

# SIST オンラインジャッジシステムのユーザビリティの改善

## Usability Improvements to the SIST online judgement system

國持 良行\*

Yoshiyuki KUNIMUCHI

We prepared a physical hosting sever outside our institute and constructed the online judgment system , which is called SIST Online Judgement (abbreviated to SOJ) , on the server. SOJ has been put into operation since November in 2021 and it has been served for introductory education of programming in these two years. This academic year, it was modified with front end parts to enforce the usability of SOJ. As a result of the questionnaire, most of the students answered affirmatively with the learning effect and the usability of SOJ. Moreover, SOJ contributed to increase the homework time of students.

### 1. はじめに

競技プログラミング等でよく使われるオンラインジャッジシステム(Online Judgement System, 以下 OJS)をプログラミングの導入教育に活用するために, 学外に専用ホスティングサーバを借りて, 2021 年 11 月そこへ OJS システムを構築した(以下, SOJ とよぶ)<sup>1,2)</sup>. そして, 2022 年度には, SOJ のフロントエンド周りを刷新して, 使いやすさ(Usability) の向上を目指した. そして, 両年度とも 1 年後期必修科目「プログラミング演習」において SOJ を使った演習を取り入れた. SOJ の効果を測るために 2 回の実力テストを実施して, SOJ の使用状況や学習効果を検証した. SOJ による演習は, 科目の成績には算入せず, 自発的な学習の取組みとした.

国内外の OJS としては, 会津大学の AOJ<sup>3)</sup>, 北京大学の POJ, AtCoder などがよく知られており, ランキング, コンテストやリクルートなどで活用されている. Moodle のプラグインとしての OJS なども提案されている<sup>4)</sup>が, サーバに負荷をかけ過ぎる不安がある.

また, OJS をプログラミング教育に活用する報告もみられる<sup>6),7)</sup>. 不正コピー防止<sup>8),9)</sup>や作問の効率向上<sup>10)</sup>についての提案もみられる.

本論文では, 文献 11)~13)の内容をもとにして, 2 章では SOJ のフロントエンド(外部仕様)の改良点を, 3 章では専用サーバ上での構築を, 4 章では「プログラミング演習」で運用した結果をそれぞれ紹介する.

### 2. OJS の概要と SOJ の外部設計

この節では, 一般的な OJS の概要を紹介し, それに沿って SOJ の外部仕様を紹介する.

#### 2.1 OJS とは

OJS は, 一般に Web 上のサーバが学習者から提出されたプログラムを自動採点し, その結果を DB へ記録, 学習者へフィードバックするシステムである.

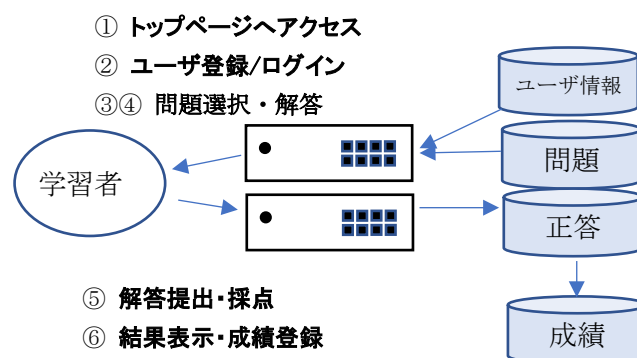


図 2.1 OJS での学習方式

学習者は, ① Web ブラウザでサイトへアクセスする. ②初回のみユーザ登録をする. まず, システムにログインする. ③用意された問題の中から 1 つの問題を選択して, ④プログラムを作成する. 学習者は, プログラムを作成し終わったら, ⑤ そのソースコードを提出欄に記述して提出(submit)する. 提出後, システムは自動採点を行い, ⑥ 学習者へ結果を返して, データベースに問題番号, 正解不正解の状態, 実行時間, 最終日などを登録する.

#### 2.2 SOJ システムの外部設計

この節では, SOJ システムの外部仕様を, Web 画面を紹介しながら説明する. SOJ に携わる人(利用者)は, 学習者, 教師, 管理者の 3 つのカテゴリに分類される. システムには最初ただ一人の管理者 root/admin が存在し, ユ

ーザ登録・削除などの特権が与えられる。学習者は、SOJシステムを使って、学習をする人たちである。教師は、利用可能機能については学習者とほぼ同じであるが、担当する学習者のグループの学習状況を CSV ファイルとして SOJ からダウンロードする機能をもっている点で異なる。

昨年度に挙げられた、外部仕様に関する課題とその対応を以下に示す。○は対応済み，×は未対応を示す。ほぼすべての課題を解決している。

- Web ページのデザインを，CSS や Java スクリプトを使って見やすいものに作り込む。
- 問題選択画面にて問題難易度の情報を載せる。
- 問題を分野別に出題する。
- 問題を問題名で検索する。
- × 問題文と提出欄を同じページに配置する。
- 提出欄をもっと広くする。
- システムが採点中に詳細な経過が分かるようにする。
- 採点が終了したときに採点結果を学習者に提示する。
- 自分の成績表を一覧でき，自分の実力を視覚化する。
- 利用者全体のランキング(上位数名)を表示する。

SOJ の外部仕様(画面)の実装状況を表 2.1 に示す。2022 年度では，使いやすさ(usability)の向上を目指してシステムの改修を行った。表のエントリ順に各画面の機能を述べていく。

