

プログラミング言語演習科目のための支援システムの構築

小坂 敏文*1, 吉本 定伸*1, 松林 勝志*1

A Support System for Programming Language Exercise

Toshifumi KOSAKA, Sadanobu YOSHIMOTO, Katsushi MATSUBAYASHI

It is strongly recommended for students in computer science course both to write lots of computer programs and to read good programs to master programming languages. In addition, it is considered that practice is one of good method to study engineering. It is hard for teachers to collect and correct lots of assignment reports with printed material. For teachers in charge, the collecting and inspecting reports system on web-site has been developed. In the system, the students are asked to upload their reports at the special website. The system checks students' programs and programming styles automatically. To build the system a teacher is asked to makes only several CSV files that are the tables of students IDs and the lists of assignment reports. The system reduces teachers load effectively, and they can offer a great deal of care to students.

KEYWORDS : program language exercise, web support system

1. はじめに

学生にプログラミングを教えるとは、課題を解決する手段を考え、それを言葉で表し、プログラミング言語で表現出来るようにすることである。そのためには、例題プログラムを追跡する練習も必要だが、最終的にはプログラムを作成する機会を多く与える「プログラム作成中心のプログラミング教育」が必要である。作成課題が多いということは学生にとっては、大変だがやりがいがあることであり、指導教員にとってはその添削作業が非常に大変になる。これを実行するためには、指導教員側のロードをサポートする必要がある。ここをしっかりと行うことで、プログラミングスキル水準を保った卒業生を送り出すことにつなげることができる。この考え方のもとに教員の課題収集・課題チェックを支援するシステムを構築したので報告する。

2. プログラミング

プログラミングは、コンピュータが行う作業という制約を考慮しながら課題の解決手順を考え、その解決手順をプログラミング言語に翻訳する作業である。少ない語彙で、言い回しの種類が少なく、自然言語の修得とは異なる困難さがある。

情報系の学生は、課題解決能力の下層部分であるプログラミング言語を、低学年で学ぶ。本校情報工学科では、2年生から初めてのプログラミング言語としてC言語を学んでいる。そこでは、主に制御構造の修得に主眼が置かれている。

このプログラミング言語の修得においては、これまでの実体験や思考体験の違いから、修得の速さにおおきな開きが生ずる。そのため一斉授業はほとんど意味をなさない状態が生まれることは、多く経験されている。

*1 東京工業高等専門学校情報工学科 (Dept. of Computer Science, Tokyo National College of Technology)
〒193-0997 東京都八王子市栢田町 1220-2 E-mail kosaka@tokyo-ct.ac.jp

修得速度分布が非常に広がった状態にある学級学生集団に対しては習熟度別対応が必要になる。しかし、習熟度別クラス編成は物理的に不可能であるため、教育リソースを駆使して、疑似習熟度別学習内容が必要になる。また、学生の中には、すでに授業内容を超えて創造性に富んだ作品に挑戦し、さらに能力を磨きたいと願う学生もいるが、その部分は外部コンテストなどを活用した別な教育手法が必要になるので、別の報告にゆだねる。本報告は大多数の学生のプログラミング言語の修得を目指す通常の授業システム構築に限定することにする。

プログラミングの修得には、

- (1) コンピュータの作業という制約を考慮しながら課題の解決手順を考えられること
- (2) 「プログラミング言語の文法」と「プログラミング言語特有の言い回し」にしたがって解決手順を記述できること
- (3) 記述されたプログラムの振る舞いを追跡でき、意図しない動作の原因究明ができること（データ構造の読み解きも含む）
- (4) 記述したプログラムが、課題で示された内容を十分に達成していることを確認できること（プログラムの検証が出来る）
- (5) コンパイル時エラーの指摘を理解し、適正に対処できること
- (6) 一定の字下げルールを用いた見通しの良いプログラミングスタイルを獲得すること

が、主に必要である。プログラミングを修得させる過程で、将来、別な言語の修得を自分で行うための能力を修得させることも重要である。自分で説明や解説を読んで修得する態度を養成しなければならない。

3. プログラミング言語を修得させるために必要なこと

プログラム作成においては、大別して reading と

writing の2つの側面があり、最終的には writing を行うのだが、前提として reading が出来ないと writing もできないという事実がある。

