

## 論文

# ファイル共有サービスを利用した プログラミング教育支援システムの開発

上地 宏一

一般向けファイル共有サービスを活用し、プログラミング演習・自習状況を遠隔でリアルタイムに確認できるプログラミング教育支援システムを開発した。状況の確認にとどまらず個々の受講者が入力したプログラムと教師の手元の完成版の間にどのような差異があるかといった差分表示や、手書きで指導事項を書きこめる描画機能を併せ持つ、実用的なシステムを実装することができた。本稿ではシステムの詳細を紹介する。

キーワード： Web アプリケーション、プログラミング教育、授業支援システム、ファイル共有

## 1. はじめに

プログラミング教育において教科書の例示プログラムを入力し実行する演習は非常に重要である。文系学部におけるプログラミング科目は、そもそもの学習時間が少ないことだけでなく、理系のそれと異なり SA や TA のクラス配置が望めない。このような受講者・教師双方にとって厳しい環境の中で効率的にプログラミング教育を実施するために、ファイル共有サービスを利用した教育支援システムを構想したことは既報の通りである。本稿では実装したシステムについて述べる。

## 2. システムの全体像

本システムは「プログラミング教育支援システム Programing Education Support System」の頭文字をもとに Progress と命名した。「Progress」は演習の進捗を把握するシステムであることと、プログラミング学習が進展してほしいという願いも含有している。システムの利用対象者は主として教師であるが、一部の機能を受講者が利用することも想定している。

### 2-1 システム本体

本システムは Web ブラウザからアクセスする Web アプリケーションであり、システムの実体は Web サーバに設置した CGI プログラムと補助プログラムの集合体である。CGI プログラムは Perl 言語で実

装され、補助プログラムは Python 言語を使っている。ページ内クライアントサイドプログラムは JavaScript 言語となる。

システムは以下の4ページから構成される。

①認証ページ

①状況把握ページ：全体・部分の状況を一覧

②差分表示ページ：特定の1ファイルの内容と正答との差分を表示

③添削ページ：特定の1ファイルに添削

①はシステム利用者の認証ページであり、認証が成功すると①のページURLを取得できる。①、②および③はそれぞれページ内リンクで結び付けられ、お互いを行ったり来たりできる。教師は①によって各受講者の状況を把握し、必要に応じて②によって特定の受講者のプログラム入力ミスを確認する。リモート指導の場合は③によって指導内容をプログラムファイルに直接描画し、メール等で連絡することになる。細かい説明は後述する。

## 2-2 受講者および教師の授業準備

受講者はDropboxアカウントを取得し(あるいは既存のアカウントを使い)、受講専用のフォルダを作成する。そのフォルダの中に教科書の例題等のプログラムを入力していく。事前に自分のPCにDropboxクライアントアプリケーションをインストールすることにより、Dropboxフォルダ内のプログラムファイルはクラウド上に同期される。その受講専用のフォルダを教師と1対1で共有しておく。この準備作業を第1回目の授業時に行う。

次ページの図1は教師側のDropboxフォルダの中身である。2つの点線枠が受講者と教師との間でフォルダ共有を行っている(=受講者側のPCと同じ状態に同期されている)受講者2名分である。受講者は「s + 学籍番号」という名前のフォルダをプログラミング系科目受講専用のフォルダとし、その中に受講する科目を示すフォルダを作成する。例では「prog1」が「プログラミング基礎」、「prog2」が「プログラミング応用」を指し、これはあらかじめ教師が科目名とフォルダ名の対応を示しておく。また「s + 学籍番号」フォルダの中に「name.txt」というテキストファイルをメモ帳で作らせ、受講者のフルネームを入力しておく。あとは受講開始後に受講者に教科書のプログラムファイルを入力させ、所定の授業フォルダ内に所定の名前で保存させる。たとえば「prog1 (プログラミング基礎)」の「044-hello.py」というファイルは、教科書44ページの「hello.py」というプログラムを指している。ここでは「s20191xx1」という学生が「prog1」の受講者であり、すでにたくさんのプログラムを入力済みであること、また「prog2 (プログラミング応用)」の受講者でもあり、こちらは授業が始まったばかりでフォルダが空っぽであることを意味している。「s20191xx2」という学生は「prog1」を受講していないためフォルダは無く、「prog2」だけの受講者である。今後、「prog2」の授業(演習、自習)が進むにつれてファイルが増えていく。以上が受講者側の準備である。