表 2.1 SOJ の画面の実装状況

SOJ の外部仕様(画面)	2021 年度版	2022 年度版
①トップ(図 2.2)	◎	○
②ログイン(図 2.3)	◎	○
③問題一覧(図 2.4)	◎	○
④問題(図 2.5)	◎	変更せず
⑤解答提出(図 2.6)	◎	○
⑥提出結果確認(図 2.7)	◎	○
⑦問題作成	◎	変更せず
⑧問題削除	◎	変更せず
⑨Q&A(図 2.9)	×	◎
⑩質問投稿(図 2.10)	×	◎
⑪ランキング(図 2.11)	×	◎
⑫マイページ(図 2.12)	◎	○

◎は新規追加，○は修正，×は未実装。

①トップ画面

SOJ の URL は，以下の通りである。ただし，現在 Basic 認証(ID と PW)を設定している。

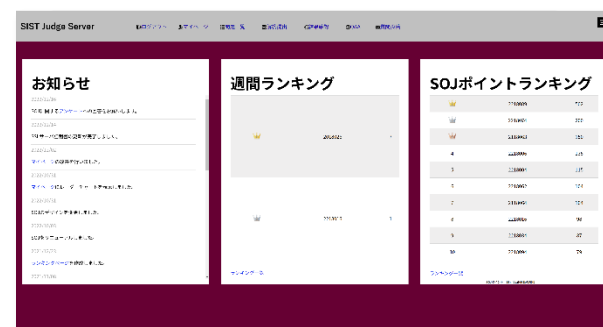
<https://sist-onlinejudge.com>

なお，このサイトは，さくらインターネットからのホスティング専用サーバ上に構築されており，IPv4 アドレスも 1 つ付与されている。ログイン認証後，SOJ のトップ画面が表示される(図 2.2)。(a)は 2021 年度版，(b)は 2022 年度版の画面である。追加した項目は，画面上部の SOJ のロ

ゴ，トップメニューである。画面中央には，週間と累積のランキングを表示した。CSS のデザインをスクールカラー(えんじ色)に変更した。



(a) 2021 年度版



(b) 2022 年度版

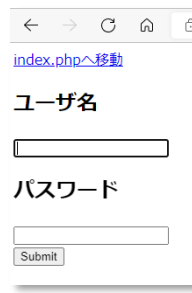
図 2.2 ①SOJ のトップ画面

② ユーザ登録画面

最初に SOJ を利用するには，ユーザ登録をしなくてはならない。トップメニューの「ログイン」をクリックし，「新規登録はこちら」を選択する。ユーザ登録画面はデザインのみ変更したので詳細については省略する。

② ログイン画面

トップメニューから「ログイン」を選択する。設定した「ユーザ名」と「パスワード」を入力して，SOJ にログインする。



(a) 2021 年度版



(b) 2022 年度版

図 2.3 ②ログイン画面の変更

### ③ 問題一覧画面

問題一覧ページ(図 2.4(b))へ移動後、問題を選択して、問題の解答に入る。問題番号と問題文、サンプル入出力例がいくつか記述されている(図 2.5)。学習者はエディタやコンパイラを使ってプログラムを作成する。

**昨年度からの変更点**：問題検索機能、ページへのリンクを追加した。問題一覧で1ページ当たりの表示問題数の上限を付けた。各問題の分野と難易度を表示した。学習者から良問と判断された問題には「いいね」を押してもらうこととした。

問題ID	問題名	提出状況
1	HelloWorld	正解
2	AxB	正解
5	XX	未正解
6	testtett	未正解
7	POW	正解
8	Q0052ITLCM	正解
10	Q0053ARBackOrder	正解
11	Q0054_IT_SUM	正解
13	Q0051_OP_divisor	正解
14	Q0058_OP_RemainderMult	正解
15	Q0059_IT_Factorial	正解
16	Q0057_IT_CenRem	正解
19	Q0019_CD_NaturalNumberComparison	正解
21	Q0021_IT_SquareDrawing	正解

(a) 2021 年度版 問題一覧画面

問題ID	問題名	提出状況	分野	難易度	いいね
1	HelloWorld	正解	初級	1	👍
2	AxB	正解	四則演算	1	👍
5	XX	正解	文字列	2	👍
6	testtett	正解	探検	1	👍
7	POW	正解	四則演算	1	👍
8	Q0052ITLCM	正解	繰り返し処理	4	👍
10	Q0053ARBackOrder	正解	数列	5	👍
11	Q0054_IT_SUM	正解	繰り返し処理	2	👍
13	Q0051_OP_divisor	正解	条件分岐	1	👍
14	Q0058_OP_RemainderMult	正解	四則演算	2	👍

(b) 2022 年度版 問題一覧画面

図 2.4 ③問題一覧画面の変更

### ④ 問題画面

この画面は、2021 年度版と 2022 年度版はほぼ同じである。学習者は、この問題文および入出力例を理解して、プログラムを作成する。まず、問題文とその直後に制約条件が記述されている。制約条件は問題の難易度(実行時間や記憶サイズ)を決定する重要なファクタである。そして、2,3 個の典型的な入出力例が記述される。問題を理解する助けとなる。必要に応じて補足説明を入れる。

問題作成については、3 年生前期の必修科目「コンピュータシステム実践演習 1」にて、3 年生に問題を作成してもらった。しかしながら、同じ問題が複数作られたり、問題文に不備があったり、サンプル入力ファイルや正答ファ

