

E-08

プログラミング中級者までを対象とした課題自動評価システム Automatic Evaluation System of Coding Assignments for Low-Middle level Students

中口 孝雄†
Takao Nakaguchi

1. はじめに

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導入が試みられている。オンラインジャッジシステムは、課題に沿ったプログラムを利用者が作成して提出し、システムが自動で評価を行うシステムである。提出されたソースコードを評価する教員の手間を削減するだけではなく、正誤判定をシステムに任せて教員が指導に集中することで、教育の質が向上することが期待されている。

オンラインジャッジシステムでは、出題された問題に沿ったプログラムが提出されると、システム内でコンパイル、実行し、出力結果が予め用意されている回答と照合される。出題から提出までに要した時間や出力の正しさ、実行時間などが評価される。TopCoder^[1]や AtCoder^[2]に代表される、いかに速く問題を読み解き、いかに効率的なプログラムを記述できるかを競うコンテストサービスで利用されている。

松永^[3]は導入段階でのプログラミング教育にオンラインジャッジシステムを用いる試みを行っている。従来教員が行っていた、プログラムの正しさのチェックやテストデータのチェック、コードの保守性やスタイルのチェックの一部を自動化するため、オンラインジャッジシステムが持つ出力の自動判定機能を利用している。猪狩ら^[4]は入門者にとってプログラミング環境構築の手間やエラーの原因調査が困難であることが学習のハードルとなっていることを指摘し、Web ブラウザのみで利用できデバッグ機能を備えたオンラインジャッジシステムを提案している。

本稿では、自動採点の範囲を拡大するため、中級者向けの講義に対して適用する際の課題を論じ、実際の講義に適用し、結果を考察する。

2. 中級者を対象とした自動評価システムの課題

従来のプログラム課題へのオンラインジャッジシステムの適用では、提出されたプログラムをコンパイル・実行した際の出力(標準出力)を通じて正しさが検証されていた。中級者向けの講義においてもこの手法が有効であるかを検証するため、実際に著者が担当した中級者向けの Java 言語の講義における課題を対象に、何を検証することで課題の採点を行っているかを調査した。結果を表 1 に示す。

講義中に出した課題の数は全部で 22 個であるが、課題毎に複数の評価を行うため、合計は 22 より大きくなる。講義前半では標準出力で判定を行う課題が大半であったが、中盤以降ファイル入出力や HTTP(HyperText Transfer Protocol)の応答といった、標準以外の出力を問う課題が出されていた。また、特定の関数の実装や、オブジェクト指向の概念を理解させるためにクラスの階層関係に条件を設けた課題も出された。中級者向けの講義では、こうしたプログラムの外部入出力要件やクラス内部の要件も考慮して、適切なコーディングが行われているかを判定する必要がある。

また、標準以外の入出力が対象となるため、ファイルシステムに対する影響や、ネットワークに対する影響も考慮する必要がある。プログラムの学習中にループ終了条件を間違えて無限ループを書いてしまうことは往々にしてあるが、このループ内でファイル出力を行って行けばディスク容量不足からシステムの不安定化を招く可能性があり、ネットワークアクセスを行って行けば意図せず他組織の Web サイトを攻撃してしまう可能性がある。

つまり、中級者を対象とした自動評価システムでは、

- i. 多様な評価基準への対応
 - ii. 実行環境の保護
- に対して考慮する必要がある。

3. システム概要

本稿で提案する自動評価システムでは、まず前章で挙げた、i.多様な評価基準への対応を行うため、階層的な評価環境構築を行う。中級者向けのプログラミング課題の提出物は、階層関係を持った複数のクラスから構成されるものがあり、ファイルも複数に分かれる。また、ソースコードを提出する場合もあれば、Web アプリケーションの様に特定のパッケージング(Java の場合 war 形式)で提出する場合も存在する。そのため、課題によって提出物をコンパイルして実行する場合もあれば、Web アプリケーションサーバへ配備して評価する場合もあり、課題によって評価システムの構成を変更できる必要がある。これを満たすため、実行環境の選択と評価対象の変換、実行、評価を階層的に表現する方式を採用する。表 2 に、階層を構成するモジュール

表 1 プログラム課題の評価対象

評価対象	課題の数	課題例
標準出力	17	カンマ区切りで入力された 2 つの数値の加減乗除を計算して出力する
ファイル入出力	2	指定されたファイルを行毎に読み取り、特定の文字列を含まない行を出力する
関数の振る舞い	4	平面上の点を表す 2 つの Point クラスのインスタンスから距離を返す calcDistance メソッドを実装する
HTTP 要求/応答	5	POST メソッドで送信された認証情報が正しければ welcome ページを、正しくなければ応答コード 403 を返すサーブレットを実装する
クラスの階層関係	2	Figure クラスを継承して Rectangle クラスを定義する

† 京都情報大学院大学, The Kyoto College of Graduate Studies for Informatics.