一方教師側は固定された教師フォルダ(ここでは「kamichi」)の中に各プログラミング科目に対応するフォルダを用意し、その中に「name.txt」を作成する。中身は科目名とする。たとえば「prog1」の「name.txt」には「プログラミング基礎」という文字データが保存されている。そして各科目フォ

ルダの中には教科書の例示プログラムの完成ファイルを準備しておく。この意味については後述する。教師フォルダは「正答」に相当するデータが置かれるため、受講者との共有を行わない。また受講者同士ではファイル共有をしない。あくまでも教師と受講者の1対1の共有が集まった1対nの関係となる。教師はシステムにおいて受講者データの準備をする必要はなく、正しく事前準備を行った受講者がシステム上に自動的に表れることになる（と同時に、準備が完了していない受講者のフォローが必要となる）。

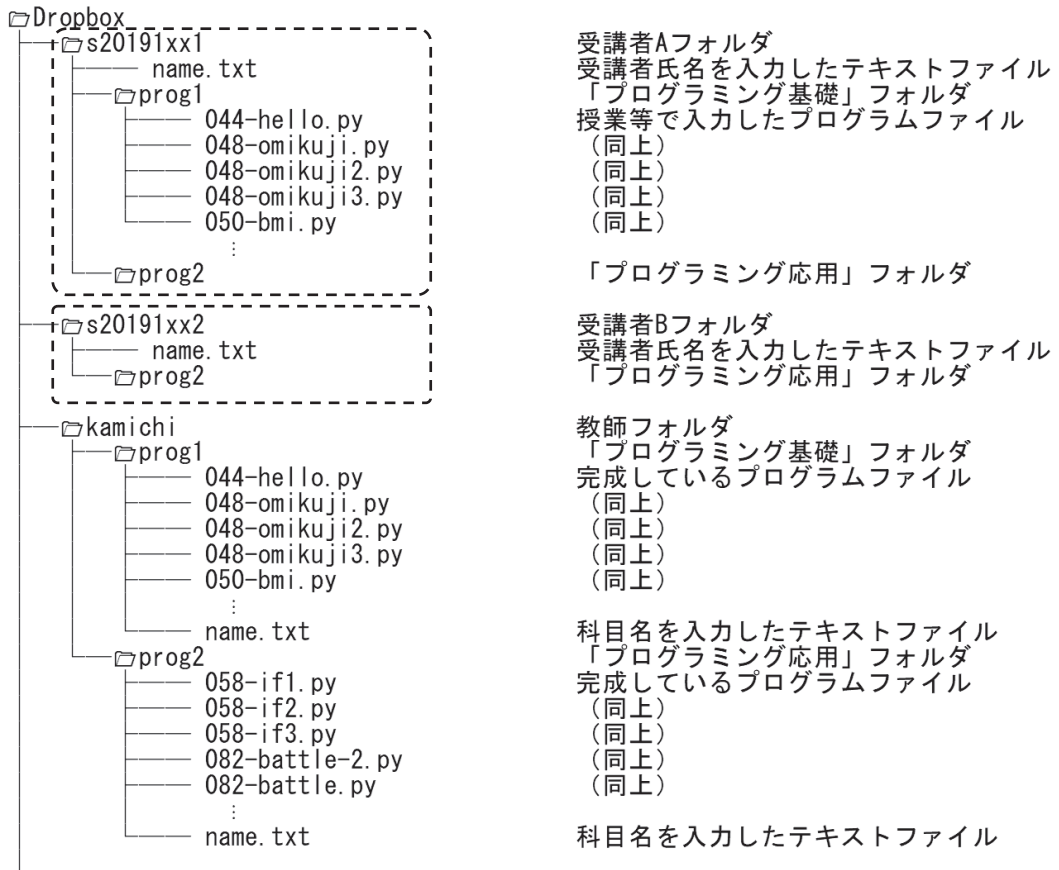


図1 教師側PCのDropbox フォルダ内容

### 2-3 システムの原理

教師側の Dropbox フォルダには準備の完了した全受講者の受講用フォルダが同期される。その中のプログラムファイルは親フォルダの名前により科目名が判別できる。またさらにその親フォルダが受講者の学籍番号を示し、「name.txt」が受講者名を表す。これらのデータを集計することにより、どの受講者がどの科目としてどのプログラムファイルをいつ作成（更新）したかについて把握することができる。また受講者のプログラムファイルと同じ位置関係にある教師フォルダのプログラムファイル（＝完成版）と比較することで、受講者の入力したプログラムのどこにミスがあるか（あるいは完成