イルに誤りが頻発した。

### 問題

二つの整数a,bが空白区切りで与えられるので、a\*bの値を改行込みで出力してください。

### 制約

1 ≤ a,b ≤ 10000

### 入出力例

#### 入力1

5 6

#### 出力1

30

#### 入力2

318 2942

#### 出力2

1818156

図 2.5 ④SOJ の問題画面

【現状の課題】プログラミングコンテストでは**制限時間**が設けられていることが多いが、本システムでは学習を対象としているので、プログラムの作成に制限時間を設けていない。また、個人成績一覧から正解していない問題を選ぶのが一般的である。学習者のレベルに応じた問題や苦手分野の問題などをシステムが AI や DS を利用して**自動出題**を可能としたい。現在 C/C++言語のプログラムのみを採点できるが、将来的には**他言語への拡張**(Python や Java など)を図りたい。

### ⑤ 解答提出

問題 ID と問題名を選択し、作成したプログラムをコピーして、貼り付ける(図 2.6(b))。そして、「Submit(問題提出)」ボタンを押すと採点が始まり、⑥解答結果確認画面へ遷移する。

**昨年度からの変更点**：解答欄のサイズを大きくして、見通しをよくした。

### Input SourceCode

```

1 HelloWorld
#include<stdio.h>
int main()
{
    puts("Hello World");
    return(0);
}
    
```

図 2.6 (a) 2021 年度版 解答提出画面



(b) 2022 年度版

図 2.6 ⑤SOJ の解答提出画面

【現状の課題】 また、自動採点には他人のプログラムの **コピー&ペースト** が問題となる。コピペ対策を論じている先行研究もある。我々は、字句解析ツール(flex など)を使って **キーワードの出現分布** をコピペ発見に活用することを考えている。

⑥ 提出結果確認画面

SOJ は解答(プログラム)を採点して、以下の結果を判定する。採点に要する時間は約 1 分である。これは 10 個のテスト入力に対して、余裕をもたせて 1 個当り 5~6 秒程度の計算時間を設けてあるからである(図 2.7)。

- ・ 正解(Accepted, AC または Correct Answer, CA)
- ・ 不正解(Wrong Answer, WA)
- ・ コンパイルエラー(Compile Error, CE)
- ・ 実行時エラー(Runtime Error, RE)
- ・ 制限時間超過(Time Limit Exceed, TLE)
- ・ 記憶制限超過(Memory Limit Exceed, MLE)

本年度の提出結果確認画面(図 2.7(b))では、問題 ID と判定結果、実行時間とメモリ使用量、提出日時、ソースコードへのリンクが表示される。この表は 1 ページで表示されるのでスクロールバーを使って表中を移動する。なお、表の上部には、問題番号などをキーとした検索ができる機能を追加した。

**昨年度からの変更点:** 昨年まではブラウザの更新ボタンを押して、採点結果が出るのを待っていた。本年度は採点していることが分かるように Java スクリプトで採点中アニメーションを表示することにした<sup>1)</sup>。残り秒数も表示される(図 2.8)。

問題ID	最終結果	実行時間	メモリ使用量	
1	AC	0	0	<a href="#">view</a>
2	AC	0	0	<a href="#">view</a>
5	WA	0	0	<a href="#">view</a>
5	WA	0	0	<a href="#">view</a>
1	WA	0	0	<a href="#">view</a>
1	AC	0	0	<a href="#">view</a>
1	AC	0	0	<a href="#">view</a>
8	AC	0	0	<a href="#">view</a>
11	AC	0	0	<a href="#">view</a>
13	AC	0	0	<a href="#">view</a>
7	AC	0	0	<a href="#">view</a>

(a) 2021 年度版 提出結果確認画面

問題ID	最終結果	実行時間	メモリ使用量	提出日時	ソースコード
42	AC	0	0	2023/2/24 11:14:30	<a href="#">view</a>
42	AC	0	0	2022/12/27 14:29:26	<a href="#">view</a>
42	AC	0	0	2022/12/27 12:39:28	<a href="#">view</a>
42	AC	0	0	2022/12/24 15:08:59	<a href="#">view</a>
355	AC	0	0	2022/12/21 17:42:27	<a href="#">view</a>
354	AC	0	0	2022/12/20 19:03:55	<a href="#">view</a>
354	WA	0	0	2022/12/20 18:49:57	<a href="#">view</a>
353	AC	0	0	2022/12/20 18:46:35	<a href="#">view</a>
352	AC	0	0	2022/12/20 18:35:10	<a href="#">view</a>

(b) 2022 年度版 提出結果確認画面

図 2.7 ⑥提出結果確認画面

【現状の課題】 TLE や MLE は試作版では実装していないが、time.ps コマンドなどを使って今後実装を図りたい。

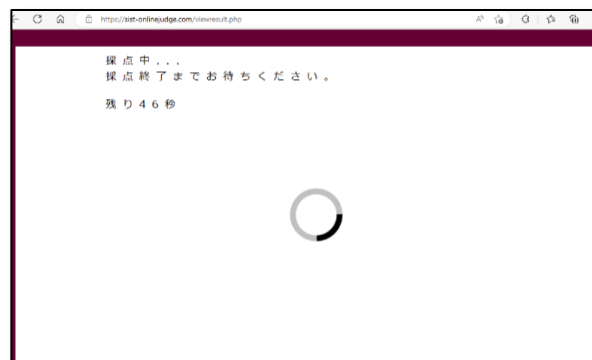


図 2.8 採点待ちアニメーション

⑦ 問題作成画面

⑧ 問題削除画面

⑦と⑧の画面については昨年度からの変更はないので、詳細は昨年度の紀要<sup>2)</sup>を参照されたい。

⑨ Q&A 画面(新規追加)