3. 1 reading

reading の範囲では、「プログラミング言語の文法」と「プログラミング言語特有の言い回し」を修得し、自分で作成したプログラムが意図しない動作をしても追跡できるようにするために、以下の内容が必要である。

- (1) 文法の解説とともに、多くのプログラム例を読む環境
- (2) インタプリタのように、1行実行のたびにプログラムの行の意味を解説し、実行の様子を学習者に動的に見せる教材
- (3) 主体的な学びの姿勢を促すために試験形式でプログラム実行の様子を追わせる教材

3. 2 writing

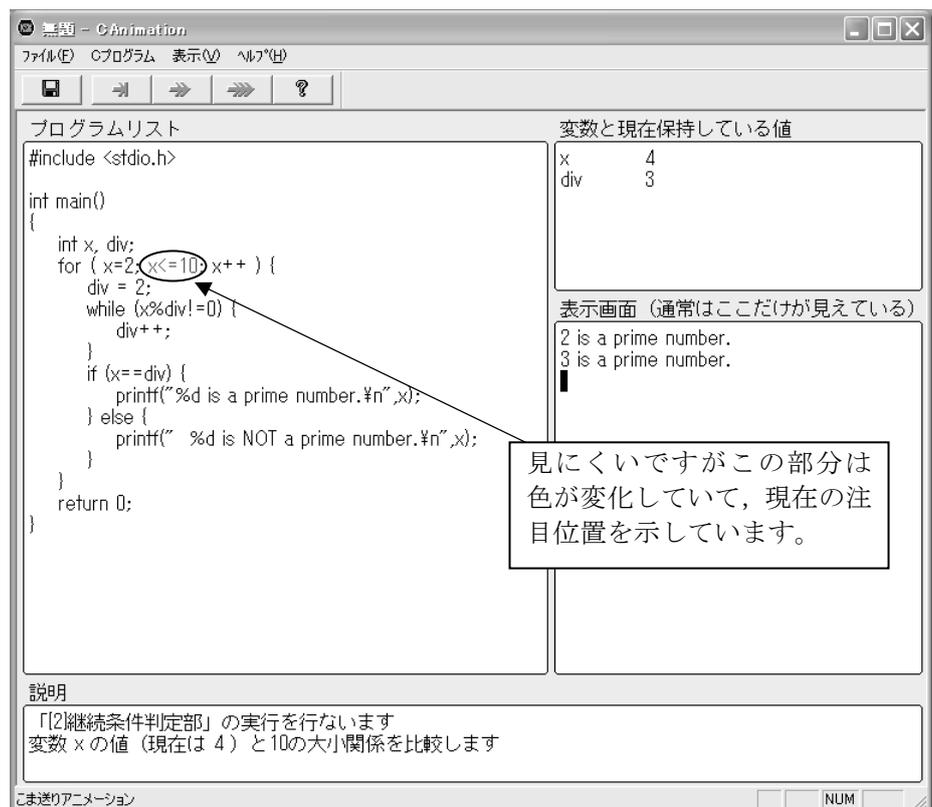


図1 「プログラムのステップ実行」説明アプリケーション for 文の条件判定部ステップ実行の様子である。1つのステップは、直前・直後の2回説明が表示される。現在表示されているのは、比較作業の直前の説明である。

writing の範囲では多くの課題の演習を含む次のような教育システムが必要である。

- (1) 課題提出状況を簡単に教員が把握できる
- (2) きめ細かく文法をチェックし、学生にフィードバックする
- (3) 字下げスタイルをチェックし、学生にフィードバックする
- (4) プログラム実行結果をチェックし、学生にフィードバックする
- (5) 教員からのコメントを学生に与える
- (6) 学生からの質問などに答える。

しかし、科目担当教員がすべてを行うことは限界があるため、これを克服する工夫が必要となる。

4. 構築した教育システム

「3」に述べたニーズに合わせて、教育システムは Web 教材+課題オンライン収集システムにより構築されている。いわゆる e ラーニングと異なり、教員が行うべき指導の一部を支援するシステムであって、学生がこのシステム上で練習問題を解くものではない。