か) を比較することが可能となる。以上の状況を自動的に集計し、ブラウザ上で提示することでプログラミング学習の状況を把握することが本システムの原理となる。なお、本システムが稼働する Web サーバ内でも Dropbox クライアントアプリケーションが動いていて、教師および全受講者の Dropbox フォルダが同期されている。

### 3. 各ページの詳細

#### 3-1 ①認証ページ

認証ページ (図 2) は学籍番号・教職員番号とイタズラ防止の確認キーを入力させるだけのページであり、情報を送信すると入力した大学アカウントに「①状況把握ページ」のページ URL が記載されたメールが送られる (図 3)。この URL を開くことでシステムにアクセスできる。URL にはパスワードに相当するトークンが埋め込まれていて、試行中の現在は月単位で変化する設定となっている。仮に URL が外部に漏洩した場合、翌月にはトークンが無効となる。一般的には日単位で変化するべきである。

## Progress 認証ページ

学籍番号を入力してください。リンクを記載したメールが大学Gmail (stアカウント) に送付されます。

学籍番号  (頭のsやtは不要)

確認キー  (sociologyと入力)

図 2 ①認証ページ

## 大東文化大学 社会学部 Progress 受信トレイ x

t059486@st.daito.ac.jp  
To t059486 ▼  
(Progress からの自動送信メールです)

リンクは以下の通りです。

[http://it-kyouzai.com/progress/view.cgi?user=059486&pwd=\[トークン\]](http://it-kyouzai.com/progress/view.cgi?user=059486&pwd=[トークン])

このリンクは今月のみ有効です。

図 3 認証後に届くメール

#### 3-2 ①状況把握ページ

図 4 が状況の全体表示ページである。「最近の更新」、「科目一覧」、「学生一覧」の 3 パートに分かれ

ている。「最近の更新」は登録されている全受講者について演習を行った結果であるプログラムファイルがファイル更新時間の新しい順にリストアップされる。30分以内の更新は赤字でハイライト表示される。学籍番号および学生名をクリックすると、当該受講者に絞った状況の表示となる(図5)。またファイル名をクリックすると当該ファイルに絞った状況の表示となる(図6)。ファイル名の右に付与されている数字は教師ファイル(正答)との差異数である。これが0ということは基本的に教科書のプログラムをそのままミスなく入力したと判断される。表示が「—」の場合は、受講者が独自に作成したファイル(=対応する教師ファイルがない)か、ファイル名の入力ミスである。「確認」リンクをクリックすると後述の「②差分表示ページ」に飛ぶ。

「科目一覧」は登録されている科目が並び、チェックボックスのオン・オフで「最近の更新」の絞り込みを行う。「学生一覧」は特定の受講者に絞りこむ状況表示へのリンクである。なお、「科目一覧」、「学生一覧」は教師として認証した場合にのみ表示される。