この機能は、本年度に追加したものである(図 2.9)<sup>12)</sup>。学習者から、「模範解答を公開してほしい」「ヒントを与えて欲しい」などの意見が多かった。しかし、模範解答を公開してしまうと、模範解答をそのまま提出され、SOJ の評価が学習者の能力を正しく反映しない懸念が心配された。そこで、模範解答を公開する代わりに、学習者からの質問を受付けて、質問に回答する方式で、ヒントを与えることにした。そして、質疑応答をスレッド状に記録して、ヒント集としての利用を目的とする。



図 2.9 ⑨ Q&A 画面

⑩ 質問投稿画面(新規追加)

この機能は、本年度に追加したものである<sup>12)</sup>。学習者が問題に対する質問を、このページから投稿する。簡易の SNS システムである。件名、内容、ソースコードが入力できる(図 2.10)。SOJ プロジェクトのメンバーがこの質問に回答する。質問内容は、問題文や入出力例における誤りの指摘、正答ファイルの内容の誤りの指摘、などが多かった。

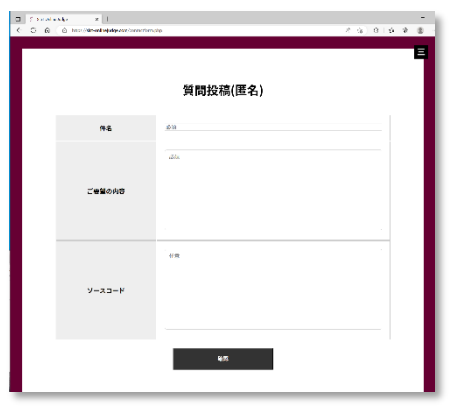


図 2.10 ⑩ 質問投稿画面(新規追加)

⑪ ランキング画面

この機能は、本年度に追加したものである(図 2.11)<sup>13)</sup>。全学習者の成績を正解数順に並べた表である。自分の順位

を全体の中で俯瞰して、解答を促すものである。



図 2.11 ⑪ ランキング画面

⑫ マイページ画面

ここではマイページについて述べる<sup>13)</sup>。SOJ のマイページでは自身の学習状況を視覚的に確認できるように提出数や正解数などのサマリ表(図 2.12)、難易度ごと、分野ごとに正解数を表示したレーダーチャートを設けた(図 2.13)。

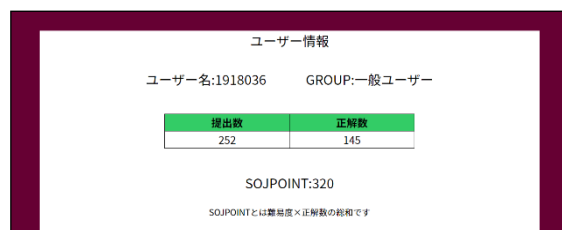


図 2.12 ⑫ マイページ画面(サマリ表)

Javascript の chart.js を用いて難易度、分野ごとの正解数についてのレーダーチャートを作成した。

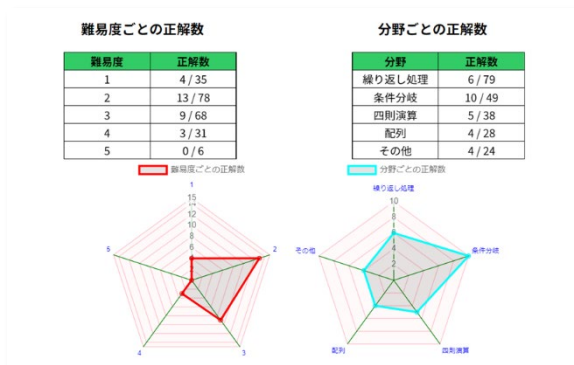


図 2.13 ⑫ マイページ画面(レーダーチャート)

### 2.3.2 教師モードと管理者モードの追加

ここでは SOJ での教師モードについて述べる。SOJ の教師モードでは担当教授が生徒の学習状況を確認できるように成績を csv 形式にて出力できるようにしたものである。担当する各学生が、各問題について AC, WA, CE, TLE などの状態が表示される。教師のマイページから、〇〇教授担当\_成績 csv ダウンロードボタンを押すことで成績をダウンロードすることができる。教師モードの作成にはユーザ管理テーブルの auth (権限番号) を使用した。この番号が、1 教師モード、2 一般ユーザ、3 國持担当クラス、4 大相担当クラスの生徒に割り振った。

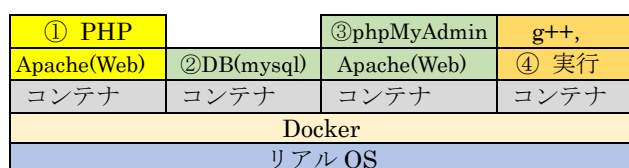
なお、教師モードは未完成であり、機能の実装は今後の課題である。

## 3. システム概要

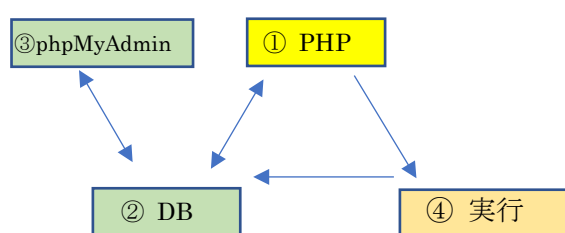
ここでは SOJ のシステムの概要を述べる。

### 3.1 システム概要

本 SOJ システムは、主に ① インターフェースを担当する PHP コンテナ、② ユーザ情報、管理情報、問題情報を管理する DB コンテナ、③ DB とのインターフェースを担う phpMyAdmin コンテナ、④ プログラムを起動し、正誤を判定する実行コンテナからなる(図 3.1)。