4. 1 プログラム読解力向上のためのシステム

reading の範囲においては Web 教材を中心として、次のようなシステムを構築した。

「プログラミング言語の文法」と「プログラミング言語特有の言い回し」を修得させるための環境として紙媒体の教科書でもよいが、紙面制約がなく、多くのプログラム例を掲載でき、必要に応じて新しいプログラム例の追加なども柔軟に出来るように Web テキストを作成し、利用している。¹⁾ この Web テキストは 2003 年からメンテナンスを続けながら使用しているもので、C 言語文法学習のためのチュートリアルになっている。「記述されたプログラムの振る舞いの追跡」のために Web テキスト中に 3 択問題練習を配置した。ま

たこの Web テキストにはプログラム課題が章ごとに配置されており、先進的な学生は授業ペースにかかわらず課題を進めることができるようになっている。なお、作成し提出するプログラムリストには、実行結果をコメントとして含ませるようにしている。

C 言語を学び始めて最初に出会うプログラム制御構造、すなわち繰り返し構造と条件分岐構造の理解に戸惑う学生がいる。そのような学生には、図 1 のようなインタプリタ形式で、プログラムを 1 行ずつ解説しながら動作する「プログラムのステップ実行」説明アプリケーション²⁾を紹介している。このアプリケーションでは、ある 1 行を実行するときに、「この行の意味は〇〇なのでこの変数をこのように変化させます。行の命令を実行したらこのように変数が変化しました」というように 2 ステップに分けて説

2J01秋山雄二さんの [プログラミング言語] 作業

2011/6/13 15:21

1. 教員からのコメント

誤りを正して再度提出して下さい。

2. 課題提出履歴

R: 受領(何らかの問題点を指摘することがある)
r: 期限遅れで受領(何らかの問題点が指摘されることがある)(遅れ日数表示あり)
N: 課題ID(ファイル名表示)またはユーザIDが不正
F: 字下げルール抵触
E: コンパイルエラー(コンパイルできない)
S: プログラム仕様違反(プログラムコードに仕様違反がある)
P: 実行結果の表示部分が不正(正しい実行結果が貼り付けられていない)
X: 提出後実行時の実行結果が不正(提出されたプログラムを実行したが、結果が変わり)
Z: 未提出

課題	提出期限	提出日	再提出日	状況
p01ex01.c	2010/8/20	2010/8/5	2010/8/5	R
p01ex02.c	2010/8/20	2010/8/5	2010/8/5	RF
p01ex03.c	2010/8/20	2010/8/5	2010/8/5	RP
p01ex04.c	2010/8/20	2010/8/5	2010/8/5	RE
p01ex05.c	2010/8/20	2010/11/11	2010/11/11	rN
p01ex06.c	2010/8/20	0/0/0	0/0/0	Z
p01ex07.c	2010/8/20	0/0/0	0/0/0	Z

3. 課題提出

提出ファイルを設定してから「アップロード」ボタンを押してください。

提出ファイル:

4. 教員向けコメント送信

教員向けコメントをここに書いてから下の「コメント送信」ボタンを押してください

図 2 学生がレポート提出する個人用 Web 画面

このページに入るには、学生は ID、パスワードの入力が必要である。提出課題の現状や、教員からのコメントも見える。下部には、パスワード変更欄もある。(以下の図はすべてデモであり、実在の氏名・課題ではない。)

明している。特に「for 繰り返し構造」では for 文の中を 3 つに分解した要素にまで分けて逐次解説しながら動作している。プログラムの制御構造がなんとなく判ったら、このアプリケーションで自分の理解とプログラムの動作が同じになるかどうかを見て、自分の理解が正しいかどうかを確認できる。

また、定期試験では主にプログラムの振る舞いの追跡を問うようにしており、定期試験も教材の一部と位置づけている。課題で作成したプログラムの一部に動作上の誤りを含ませた問題などが、学生の力を見る上で大事である。さらに多くの試験問題（過去の定期試験問題）を公開し、練習できるようにしている。

4.2 プログラム作成力向上のためのシステム

writing は総合力の養成になるので、課題プログラムを時間が許す限り多数作成させるようにしている。しかも教員が適切な添削とコメントを与えることが必要である。紙媒体での課題提出ではチェックすることはほとんど不可能である。また、メール添付ファイルによる課題提出も集計に多大な時間がかかり、現実的ではない。e ラーニングシステムには課題提出状況を管理してくれるものもあるが、それ以上の手助けはしてくれない。そこで教員の作業を支援する次のような課題オンライン収集システムを作成した。