最近の更新 (09:21)						
•	12/23 (金) 22:32	20191	李	プログラミング応用	block6.py	--- 確認
•	12/21 (水) 17:47	20191	佐	プログラミング応用	block6.py	--- 確認
•	12/21 (水) 17:43	20191	彦	プログラミング応用	block6.py	--- 確認
•	12/21 (水) 17:43	20191	松	プログラミング応用	block6.py	--- 確認
•	12/21 (水) 17:34	20191	杏	プログラミング応用	177-city3.py	--- 確認
•	12/21 (水) 17:33	20191	杏	プログラミング応用	177-city2.py	--- 確認
•	12/21 (水) 17:33	20191	杏	プログラミング応用	chap5-13.py	--- 確認
•	12/21 (水) 17:33	20191	杏	プログラミング応用	177-city1.py	--- 確認
•	12/21 (水) 17:33	20191	杏	プログラミング応用	chap5-8.py	--- 確認
•	12/21 (水) 17:27	20191	杏	プログラミング応用	block6.py	--- 確認
•	12/21 (水) 16:57	20191	山	プログラミング応用	block6.py	--- 確認
•	12/21 (水) 16:57	20191	成	プログラミング応用	block6.py	--- 確認
•	12/21 (水) 16:56	20191	中	プログラミング応用	block6.py	--- 確認
•	12/21 (水) 16:55	20191	阿	プログラミング応用	block6.py	--- 確認
•	12/21 (水) 16:55	20191	田	プログラミング応用	block6.py	--- 確認
•	12/21 (水) 16:51	20191	中	ネットコンプ	177-city1.py	6 確認
•	12/21 (水) 16:47	20191	平	プログラミング応用	block6.py	--- 確認
•	12/21 (水) 14:42	20191	中	ネットコンプ	177-city1.py	14 確認
•	12/21 (水) 14:41	20191	彦	ネットコンプ	177-city1.py	5 確認
•	12/21 (水) 14:40	19191	佐	ネットコンプ	177-city1.py	5 確認
•	12/21 (水) 14:40	20191	山	ネットコンプ	177-city1.py	5 確認
•	12/21 (水) 14:39	20191	佐	ネットコンプ	177-city1.py	9 確認
•	12/21 (水) 14:38	20191	成	ネットコンプ	177-city1.py	6 確認
•	12/21 (水) 14:38	20191	渡	ネットコンプ	177-city1.py	5 確認

図4 ①状況把握ページ(全体表示)

## 科目一覧

- プログラミング基礎
- プログラミング応用
- ネットコンプ
- 社会学演習 I
- 社会学演習 I (Web)
- 社会学演習 II

## 学生一覧

- |         |   |         |   |         |   |
|---------|---|---------|---|---------|---|
| • 18191 | 大 | • 20191 | 飯 | • 21191 | 大 |
| • 19191 | 中 | • 20191 | 阿 | • 21191 | 磯 |
| • 19191 | 和 | • 20191 | 松 | • 21191 | 坂 |
| • 19191 | 劉 | • 20191 | 平 | • 21191 | 中 |
| • 19191 | 李 | • 20191 | 松 | • 21191 | 高 |
| • 19191 | 白 | • 20191 | 山 | • 21191 | 赤 |
| • 19191 | 飯 | • 20191 | 福 | • 21191 | 成 |
| • 19191 | 佐 | • 20191 | 小 | • 21191 | 白 |
| • 19191 | 清 | • 20191 | 成 | • 21191 | 鈴 |
| • 19191 | 大 | • 20191 | 中 | • 21191 | 保 |
| • 20191 | 渡 | • 20191 | 張 | • 21191 | 館 |
| • 20191 | 田 | • 20191 | 陳 | • 21191 | 伊 |
| • 20191 | 松 | • 20191 | 李 | • 21191 | 桑 |
| • 20191 | 彦 | • 20191 | 小 | • 21191 | 坂 |
| • 20191 | 長 | • 20191 | 岩 | • 21191 | 王 |
| • 20191 | 杏 | • 20191 | 中 | • 21191 | 上 |
| • 20191 | 高 | • 20191 | 佐 |         |   |