(a) システム動作環境



(b) システムの処理の流れ

図 3.1 システム概要図

①～④は Docker の仮想環境で動作するコンテナとして実現されている。コンテナは、仮想のマシン/OS 環境であり、特定の環境下でソフトウェアを動作させることを可能にする。したがって、1 台のリアルマシンによって、複数台のマシン上で動くソフトウェアが協調するシステムを構成することができる。

① PHP コンテナは、Web サーバ上で動作し、利用者の対応(front end)を担当する。ユーザからは、プロフィール

設定(登録・削除・変更・プロフィール確認)や操作要求(ログイン・ログアウト・問題選択・採点・成績表示・履歴おいて表示)などの要求を処理する。

② DB コンテナは、ユーザ情報、管理情報、問題情報などを管理する。ユーザ情報はプロフィール、提出履歴、成績などが含まれる。管理情報には、クラス編成、コンテストの情報などがある。問題情報には、問題集とその各問題に対するテストデータが含まれる。問題には学習分野と難易度を付けたい。問題は、学習者(ユーザ)に理解できる言語(日本語や英語)で書かれ、解答されるプログラム言語に依存しないことが望ましい。

③ 実行コンテナは、学習者(ユーザ)が解答(プログラム)提出時に選択した言語処理系へ渡して、実行結果と正答データと比較して、正誤を判定する。現在選択できる言語処理系は C/C++コンパイラのみであるが、順次言語処理系を増やしていく予定である。

正答データは、テスト入力(ファイル)とそれに対する正しい出力が格納されたファイルの複数の集まりである。すべてのテスト入力に対して、提出プログラムが正しい結果を出力した場合に、正解と判定する。テスト入力に対して制限時間内に実行が終わらなければ、その実行を Docker コンテナ(内の OS)が強制的に打ち切り、「制限時間オーバー」(TLE)を判定する。また、実行中にエラー(例外など)が発生したときにも、強制的にプロセスを停止し、「実行時エラー」(RE)と判定する。これらの結果は DB の提出履歴と成績に反映される。

悪意のあるプログラムが入力される恐れがあるので実行コンテナの実行権限、ファイルアクセス権限は最低に設定する。

### 3.2 システムの実行環境

SOJ のシステム構成は、2023/3/1 現在以下の通りである。

【ハードウェア・OS】

モデル名:RX1330 M3 4 コア 1CPU

CPU: E3-1220 v6 3GHz (4Core)

メモリ: 8GB

ストレージ 1: SSD 480GB x2

OS: Ubuntu 20.04

【ソフトウェア】

Docker(プロセス管理): 20.10.8

Apache( Web サーバ): 7.2

PHP(インターフェース): 7.2.34

MySQL(データベース) : 8.0.26 - MySQL Community Server - GPL

GCC(コンパイラ): 8.3.0

### 3.3 バックエンドについての現状

今年度は、フロントエンドの改善を目標に開発を進めた。

バックエンドは2021年度からはメンテナンス程度の変更しかない。

### Docker とは

**Docker** とは、コンテナ仮想化によってアプリケーションを開発・実行するためのオープンソースソフトウェアである。この機能を用いることによって、簡単かつ高速にコンテナを立ち上げることができ、コンテナを分散することによって、意図せぬ動作による不具合の影響も軽微に抑えることが可能になる。

**Docker-compose** とは、1つのアプリケーションを複数のコンテナとして組み合わせることで定義するファイルである。これを用いて、PHPコンテナ、DBコンテナ、実行コンテナを合わせて1つのSOJアプリケーションとして構築されている。なお、1つのコンテナは、1台の仮想的な計算機とみなすことができるので、SOJのコンフィギュレーションを変更する場合には、それぞれのコンテナに設定する必要がある。例えば、毎年11月にはSSLサーバ証明書の更新を行う必要があり、それぞれのコンテナにおいて、証明書を入れ替えなくてはならない。

### 実行コンテナ

PHPコンテナに提出されたプログラムは、まず実行コンテナに送られる。実行コンテナに送られた後は、拡張コマンドtask-spoolerなどを用いてキューに登録し、実行用のプロセスが空き次第実行に移す。採点の実行後は、採点の結果をデータベースに格納しプロセスを終了する。

現在は実行用のコンテナが1つであり直列での処理になっているが、ゆくゆくは3~5個程度の実行コンテナを用いて並列処理を行う予定である。コンテナの状況を見て割り振りを行う必要があり、課題である。

### データベースコンテナ

データベースのテーブルのうち主なものは、ユーザ管理用テーブル、問題管理用テーブル、提出結果管理用テーブルの3つである。利用者の使用状況を把握するには、提出結果管理用テーブルを参照する(表3.1)。4章でデータ分析するために、利用者の使用状況データを、phpMyAdminを用いてエクスポートした。このテーブルに格納されているエントリは6607件であり、これは総提出件数に一致する。データベース内では限られた列に情報を圧縮しているため、エクスポートされたCSVファイルから、VBAマクロを分析しやすいようにデータを抽出する作業が必要であった。

表 3.1 提出結果管理用テーブル

列名	データ型	説明
Id	Int	[主キー]提出結果のid
User	Int	提出ユーザのid
Problem	Int	提出問題のid
Lang	Int	言語
Status	varchar(16)	実行結果
Time	Int	実行時間
Memory	Int	使用メモリ
Timestamp	Date	提出日時
Path	varchar(256)	正答の格納パス