このシステムは、学生からみると、図 2 に示す Web 画面から次のことが出来るようになっている。

- (1) ID、パスワードを使って、課題を電子提出できる
- (2) 課題の初歩的誤り（コンパイルエラー、字下げルール抵触、実行結果未貼付、課題番号・学生番号の不正）は即座に指摘され再提出が出来る
- (3) 担当教員へのメッセージを伝えることができる
- (4) 教員からのコメントを受け取ることができる

2J 「プログラミング言語」 レポート提出集計

2011/6/13 15:32

[ダウンロード](#) 集計表を csv 形式でダウンロード

○: 期限通り提出
 △: 期限遅れ提出 (遅れ日数表示あり)
 ▲: 提出されたが、再提出を求められている
 ×: 提出されていない

N: 課題ID(ファイル名表示)またはユーザIDが不正
 F: 字下げ(ルール抵触)
 E: コンパイルエラー(コンパイルできない)
 S: プログラム仕様違反(プログラムコードに仕様違反がある)
 P: 実行結果の表示部分が不正(正しい実行結果が貼り付けられていない)
 X: 提出後実行時の実行結果が不正(提出されたプログラムを実行したが、結果が変化した)

No	ID	p01ex01.c	p01ex02.c	p01ex03.c	p01ex04.c	p01ex05.c
		2010/8/20	2010/8/20	2010/8/20	2010/8/20	2010/8/20
1	2J01 秋山雄二	○	▲F	▲P	▲E	▲N
2	2J02 飯坂孝平	×	×	×	×	×
3	2J03 小倉理恵	×	×	×	×	×
4	2J04 金沢啓太	×	×	×	×	×
5	2J05 木村雄平	×	×	×	×	×
6	2J06 熊沢直	×	×	×	×	×

図3 教員が提出状況を確認する Web 画面
 提出状況が一覧できるほか、csv ファイルとして取得も出来る。

2J 「プログラミング言語」 レポート閲覧

提出期限 2010年8月20日

1	2J01	秋山雄二	p01ex01.c	p01ex02.c	p01ex03.c	p01ex04.c	p01ex05.c
2	2J02	飯坂孝平	p01ex01.c	p01ex02.c	p01ex03.c	p01ex04.c	p01ex05.c
3	2J03	小倉理恵	p01ex01.c	p01ex02.c	p01ex03.c	p01ex04.c	p01ex05.c
4	2J04	金沢啓太	p01ex01.c	p01ex02.c	p01ex03.c	p01ex04.c	p01ex05.c
5	2J05	木村雄平	p01ex01.c	p01ex02.c	p01ex03.c	p01ex04.c	p01ex05.c
6	2J06	熊沢直	p01ex01.c	p01ex02.c	p01ex03.c	p01ex04.c	p01ex05.c

図4 教員が提出課題を巡回目視するための Web 画面
 表内リンクを順番にクリックしてゆくと各プログラムの目視検査、およびコンパイル実行をすることができる。

- (5) 現在の自分の提出状況が判る
 - (6) クラス全員の提出状況が判る
 - (7) クラスの友人の提出課題が見える（これは見えないようにすることもできる）
 - (8) 自分のパスワードは自分で変更できる
- また、教員からみると、図3の Web 画面により次のことがいつでも出来るようになっている。
- (1) 一覧表画面による提出状況・提出期限遵守状況のチェック
 - (2) 提出状況一覧表 csv ファイルの自動生成
- また、図4の Web 画面により

- (1) 特定の課題について学生毎に作品を順にたどること
- (2) 特定の学生について課題番号順に作品をたどること

が出来るようにになっている。作品プログラムを精査しコメントを学生に与える時には必要な機能である。クリックすれば、プログラムリストを見ることができ、コンパイル・実行して検証できる。

学生と教員間でコメントやメッセージを通信するために図5のWeb画面により次の2つの機能を持っている。

- (1) 学生向けコメントの送信

- (2) 学生からのメッセージ読み出し

教員には意識されないが、学生がプログラムをアップロード提出するたびに以下のことが行われる。

- (1) 提出されたプログラムのコンパイル自動チェック（これが出来るだけでも大いに助かる）
- (2) 提出されたプログラムの字下げルールの自動チェック（本システム稼働以前では、学生へのコメントはほとんどがこの内容であった）
- (3) 提出されたプログラムの実行結果の自動チェック（自動では厳密にはできないので、教員の精査が必要）