Progress

図4 ①状況把握ページ（全体表示）（前ページからの続き）

最新の更新 (09:26) - 20191 中						
• 12/21 (水) 16:51	20191	中	ネットコンプ	177-city1.py	6	確認
• 12/21 (水) 14:22	20191	中	ネットコンプ	chap5-13.py	4	確認
• 12/21 (水) 14:13	20191	中	ネットコンプ	chap5-11.py	2	確認
• 12/21 (水) 13:40	20191	中	ネットコンプ	chap5-8.py	2	確認
• 12/21 (水) 13:20	20191	中	プログラミング応用	block6.py	---	確認
• 12/14 (水) 18:06	20191	中	プログラミング応用	block3.py	---	確認
• 12/14 (水) 18:00	20191	中	プログラミング応用	block5.py	---	確認
• 12/14 (水) 18:00	20191	中	プログラミング応用	block4.py	---	確認
• 12/14 (水) 17:50	20191	中	プログラミング応用	block2.py	---	確認
• 12/14 (水) 17:50	20191	中	プログラミング応用	block1.py	---	確認
• 12/14 (水) 17:04	20191	中	プログラミング応用	battle_turn.py	0	確認
• 12/14 (水) 17:04	20191	中	プログラミング応用	column11.py	0	確認
• 12/14 (水) 17:04	20191	中	プログラミング応用	battle_message.py	0	確認
• 12/14 (水) 17:04	20191	中	プログラミング応用	battle_start.py	0	確認
• 12/14 (水) 14:27	20191	中	ネットコンプ	fr24.py	---	確認
• 12/14 (水) 14:19	20191	中	ネットコンプ	fr24-2.py	---	確認
Progress						

図5 絞り込み表示：受講者

最新の更新 (09:26) - chap5-11.py						
• 12/21 (水) 14:23	20191	山	ネットコンプ	chap5-11.py	7	確認
• 12/21 (水) 14:13	20191	中	ネットコンプ	chap5-11.py	2	確認
• 12/21 (水) 14:01	20191	平	ネットコンプ	chap5-11.py	4	確認
• 12/21 (水) 14:00	20191	中	ネットコンプ	chap5-11.py	5	確認
• 12/21 (水) 13:58	20191	佐	ネットコンプ	chap5-11.py	4	確認
• 12/21 (水) 13:57	20191	長	ネットコンプ	chap5-11.py	2	確認
• 12/21 (水) 13:57	20191	彦	ネットコンプ	chap5-11.py	2	確認
• 12/21 (水) 13:56	19191	佐	ネットコンプ	chap5-11.py	4	確認
• 12/21 (水) 13:56	20191	渡	ネットコンプ	chap5-11.py	4	確認
• 12/21 (水) 13:56	20191	成	ネットコンプ	chap5-11.py	2	確認
• 12/21 (水) 13:56	20191	松	ネットコンプ	chap5-11.py	4	確認
• 12/21 (水) 13:55	20191	松	ネットコンプ	chap5-11.py	3	確認
• 12/21 (水) 13:54	20191	岩	ネットコンプ	chap5-11.py	4	確認
• 12/21 (水) 13:53	20191	松	ネットコンプ	chap5-11.py	4	確認
Progress						

図6 絞り込み表示：ファイル

この状況把握ページにより、受講者の学習状況を一覧できる。加えて、気になった受講者だけの状況や、特定のプログラムファイルに対する受講者の演習状況の確認も容易となっている。ページは1分毎に更新（リロード）される。

### 3-3 ②差分表示ページ

先述の①状況把握ページの「最近の更新」ではファイル名の右に数字（入力ミス数）が表示されるが、その横の「確認」リンクをクリックすると、その入力ミスの内容を確認する②差分表示ページが開かれる。受講者が入力した個々のプログラムファイルと教師が持っている完成ファイルと比較することで、受講者の入力に間違いがある箇所を即座に確認することができる。以下の図7は差分表示の例で、入力ミス数「3」の内容である。矢印の部分が「差異あり」として着色されている。なお、対応する教師ファイルが存在しない場合は、受講者のプログラムファイルの内容だけが表示される。差分表示はPython 言語の標準ライブラリである「difflib」モジュールを活用している。

テキスト	学習者
1 import pandas as pd	1 from ast import Store
2 import folium	2 import pandas as pd
3 df = pd.read_csv("898.csv")	3 import folium
4 store = df[["緯度", "経度", "店舗名(日本語)"]].values	4
5 m = folium.Map(location=[35.942957, 136.198863], zoom_start=16)	5 df = pd.read_csv("898.csv")
6 for data in store:	6
7     folium.Marker([data[0], data[1]], tooltip=data[2]).add_to(m)	7 store = df[["緯度", "経度", "店舗名(日本語)"]].values
8 m.save("store.html")	8
	9 m = folium.Map(location=[35.942957, 136.198863], zoom_start=16)
	10 for data in store:
	11     folium.Marker([data[0], data[1]], tooltip=data[2]).add_to(m)
	12
	13 m.save("store.html")