表2 評価を構成するモジュールの例

モジュール名	対象	動作内容
展開	zip ファイル	zip ファイルを一時ディレクトリに展開する. zip 形式でなければ失敗
ファイル存在確認	ディレクトリ	指定されたファイルが存在しなければ失敗
コンパイル	ディレクトリ	たディレクトリ内のファイル群がコンパイルできなければ失敗
実行	ディレクトリ	指定されたクラスを実行し戻り値が0でなければ失敗
1行入力	入力	1行読み込みなければ失敗
テキストマッチ	文字列	文字列が指定された正規表現にマッチしなければ失敗
配備	war ファイル	与えられた war ファイルをウェブアプリケーションとして配備できなければ失敗
HTTP 要求	URL	HTTP アクセスできなければ失敗
応答コード	HTTP 応答	応答コードが指定外の値なら失敗
HTML 切り出し	HTTP 応答	HTTP の body 部から指定のタグ又は属性値が切り出せなければ失敗

の例を示す。モジュールが実行や評価の対象とするのはファイルやディレクトリ、文字列などであり、モジュールによっては処理の結果新たな処理対象を生成するものもある。その場合、他のモジュールを子要素に持つことができ、処理対象を子要素に渡す事により、階層的な処理を行う。これにより、例えば展開モジュールは提出された zip ファイルを展開し、子要素のコンパイルモジュールがクラスファイルを生成し、最終的にテキストマッチモジュールが出力結果を検証するという構成が可能である。

図1に、処理の構成例を示す。図1a)は、zip ファイルとして提出されたソースファイル群をコンパイル・実行し、出力を1行ずつ検証する構成の例である。図1b)は、ウェブアプリケーションを評価する構成である。実際の講義の課題では、これらを用いて、指定の2つの数値を四則演算して出力する課題や、ユーザ登録、ログイン、ユーザ毎のTODOリスト管理を行うウェブアプリケーションの課題の自動評価が行われた。評価システムはいつでも利用可能なように Web アプリケーションとして実装し、学生が任意のタイミングで利用できるようにした。

前章の ii, 実行環境の保護に対する解決策としては、コンテナ型仮想環境への対応を行った。具体的には、実行モジュールと配備モジュールにおいて、Docker を用いたコン

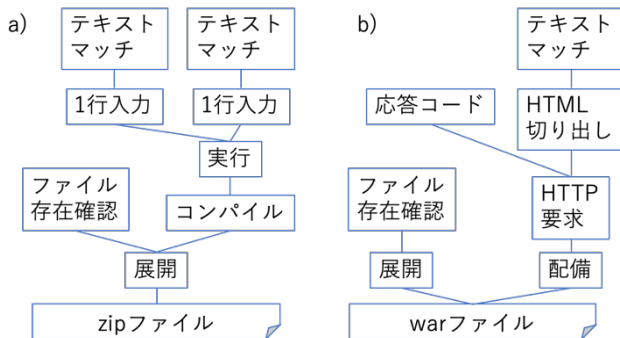


図1 評価モジュールの構成例

表3 アンケート結果概要

課題採点システムを利用しましたか?	
何度も利用した	19
ある程度利用した	7
どちらでもない	3
あまり利用しなかった	6
全く利用しなかった	0
課題採点システムは学習の役に立ちましたか?	
非常に役に立った	9
ある程度役に立った	16
どちらでもない	4
あまり役に立たなかった	5
全く役に立たなかった	1

テナを起動し、コンテナ内でプログラムの実行やウェブアプリケーションの配備を行うことにより、提出プログラムが不正な振る舞いを行った場合でもその影響がコンテナ内にとどまるようにした。

4. 講義への適用と評価

提案したシステムを実際の講義における提出課題の評価に利用し、受講者 50 人に対してアンケートを実施して、35 人から回答を得た。結果の概要を表3に示す。アンケートには自由記述も設けており、良かった点として、“教員の対応を待つことなく正誤を確認できる”、“全学生に対して同じ基準で評価されていることが明らかで公平である”、が挙げられた。改善点として、“エラー時にどこが間違っているかがわかりにくい”、“正解のはずなのにエラーとなる”、が挙げられた。改善点が挙げられた原因としては、課題の位置づけを自然言語で記述された内容に沿ってプログラムを作成する能力を測るもの、とし、採点システムの出力に頼って学生がプログラムを修正することを防ぐために詳細なエラー内容を表示させなかったことが考えられる。実際講義中個別にフォローアップした際には、同様の点を訴えた学生のほとんどが、問題文と自身のプログラムを詳細に突き合わせることで間違いを発見していた。中級者でも採点システムが提供されると、本来課題説明から得べき情報をシステムの出力から得ようとする傾向があり、今後利用を続けていく上で改善が必要な点だと考える。

5. さいごに

プログラミング中級者までを対象とした自動評価システムを提案、実装し、実際のプログラミング言語の講義における提出課題の評価に利用した。その結果、提案した方式で全ての課題が自動評価できることを確認し、受講した学生へのアンケート結果で一定の利便性を確認した。今後は対応するプログラミング言語や技術を拡大すると共に、教育的効果についても検証していく。

参考文献

- [1] TopCoder: <https://www.topcoder.com/> (2019/7/25 参照).
- [2] AtCoder: <https://atcoder.jp/> (2019/7/25 参照).
- [3] 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, 情報科学教育, No. 31 (2010).
- [4] 猪狩 知也, 速水 治夫: プログラミング入門者向けのオンラインジャッジシステムによるプログラミング学習支援, DICOMO(2014).