

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

Constructing and Comparing Programming Learning Environments Which Use Online Judge System

古谷 勇樹, 林 真史, 山本 隆弘, 長尾 和彦
 Yuuki FURUTANI, Hayashi MASAHUMI, Takahiro YAMAMOTO, Kazuhiko NAGAO
 弓削商船高等専門学校
 National Institute of Technology, Yuge College
 Email: nagao@yuge.ac.jp

あらまし：オンラインジャッジシステム(OJS)とはプログラム問題の自動採点を行うシステムである。OJSを使用することで教師が行う採点の効率化を行う。また、本校の e-Learning システムである moodle とシームレスな接続が可能な OJS の実装・比較を行い、特徴、活用方法を報告する。

キーワード：オンラインジャッジシステム、moodle、Java

1. はじめに

近年、ICT の普及により、プログラマの不足が深刻化している⁽¹⁾。そのため、プログラミング学習が重要視され、中学校での必修化などプログラマを育成する取り組みを行っている⁽²⁾。

プログラミングスキルの習得には多くのコードを書くことが重要であり、授業時間外での自学自習が必要になる。しかし、課題の採点は教師の大きな負担となっている。

そこで、問題の自動採点を行うオンラインジャッジシステム(OJS)を利用し、採点の効率化を行う。本研究ではさまざまな課題形式に対応するために、各課題に対応した OJS の実装・試用を行い、特徴と活用方法について報告する。

2. 本校の学習環境と問題点

本校の情報工学科では IT 技術者を育成するために、Java 言語をメインとしたプログラミング学習を行っている。プログラミングスキルを習得させるために、プログラミング課題の実施を行っている。しかし、課題の採点は教師が手動で行っており、大きな負担となっている。また、課題の提出には FTP を使用しており、セキュリティの都合上、学外からの提出を無効にしていたため、学生は自宅での自学自習ができないという問題があった。

本校では、テストファーストを用いた学習をすることで、プログラムの仕様を明確に理解した上で開発を行っている。テストケースの作成にはスキルが必要なため、授業中に教師と一緒に作成している。

3. moodle による学外からの課題提出

学外からの課題提出を可能にするために、本校の e-Learning システムである moodle⁽³⁾を利用する。moodle はセキュリティ対策の施された HTTPS を用いた通信が行えるため、FTP より安全な課題提出が可能である。

moodle には学生の学習活動を履歴として記録す

ることで、教師は学生の学習活動を容易に把握することができる。また、既存のシステムを利用するため、ユーザ登録などの設定をする必要がない。

4. オンラインジャッジシステム (OJS)

OJS はプログラムの自動採点を行うシステムである。自動採点の手順を図 1 に示す。3 章で記述したように、課題提出に moodle を利用するため、OJS を moodle のプラグインとすることで、シームレスな接続にする。上記の条件を満たす 3 種類の OJS を本校では利用した。

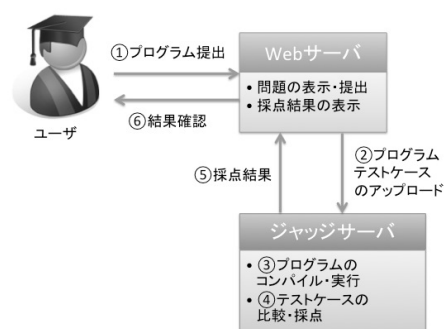


図 1 自動採点の仕組み

4.1 Online Judge Plugin for Moodle (OJPM)⁽⁴⁾

プログラムファイルを提出することで採点を行う OJS。ジャッジサーバは外部のサーバである Ideone⁽⁵⁾を利用した。

4.2 upchecker⁽⁶⁾

ジャッジサーバを作成することで、プログラム課題やレポートの類似度チェックなど、授業特有の採点が可能な OJS。本校ではプログラムファイルの自動採点と、問題とテストのファイルが入ったプロジェクトの自動採点を行う 2 種類のジャッジサーバを作成した。

4.3 CodeRunner⁽⁷⁾

Web ページ上で解答したコードからファイルを作成し、自動採点を行う OJS。ジャッジサーバ Jobe⁽⁸⁾を学校のサーバに構築して利用した。

5. 各 OJS の比較

各 OJS を以下の項目で比較を行った。

- ① OJS の実装による負担
- ② 問題作成・解答提出・採点時間
- ③ 機能

5.1 OJS の実装による負担

各 OJS を moodle へ実装するための作業を表 1 に示す。OJS の実装にあたり、web サーバである moodle に各 OJS のプラグインを組み込む。ジャッジサーバの実装は各 OJS で異なり、実装作業に差がでた。