Progress

図7 差分表示ページの例（入力ミス数：3）

差分表示画面は左右2つを並べる形で表示され、左が教師（完成版）、右が受講者の入力内容である。受講者の1行目に「from ast import Store」という不要な1行（ただし実害はない）が含まれていること、末行の「」記号を「”」記号で入力していることが色付けで指摘される。緑色が「余計なもの」、赤色が「足りないもの」、黄色が「違うもの」を指す。両者の行番号に注目してほしいが、受講者のプログラムは教師のそれより空行が多く入力されている。空行や余計なスペースおよびコメントはプログラムの挙動に影響しないため、教師ファイルと異なっても入力ミスとはカウントしない。（印刷では見えないと思われるが）薄い黄色でプログラムの内容に影響のない差があることを示している。

一方で、先述の「”」と「‘」の違いもプログラム上違いはないが、ここではミスとして指摘してしまっている。このような「ミスではない差異」をカウントさせない機能を用意していて、画面左上の「OK とする」リンクをクリックするとこの入力内容（プログラムファイル）を「正答」とみなすことになる。そうすると①状況把握ページの「最近の更新」ではカッコ付けでミス数が表示される（図8）。教師はカッコ付きの数字は0と等しい（=完成）と考えることができる。

そして「添削」リンクをクリックすると後述の③添削ページに飛ぶ。



最新の更新 (09:54) - chap4-22.py						
•	12/07 (水) 14:29	20191	岩	ネットコンブ	chap4-22.py	2 確認
•	12/07 (水) 14:27	20191	張	ネットコンブ	chap4-22.py	0 確認
•	12/07 (水) 10:00	19191	佐	ネットコンブ	chap4-22.py	5 確認
•	12/07 (水) 03:22	20191	中	ネットコンブ	chap4-22.py	0 確認
•	12/07 (水) 02:09	20191	福	ネットコンブ	chap4-22.py (3)	確認
•	12/07 (水) 00:09	20191	松	ネットコンブ	chap4-22.py	2 確認

図8 ミスではない差異を許容した場合の表示例

### 3-4 ③添削ページ

添削ページは受講者のプログラムに対して描画ツール機能を使って指導内容を手書きすることができる(図9)。色付き細ペンおよび蛍光太マーカーによってホワイトボード上のプログラムに自由に線を描いたり、直線を引いたりすることができる。本システムはブラウザ上で利用するものであるが、パソコンだけでなく iPad のようなタブレットでの使用も想定していて、タブレット(とスタイラスペン)で利用するとこの描画機能を有効に活用できる。