大量の提出によるデータベースの容量不足が危惧されるため、今後はuserごとに提出結果管理用テーブルを作成するなど、データベースの再構築が課題となっている。

## 4. 試験運用とその結果

SOJの運用状況と運用結果を述べる。

### 4.1 外部サーバへの移植過程

2021/8/25に、さくらインターネットの専用サーバPHYを1台レンタルする契約を結んだ。まず、さくらインターネットの「コントロールパネル」ページから専用サーバへ、以下のOSをインストールし、rootやネットワークの設定をした。

**OS: ubuntu20.04**

**IPv4 アドレス: 153.126.242.45**

そして、専用サーバでの設定・開発・運用を安全かつ高速に行うために、SSH通信経由でコマンド実行している。そのためSSHのユーザ設定などを行った。

次に、Dockerをインストールして、仮想サーバに切り分けた。その後仮想サーバへApache、PHP、MySQL、GCCなどのソフトウェアをインストールした。Dockerを用いた利点は、将来システムを拡張する際に、処理を並列化して大量のトランザクションに対応可能となることである。

9~10月にはDNS名の取得、サーバ証明書の取得、SSL通信を実施した。

**サーバURL: <https://sist-onlinejudge.com>**

なお、セキュリティのため、Basic認証がかけられている。

2021/11/23のプログラミング演習の第10回目の授業にて、1年生へSOJの紹介を行った。なお、Sisterネットワークからは、本学のインターネットの間にあるファイアウォールによって、いくつかのポートが閉鎖されていて、SisterネットワークからSOJサーバへアクセスができない。これについては、SOJへアクセスする場合にはeduroamネットワークへ接続することで対応した。

2022年度8月までに、「プログラミング演習」の範囲(初歩から配列・関数まで)の問題231題を用意した。PDFの問題集を作成し、ilearn上に公開した。また、ホスティング専用サーバと同じシステム構成のクローンを学内のローカルサーバ上にも構築し、開発のテスト体制を整えた。次節では、2021年度と2022年度の後期に実施したSOJの運用状況と運用結果を報告する。

### 4.2 運用状況

情報学部1年生後期必修「プログラミング演習」(火曜日2時限目、対面実施、2021年度は国持クラスの履修者数23名、2022年度は国持クラスと大相クラスの履修者51名)において、SOJを運用した。この科目は定期試験の得点で評価することになっているので、「SOJの成績、授業内の演習、実力試験の成績は、科目の成績評価には使わない」とあらかじめ伝えておいた。

2021年度は、10回目の授業に当たる2021/11/23(火)から

最終回まで、2022年度は第2回目から最終回まで演習に利用した。それぞれの初回にSOJの目的・操作手順・成績閲覧の方法を紹介し、問題一覧から問題を選択して、演習時間と家庭学習にて解答するように指示をした。途中で2回の実力試験を実施した。

### 4.3 SOJの利用状況

2021年度後期と2022年度前後期の利用状況について報告する。2021年度後期は「プログラミング演習」(1クラス)、2022年度前期は「コンピュータシステム実践演習1」、2022年度後期は「プログラミング演習」(2クラス)にてSOJを利用した。まず、総提出件数は6607件であった。そのうち採点結果

の内訳は、

正解(AC)の数：3102 (47.0%)

不正解(WA)の数：2417 (36.6%)

コンパイルエラー(CE)の数：1031 (15.6%)

制限時間エラー(TLE)の数：57 (0.8%)

実行時エラー(RTE)の数：0 (0%)

であった。あらかじめ自分のPCで実行できることを確かめてから、コピー&ペーストで提出するので、CEは1割程度にとどまった。それを除くと、正答と誤答はほぼ同数ということになる。TLEのほとんどは無限ループに陥るものと推測される。ゼロ除算のようなRTEが検出されなかったのは意外であったが、エラー通知を取り逃して検出ができていない可能性もある。

次にSOJによる家庭学習の時間が増えているかどうかを調べる。そのため、日別時間帯別、曜日別に集計してみた。SOJの家庭学習の時間を増やす目的については、効果があった。

図4.1は、プログラムを提出した回数を月別に集計したものである。2022年度後期は「プログラミング演習」(2クラス)で利用があったため、全体の2/3の提出数を占めた。休業中、とくに春休みの利用は極端にすくないが、定期的に使っている学習者もみられた。

提出件数(件)

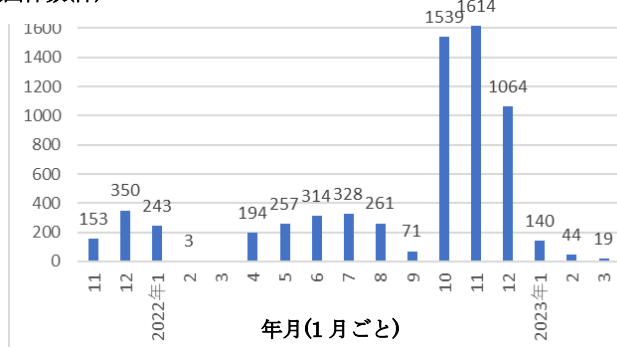


図 4.1 月ごとの提出件数

次に、図4.2は、1時間単位の時間帯別の提出回数を集

計したものである。授業時間帯の利用が多いことが分かる。また、深夜の時間帯に学習をしている人が予想外に多かった。夕方から21までの利用率が低いのは、バイトなどで忙しいためではないだろうか。SOJは家庭学習を促進する一定の効果があった。

提出件数(件)