表 1 OJS の実装行程

| | OJPM | upchecker | CodeRunner |
|---------|--------------------|------------|------------|
| moodle | moodleプラグインのインストール | | |
| ジャッジサーバ | Ideoneのアカウントを作成 | フルスクラッチで作成 | サーバにインストール |
| 実装の手間 | ○ | × | △ |

5.2 問題作成・解答提出・採点時間

問題作成、解答提出におけるブラウザ操作回数を測定することで、各作業の手間を求めた(表 2、3)。また、解答提出から結果表示までの時間を計測した(表 4)。対象とした問題を図 2 に示す。

- 問題名 : FizzBuzz
- メソッドFizzBuzzの引数の値が
 - ・ 3の倍数の場合はFizz
 - ・ 5の倍数の場合はBuzz
 - ・ 3と5両方の倍数の場合にはFizzBuzz
 - ・ それ以外はなにも出力しない
- テストケース数 : 4つ

図 2 問題内容

表 2 問題作成の手間

| | OJPM | upchecker | CodeRunner |
|---------|------|-----------|------------|
| クリック数 | 20+n | 13 | 14 |
| キー入力箇所 | 4+2n | 9 | 6+2n |
| ページ遷移回数 | 6 | 8 | 8 |

n=テストケースの数

表 3 解答提出の手間

| | OJPM | upchecker | CodeRunner |
|---------|-----------------|---------------|--------------------|
| クリック | 6 | 11 | 4 |
| キー入力 | クラス名変更 (1カ所) | パス入力 (1カ所) | コピー & ペースト (1回) |
| ページ遷移回数 | 3 | 6 | 3 |

表 4 採点時間 (平均)

| | OJPM | upchecker | CodeRunner |
|---------|---------|-----------|------------|
| 採点時間(秒) | 104.027 | 5.646 | 3.267 |

測定回数 各10回

5.3 OJS の機能比較

各 OJS が提供する機能を表 5 にまとめる。テストファーストの項目は Junit プログラムの採点が可能かどうかで判断する。CodeRunner はジャッジサーバの仕様を変更することで、Junit プログラムの採点が可能であるため△とする。

表 5 OJS の機能

| | OJPM | upchecker | CodeRunner |
|----------|------|-----------|------------|
| テストファースト | × | ○ | △ |
| セキュリティ | ○ | × | ○ |
| 実行結果の表示 | ○ | ○ | ○ |

6. 考察

各 OJS の特徴、活用方法をまとめる。

OJPM : 外部サーバである Ideone を利用することで環境構築を容易に行うことができる。サーバ構築ができない環境に向くが、外部サーバを利用しているため、レスポンスが悪く実用的でない。

upchecker : ジャッジサーバをフルスクラッチで作成するため、環境構築に非常に時間を費やす反面、様々な解答形式に対応することができる。活用方法はテストファースト問題の自動採点などがある。

CodeRunner : ジャッジサーバをインストールすることで構築が行えるため、比較的容易に環境構築が行える。web 上でコーディングを行うため、IDE のない環境でも解答が可能である。このことから、プログラミング課題を実施している学習環境での利用に向いている。

7. まとめ

プログラミング課題における採点の効率化を目指すために、OJS を本校の学習環境に導入した。OJS は本校の LMS サーバである moodle のプラグインを利用することで、本校の学習環境とシームレスな接続を実現した。また、各 OJS の試験運用、比較を行った結果、専用の OJS と運用が可能な CodeRunner が適していることがわかった。今後は LMS 用に問題を充実させていく予定である。

参考文献

- (1) Business Journal : IT 技術者不足
http://biz-journal.jp/2014/06/post_5239.html
- (2) 日経ビジネス : 中学校でのプログラミング学習必修化
<http://business.nikkeibp.co.jp/article/report/20141001/271963/?rt=ocnt>
- (3) moodle <https://moodle.org/>
- (4) Online Judge Plugin for Moodle
<https://github.com/hit-moodle/onlinejudge/wiki>
- (5) Ideone <https://ideone.com/>
- (6) 伊藤恵 美馬義亮 大西昭夫 “コース管理システムと授業固有の課題チェック機能の web サービスによる連携”
- (7) CodeRunner <https://github.com/trampgeek/CodeRunner>
- (8) Jobe <https://github.com/trampgeek/job>