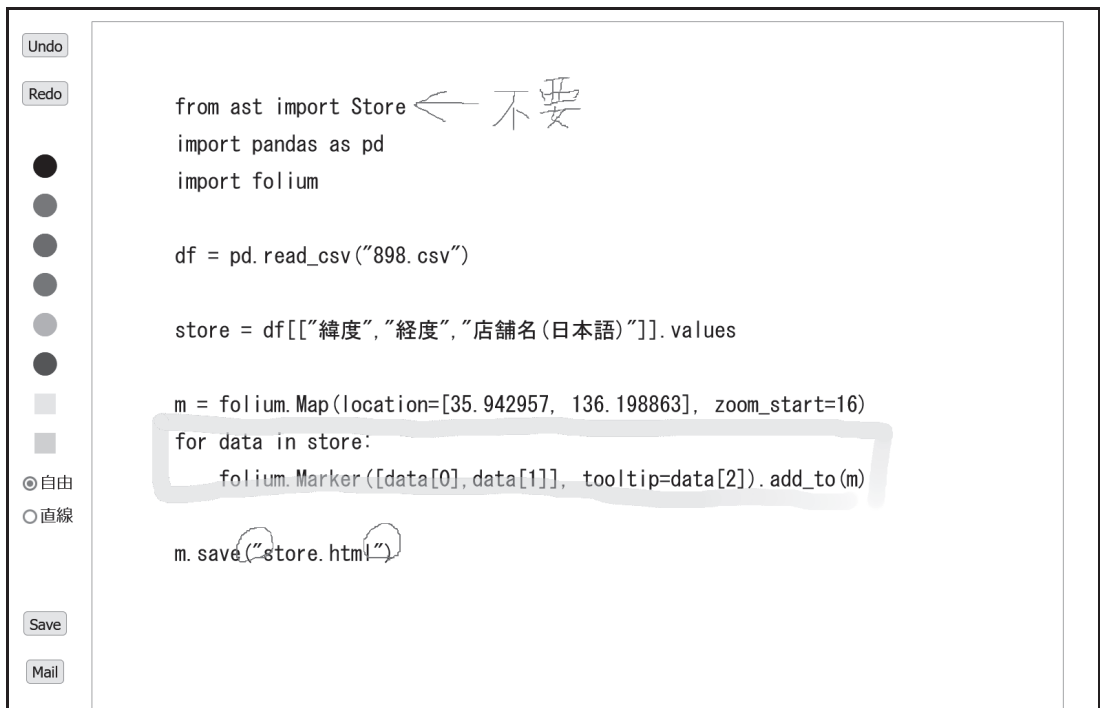


図9 プログラムの手書き添削例

添削した画像は「Save (保存)」ボタンでPC内に保存し、「Mail」ボタンでOSの標準メーラーとして設定されたメールソフトを起動する。宛先や題名(例では「[ネットコンブ]chap22-4.pyの添削」)、

本文（「〇〇さん、プログラムの添削画像を添付します。参考にしてください。」）は定型文が自動入力されるので、「Save」ボタンで保存した画像をメールに張り付けて送信するだけで良い（図 10）。