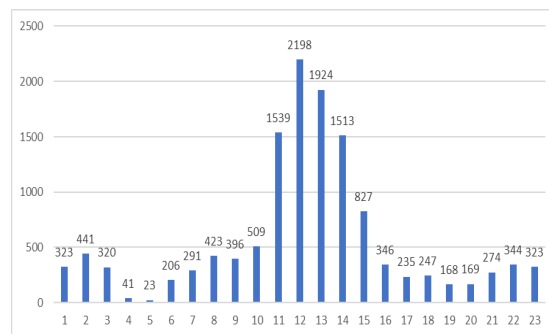


図 4.2 時間帯別の提出件数

図4.3は、曜日ごとの提出件数である。プログラミング演習の開講日である火曜日の利用が多い。週末の利用は少なかった。

提出件数(件)

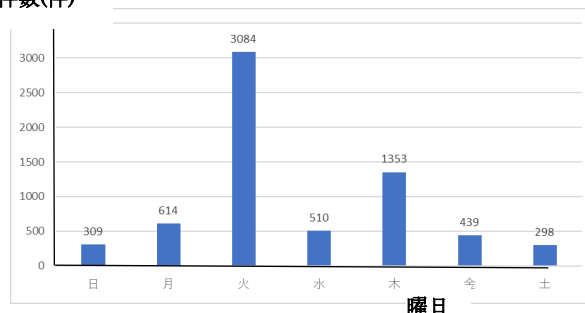


図 4.3 曜日ごとの提出件数

### 4.2 SOJと実力試験の関係

2022年度プログラミング演習では、第1回(2クラス受講者51名、受験者47名、欠席者4名)と第2回(2クラス受講者51名、受験者44名、欠席者7名)の実力試験を、第9回(11/22)と第14回目(12/27)の授業中にそれぞれ実施した。それぞれ100点満点で、プログラミング能力を測定する。なお、SOJや実力試験の成績は、科目の成績には算入しないことを履修者に周知した。両テストの平均点は以下の通りである。

表 4.1 実力試験の平均点の比較

	1回目 (11/22)	2回目 (12/27)
平均点(満点 100)	66.3 点	62.4 点

最初以下のような予想がなりたつと考えて、データ分析に望んだ。



予想:「実力試験の得点と SOJ の AC 数に間には正の強い相関がある」

2022 年度の 1 回目の実力試験の得点と SOJ の AC 数の相関を求めてみた(表 4.2)。いずれも弱い正の相関があるとの結果であった。上記の予想は成り立たない。実際に、定期試験における相関係数も同じような傾向がみられた。

表 4.2 実力試験の得点と SOJ の AC 数の相関係数

	1 回目 (11/22)	2 回目 (12/27)
相関係数	0.38	0.34

そこで、次に、2 回の実力テストの得点と SOJ の AC 数の分布を分析することにした(図 4.4)。丸いマーク(●)は 1 回目の、四角いマーク(■)は 2 回目の実力テストのそれぞれ得点を表している。どうも上記の相関係数が予想ほど上がらない原因は、SOJ をほとんど利用しなかった割に高得点をとった人(図中の領域 C)、および高得点ではあるがそれをはるかに超えて利用した人(領域 B)がいることが大きな要因である。これは、昨年度も似た傾向が見られた。

その結果、この 2 カ年を分析してみて、学生のタイプを大まかに A~D の 4 つに分類できるのではないかと感じている。

**Aタイプ**: 日常の学習に SOJ を利用し、実力アップにつながった。これは我々が予想していた学習者層である。

**Bタイプ**: SOJ にはまってしまい、高い実力を身につけた人たち。専門性に強く、将来研究開発に向いていると思われる。このタイプの学生の能力を伸ばすと興味深いことが起きることが期待できる。

**Cタイプ**: SOJ に頼らなくても、実力が身についた。努力よりは要領よく勉強する傾向が見られるのではないだろうか。今回は、SOJ の結果は成績に反映しないので、あえて SOJ を利用しなかったとも考えられる。

**Dタイプ**: AOJ を利用せず、実力アップしなかった。SOJ の AC 数が 0 近辺の人たちは、普段の学習習慣が身につけていないと思われる。

得点(点)

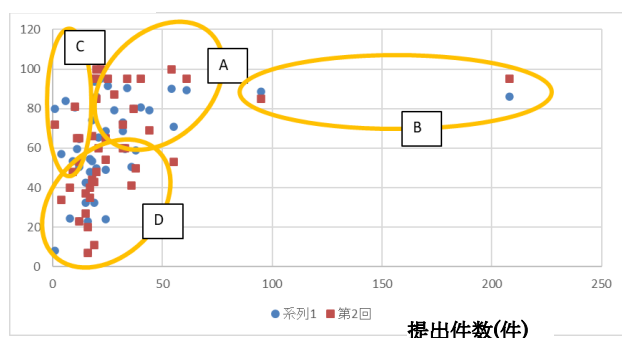


図 4.4 SOJ の AC 数と実力テストの点数の関係

SOJ とプログラミング能力の関連性を調査するに当たっては、以下のことが課題である。

- 不正行為への対策が必要である。他人の正解をコピー&ペーストして、提出する。
- 問題の質を上げ、学習範囲を網羅するのに十分な問題量を用意する。
- 実力テストの測定精度を上げる。

### 4.3 使い勝手に関するアンケート調査

このアンケートの結果は、文献 11)で詳しく述べられている。ここでは要約のみを述べる。2022 年度「プログラミング演習」履修者に依頼し、12/10~1/17 の期間に 27 件の回答を得ることができた。アンケートの内容は以下のとおりである。(1), (4)~(5)は 5 段階評価(1:いいえ, 5:はい)である。

