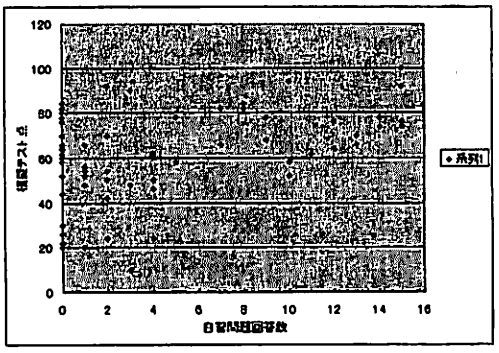

図 7: 自習問題回答数と模擬試験の点数の相関グラフ

3.3 学生の評価

F-Java システムを使ったすべての学生に対し「F-Java システムは使いやすかったか」についてアンケートを取った。その結果は図 8 のようになった。ボタンを押すだけでコンパイルや実行の作業を行うことができたり，ヘルプ機能によるサポートを行ったりと，FLASH によってわかりやすいユーザインタフェースを実現したことにより，学生から高い評価を得る結果となった。

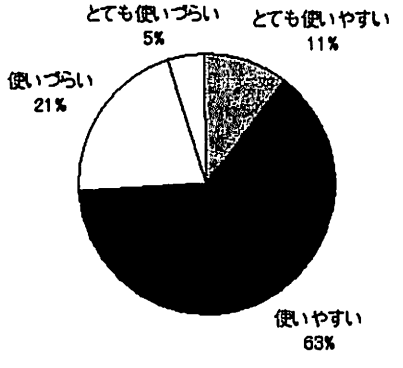

図 8: アンケート結果 1

自習採点システムを使っている学生に対し，「自動採点システムによる自習問題はプログラミング学習の上で役に立ったか」についてアンケートをとった。その結果は図 9 のようになった。多くの学生から高い評価を得る結果となった。

また，模擬試験の結果との対比も行ってみた。アンケートにおいて「とても役に立った」と答えた学生の平均点は 75.0 点，「役に立った」と答えた学生の平均点は 60.4 点，「あまり役に立たなかった」と答えた学生の平均点は 37.0 点であった。自動採点システムによる自習問題に対する評価が試験の成績にも反映される結果となった。

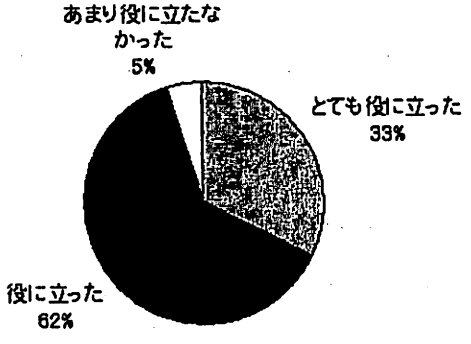

図 9: アンケート結果 2

4 従来の自動採点システムとの比較

同じような自動採点システムとして，プログラミングコンテスト [1] で用いられている PC^2[2] がある。しかし，PC^2 については，標準出力の出力結果から判断して採点を行うため，回答者に対し表示形式を整形しなければならないという負担がかかってしまう。対して，本システムでは，採点に関して標準出力部分は全く見ていないので，表示形式を整形しなければならないといった生徒側の負担は全くない。

自動コンパイルを行うプログラム提出システムとして[3]がある。コンパイルチェック・基本的な動作確認を行うことにより，学習者から提出されるプログラムの品質が保障され，採点者の手間を省くというシステムである。最終的なプログラミングの正誤評価は，採点者にゆだねられ，自動採点というよりかは採点支援の側面が強い。

C言語による比較的大きなプログラミング課題のための自動採点システムとして[4]がある。このシステムでは実行結果をみてキーワードが指定された順番に出力されたかどうかにより正誤判定を行う。また，中級者向けの課題の詳細な判定のためにデータ構造の検査・メモリ確保/解放関数が適切に使用されてるかなどソース内容の検査も行う。ある入力に対し，絶対的な正答が存在しないため，正誤判定の精度が100％保障されるものではない。

5 おわりに

今後も本校のプログラミングの授業において本システムを続けて使い，データベースに蓄積されたデータや，学生に対するアンケート，期末テストの成績から学習効果などをさらに深く分析する。その結果は追って報告する予定である。

参考文献

[1] ACM/ICPC
http://www.teu.ac.jp/icpc/jp/index.html

[2] CSUS Personal Computer Programming Contest (PC^2)
http://www.ecs.csus.edu/pc2/

[3] 望月将行, 森田直樹, 北英彦, 高瀬治彦, 林照峯, "自動テスト機能を備えたプログラム提出システム," 2003 PC カンファレンス, pp343-344, 2003.

[4] 田上恒大, 阿部公輝, "比較的大きなプログラミング課題のための自動採点システム," 情報処理学会研究会報告（コンピュータと教育）, pp.135-140, Feb. 2006.