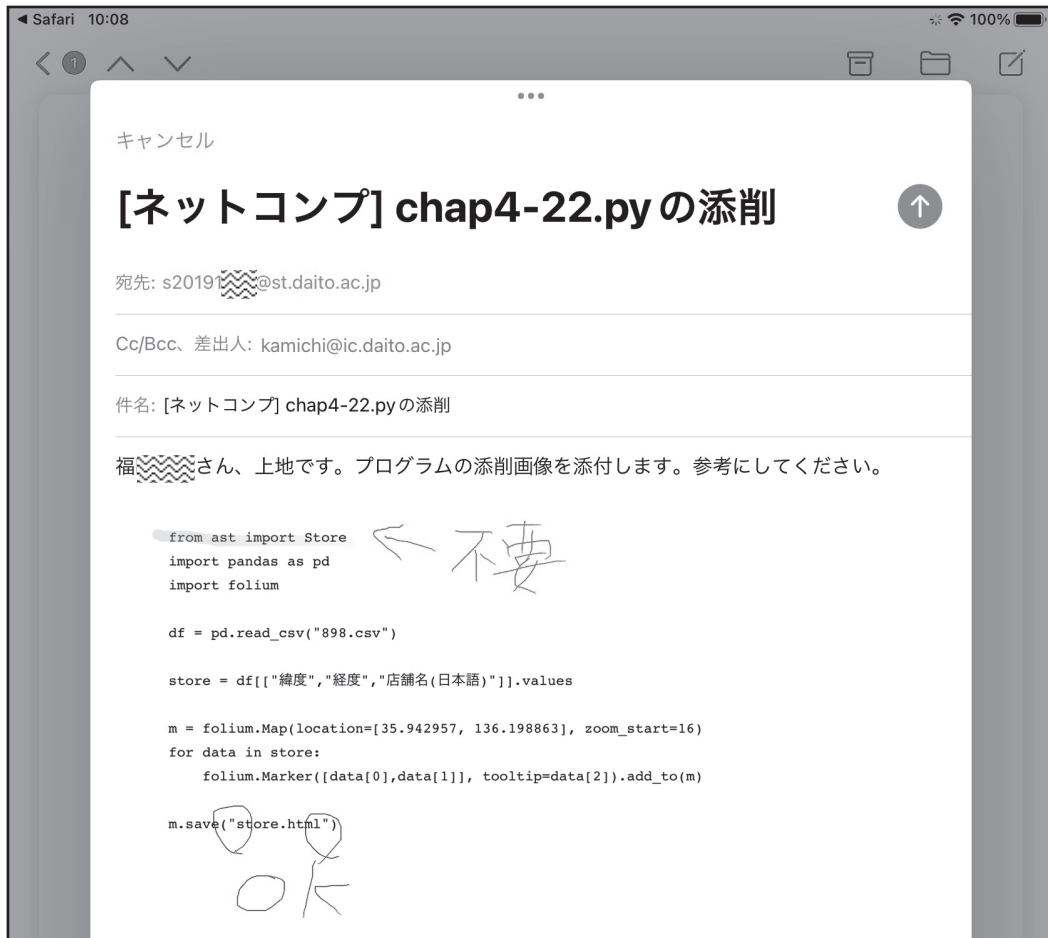


図 10 添削結果のメール送信画面（iOS の標準メールアプリ）

以上述べたように、まず全体の状況を把握し、必要に応じて個々の状況にアクセスする。場合によって添削を行うなどの指導ができるシステムとなっている。

### 3-5 受講者向け状況ページ

本システムは主として教師向けに作られている。受講者は自分が入力したプログラムにミスがあって動かない場合に、何カ所間違っているかの情報は意味がないためである。また教科書のプログラムの場合「最終形（完成版）」が存在するが、自分で作るプログラムには最終形が存在しない（自分が作ったところまでが最終形）。将来のことを考えると「どうして動かないか、何が間違っているか」をエラーメッセージから導く能力が必要になる。本システムはあくまでも教師が個々の受講者のミスを即座に把握することによって指導時間を確保するための補助システムである。

一方で、プログラミングの教科書は後半になるほど入力するプログラムの分量が大きくなり、受講者にとって入力のハードルが高くなる。反転学習のような自習形式の場合に、多くの受講者が頓挫することになる。その段階になった際、本システムの存在を紹介し、受講者向け状況ページを開放する。①認証ページにおいて教師ではない受講者が認証を行った場合に、メールで送られる URL で表示される①状況把握ページは「最近の更新」について受講者本人のファイルだけが並ぶ状態となる。

## 4. 授業形態別の活用

### 4-1 反転学習

プログラミング科目を対面形式の授業形態で行う際、教科書のプログラムをその都度入力させていると授業時間が確保できない。そこで反転学習と組み合わせ、先に教科書のプログラムを入力・実行させる。その際に本システムを活用すると、個々の受講者の状況が容易に把握できる。受講者から質問があった場合に即座に状況を確認し、リモートまたは添削メールによって適切な指導を行える。

### 4-2 授業時演習

授業内でプログラムを入力する場合も①状況把握ページによって各受講者の入力内容と、対応する教師ファイルがある場合に入力ミスの箇所をチェックできる。教室にタブレットを持ち込むとリアルタイムに教室内の受講者の状況を把握できる。従来、通路から奥まった場所の着席者の PC をのぞき込むことが困難であったが、本システムでは本人の PC を見なくても指導が可能となる。

### 4-3 オンライン（オンデマンド）学習

反転学習に準ずる。受講者はある程度まとまった時間で演習を行うため、入力されるプログラムの更新時間を見ることで、どれぐらいの時間を演習に費やしていることが推測できる。想定以上に時間がかかっていると判断した場合は、質問の問い合わせが来る前に教師側からアドバイスや状況の問い合わせを行うなどの指導も可能となる。

## 5. おわりに

以上で述べたとおり、プログラミング教育支援システムの実装が完了した。すでに2年間のシステム評価を行い、課題も見えてきたが全般的に良い感触を得られている。その内容については改めて報告の機会を持ちたい。

### 〈参考文献〉

Dropbox (クラウドファイル共有サービス), <https://www.dropbox.com/>, (参照 2023 年 1 月 1 日).

Python Software Foundation, 「difflib モジュール」, <https://docs.python.org/ja/3/library/difflib.html>, (参照 2023 年 1 月 1 日).

# Development of a Programming Education Support System Using File Sharing Services

KAMICHI, Koichi

Herein, a programming education support system has been developed that enables remote, real-time monitoring of the status of programming exercises and self-study using a file-sharing service.

The developed system checks the status of the program and displays the differences between the program file typed by each student and the completed version. In addition, the system has a drawing function that allows the teacher to write handwritten comments on the program files typed by the students.

This paper presents the details of the developed system.

Key words : Web Application, Programming Education, Class Support System, File Sharing