さらに教員を支援する内容として以下の機能を持っている。

- (1) 外部評価向けのエビデンスとして、「全提出課題をリンク付きのWebサイトとして見えるイメージ」の圧縮ファイルを生成
- (2) 学生がパスワードを忘れてしまう場合に備えて、ワンクリックで個々の学生のパスワードを初期化する

これらの内容をもった、課題提出・チェックシステムを構築したことで、教員は提出された多くの課題に埋もれたり、提出状況チェックに追われたりすることなく、適切なコメントを発し、学生の学習に主眼を置いた指導ができるようになった。

5. 課題オンライン収集システムのための教員の準備作業

どんなに教員支援能力が高くても、年に2回しか行わない初期設定が複雑なら使え

ないため、初期設定作業を出来る限り単純化した。このシステムでは、図6のWeb画面において

- (1) 学生リストを与える学級情報 csv ファイル
- (2) 課題リストを与える科目情報 csv ファイル
- (3) 上記2つのファイルの結びつきを指示する csv ファイル

をアップロードし、システム構成をボタンで指示す

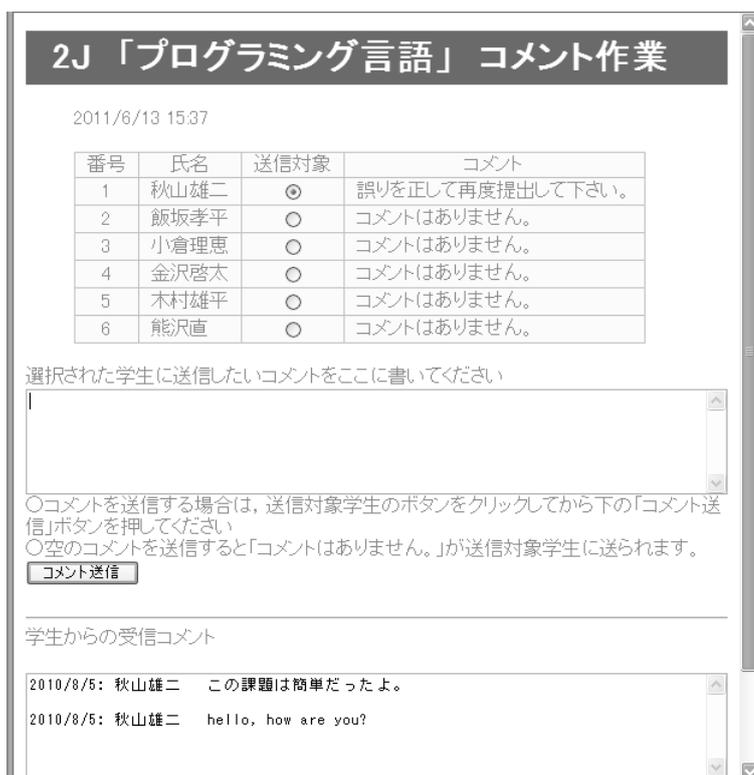


図5 教員がコメントを送信するWeb画面
送信対象を選んで、コメントを書いて、送信する。逆に学生からのメッセージを読むことができる。

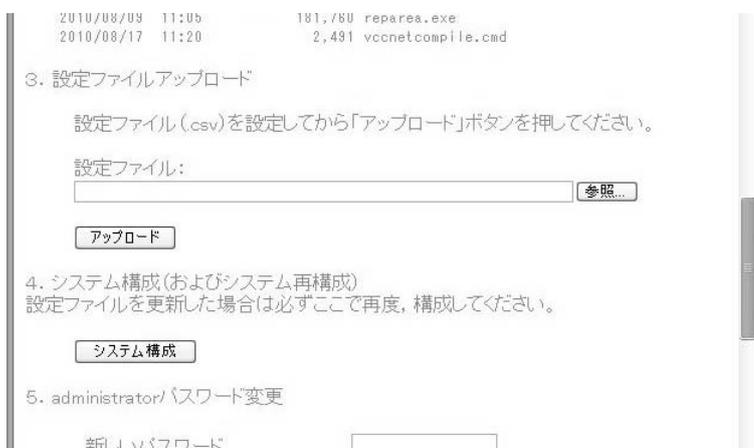


図6 教員が初期設定するWeb画面
授業期間開始時期に設定する。途中で科目の追加設定も可能である。

るだけで全システムが構成される。

学級情報 csv ファイルは、学生氏名とユーザ ID (初期パスワードはユーザ ID と同じ) のリストであり、科目情報 csv ファイルは、課題ごとの提出期限、課題に含まれるべき文字列 (正規表現) などのリストである。