#### 質問項目

- (1) SOJ はプログラミング学習に役立ったか
- (2) プログラミング学習に役立ったと考える理由
- (3) ランキング機能はモチベーション向上に繋がったと感じるか
- (4) SOJ の使いやすさ (UX) は向上したと感じるか
- (5) SOJ のデザイン (UI) は見やすいと感じるか
- (6) 追加してほしい機能について
- (7) 使いづらいと感じたページや、見づらいと感じたページについて

質問(1)の平均得点は 3.85 であり、次の質問の回答をよく集約している。質問(2)では、肯定的な意見(理解や達成感)が多数であるが、否定的な意見(困ったときの支援がない)も少し見られた。SOJ は学習の助けになったと考える。質問(3)の平均得点は 3.07 であり、今回のランキング機能はモチベーション向上には効果が薄かった。質問(4)(5)の平均得点は、各々 3.78, 3.89 であり、今回のフロントエンドの改善は初期の目的を達成した。より一層の改良を計画したい。質問(6)は、昨年度と同様に模範解答やヒントが欲しいという意見が多数であった。問題文にヒントを入れることも考えたい。質問(7)の回答については本年度中に改良できるところは対処した。

## 5. あとがき

プログラミングの導入教育に活用するために、学外に専用サーバ上で稼働している SOJ に対して、主にフロントエンドの改修を行った。そして、2022 年度 1 年後期必修科目「プログラミング演習」において運用した。また、SOJ の効果を測るために 2 回の実力テストを実施した。家庭学習時間や学習者の理解度においては寄与したが、プログラミング能力との関連については十分な検証ができなかった。今後の取り組みについては、

(1) 記憶効率や処理効率を向上させるために、データベースを再設計し、蓄積されたデータの再構成を行う。

(2) 対応するプログラミング言語の種類(Python や JavaScript を優先する)を増やすこと、不正コピー&ペーストの検出と防止、採点過程の表示、計算時間や記憶容量の測定などバックエンド側の機能を実現させる。

(3) プログラミング能力の実力診断、問題の自動選択など AI を用いた機能を付加する。

(4) 「コンピュータシステム実践演習 1」にて SOJ のコンテンツ、とくに問題の作成を促進する。良問をたくさん作成することが SOJ の学習効果を上げる一番の課題である。

(5) ①情報学部内部でプログラミングコンテストを実施して DP 到達度試験の 1 つの指標として、②副専攻の受講生に対する C 言語の副教材として、③入学生確保のための新しい入試制度の学力評価指標として、それぞれ活用を検討する。学習者向けの学習支援については、動画教材をパワーポイントで作成し、予復習に活用する。SOJ に掲載された問題を PDF 問題集として配布したい。

**謝辞** 本研究は 2021 年度および 2022 年度に静岡理科大学「提案型教育研究 教育推進プロジェクト」の支援を受けた。関係各位に感謝の意を表します。

## 参考文献

- 山下颯也, 國持良行. "プログラミング導入教育を目標とした SIST オンラインジャッジシステムの試作と提案," 静岡理科大学紀要, 29, pp.109-117 (2021-08-31)
- 山下颯也, 高田洗, 國持良行. "SIST オンラインジャッジシステムの授業における運用報告," 静岡理科大学紀要, 30, pp.129-136 (2022-08-31)
- 渡部有隆. "オンラインジャッジの開発と運用-Aizu Online Judge," 情報処理, 56, 10, pp. 998-1005 (2015)
- 松永賢次. "導入プログラミング教育におけるオンラインジャッジシステムの活用の試み," 情報科学研究, 31, pp. 25-41 (2011)
- 古谷勇樹, 林 真史, 山本 隆弘, 長尾 和彦. "RK-003 オンラインジャッジシステムと連携可能な Moodle プラグインの実装と比較 (K 分野: 教育工学・福祉工学・マルチメディア応用, 査読付き論文)," 情報科学技術フォーラム講演論文集, 14,3, pp.89-94 (2015).
- 長尾 和彦, 古谷 勇樹, 峯脇 さやか. "オンラインジャッジシステムのプログラミング演習への導入と評価," 第 78 回全国大会講演論文集, 1, pp.537-538 (2016).
- 古谷勇樹, 林真史, 山本隆弘, 長尾和彦, "オンラインジャッジシステムを用いたプログラミング学習環境の構築と比較," 教育システム情報学会 2014 年度学生研究発表会
- 松本彩花, 松原 南美, 渡邊遥輔, 多田 拓, 倉光 君郎, "Sumomo: ブロックチェーンを用いた教育用オンラインジャッジの提案," 情報教育シンポジウム論文集 2019 (2019): 321-325.
- 岩本舞, 中村真人, 小島俊輔, 中嶋卓雄, "不正コピー検出手法を備えたオンラインジャッジシステムの開発," 情報処理学会論文誌教育とコンピュータ (TCE) 1.4 (2015): 38-47
- 関根遼, 伊藤恵, 奥野拓. "数学文章題を利用したオンラインジャッジシステム向け問題自動生成手法の提案," 教育システム情報学会 2019 年度第 5 回研究会, (2020).
- 池田智哉, "SOJ システムの開発と運用—UI/UX 向上を目的としたフロントエンドについて—", 2022 年度静岡理科大学卒業研究論文, (2022).
- 尾上将輝, " SOJ システムの運用と開発—学習状況の可視化について—", 2022 年度静岡理科大学卒業研究論文, (2022).
- 寺田晃陽, " SOJ システムの開発と運用—学習の支援と効果の分析について—", 2022 年度静岡理科大学卒業研究論文, (2022).