このシステムは PC (Windows7) 上で Apache および php, そしてコンパイラとして Microsoft Visual Studio 2010 を利用して動作させており、学内 LAN 内で運用している。

6. 学生アンケート

現在授業を行っている情報工学科 2 年生 44 名を対象にアンケートを実施した。ちょうど繰り返し構造を学び、次に条件分岐構造を学ぶところでの実施となった。実施結果を図 7 に示す。

全体にポジティブな受け止め方をされているのが判る。自分のペースで課題作成を進めることができることがをよいと感じているようである。特に、自分で資料を読みながら言語習得する方法を修得したことは、今後の自学自習での言語修得が期待される。

プログラムステップ動作アプリの利用率が低いのは、まだ繰り返し構造など学ぶ前に紹介したため、このアンケート直後に再度紹介している。

7. まとめ

本システム稼働前は、受領したプログラムリストのコンパイルを手動で行ない、コンパイルエラーの有無をチェックし、また字下げルール抵触へのコメントを多く与えていた。これらが自動化され、担当教員は実行結果のチェック (これも簡単な文字列チェックが自動で行われた後の作業) を行い、本質的なチェックとコメント生成に時間を割くこと出来るようになった。Web テキスト上の事前提示課題により、学生たちは自分のペースで課題プログラム作成、提出、自動チェック、担当教員チェックのループで先に進むことができるようになった。そして、課題演習に重きを置いた学習により、プログラミング言語の自学自習方法の修得に結びついている。

本システムおよび教育の課題として 1) 技術的課題として、教員・学生の ID 管理を一般化、2) 学生の使い勝手として、複数課題同時提出、3) 教育効果上の課題として、教材と課題のつながりの精査、4) 学生たちのモチベーションアップのため、本システムのような実用システムの舞台裏をのぞかせる

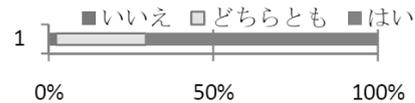
ことを考えている。

参考文献

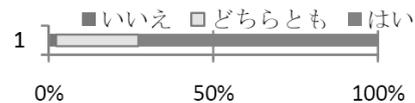
1) http://xythos.tokyo-ct.ac.jp/usr/kosaka/web/for_students/CIntro/CIntro.html (2011.6.15 現在)

2) <http://www.vector.co.jp/soft/win95/edu/se165337.html> (2011.6.15 現在)

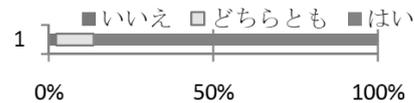
プログラミングの学習は充実していますか。



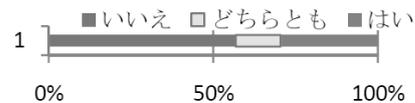
Webテキストをしっかりと読んでいますか。



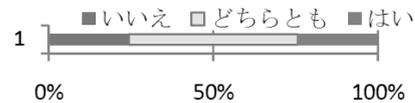
Webテキストを読むことで理解が深まりましたか。



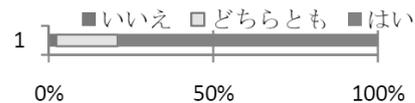
プログラムステップ動作アプリを利用しましたか。



利用時に、理解の正しさが確認できましたか。



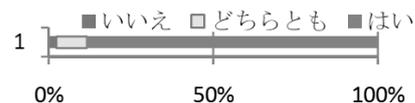
作成課題は自力で解くことが多いですか。



課題提出方法に慣れましたか。



マイペース進行授業方式は都合よいですか。



「自分で言語修得」の方法を修得できましたか。



図 7 学生アンケート結果