E-02

プログラミング初学者のためのリアルタイムテストシステムの試作

Prototyping of Realtime Test System for Programming Beginner

福本 大介†　　市川 嘉裕†　　山口 智浩†
Fukumoto Daisuke　Ichikawa Yoshihiro　Yamaguchi Tomohiro

1. はじめに

本学の情報工学科では，本科 2 年生からプログラミング授業を行っている．この授業では，プログラミング知識を身に着ける座学と，実際に課題ベースでプログラミングを行う実習で構成されている．課題を提出する方式では，学生がプログラムを実行しない，または，十分な入力パターンで動作確認せずにソースコードを提出する場合が存在する．これにより，初学者が自主的に論理エラーに対処しようとしなくなることが問題となる．プログラミングは習うより慣れろであるが，上記のような状況ではプログラミング経験を通じて学ぶべき要素の一部が抜け落ちてしまうことが重大な問題である．

本研究の目的は，初学者である学生が論理エラーを含むプログラムを提出することを防ぎ，プログラミング能力の向上を促進するシステムを提案することである．目的の達成のために，プログラムをコーディング時にリアルタイムで実行することで，実行時の入力の手間の削減と素早いフィードバックを実現するシステムを試作した．本稿では，提案システムの要件と試作の概要を述べたのち，初学者によるテストから得られた課題について述べる．

2. 本研究の位置づけ

論理エラーを含むプログラムの提出を防ぐ方法にオンラインジャッジの利用が挙げられる．Web システムを用いてソースコード等を提出させることで，サーバ上で実行したテスト結果をすぐに返すことができる．プログラムの評価がすぐに明示されるため，プログラムを修正して再提出するサイクルが短くなり，結果として正しいプログラムの提出が促進される．プログラミング学習へオンラインジャッジを適用する研究として，稗方が開発した Web アプリケーション JavaEP[1]は，ブラウザ上で取り組む問題の選択，コーディング，実行，採点を行うことができる．これによって，初学者が指導者によらず自主的に学習するシステムを構築することに成功している．一方で入出力パターンは指導者によって用意されるため，学習者はプログラムが完成したかの判断をシステムに依存してしまう．つまり，学習者のテストをする能力が養われないことが問題となる．本研究では，学習者自身によるテストの実施を促すことでテスト能力の向上させることを考える．

初学者の論理エラーへの対処を促すシステムとして，片桐らが開発したプログラミング支援ツール neko[2]がある．neko は，問題に対するプログラムの仕様を学習者自身が整理する機能と，整理した仕様から単体テストコードを自動生成し，入出力が正しいかを確認する機能がある．neko を使用した実験では，被験者のコーディングの正確性が向上したと報告されている．一方で，neko はテストの対象がメソッドやクラスであり，それらの知識を持つ初学者が対象である．本研究は，標準入力を使ったテストを想定し，プログラミングを始めたばかりの初学者も対象にすることを考える．

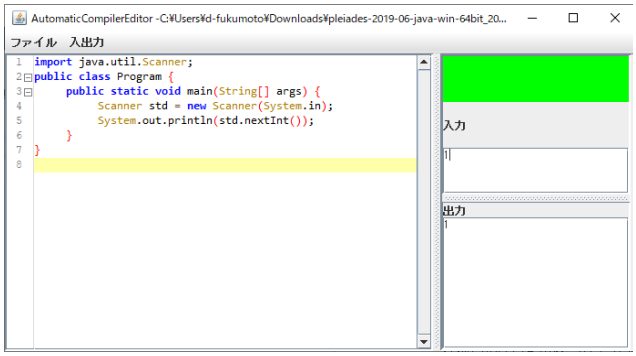

図 1　ACE の外観

3. 提案手法

正しいプログラムの作成には，コーディング，テスト，デバッグを繰り返し行い，想定される入力に対して，正しい出力がされることを確認する必要がある．特に論理エラーを防ぐには，テストで適切な入力パターンを用意することと，それらを用いて動作確認をすることが必要である．そのために学習者は，プログラムを保存，実行，入力を繰り返す必要がある．また，複数の入力パターンをテストする場合には，パターン毎に実行，入力を手動で行う必要がある．プログラムをテストする際のこれらの手間を初学者が面倒だと感じ，省略した場合，テストが不十分なプログラムが提出されることになると考えられる．

以上を踏まえて現状の問題点を解決するために，本稿では，学習者が利用するコーディングツールが備えるべき 2 つの機能を提案する．

（1）　自動的にプログラムを実行する
（2）　複数の入力パターンを並列に実行する

機能（1）により，プログラムが記述された際に自動的に実行されれば，実行せずにプログラムを提出することを防ぐことができると考える．機能（2）によって複数の入力パターンについて同時に実行されれば，一度入力パターンを用意するだけで，それ以降の入力する手間を大きく減らすことができる．

4. 試作システムの概要

提案手法からテスト支援ツール ACE を試作した．ACE の外観を図 1 に示す．図 1 は ACE で Program.java を編集しているときの外観である．Program.java は入力された整数を出力するプログラムである．ACE は Java の GUI の開発フレームである Swing を用いて実装している．図 1 のように左半分のエディタ部と右半分の入出力部で構成されている．エディタ部はユーザが実際にプログラミングを行うテキストエリアである．入出力部は縦に 2 分割されており，

†奈良工業高等専門学校，National Institute of Nara College

図 2　入出力部の追加（４つ）

上半分が入力部，下半分が出力部である．入力部のテキストエリアにプログラムの入力を記述する．出力部はプログラムの実行結果が表示される．図 1 では 1 を入力しているため出力部に 1 と表示されている．

まず，前章で述べた機能（1）の実装について述べる．プログラムの自動実行は 2 つのタイミングで行われる．1 つ目はエディタ部でソースコードを編集したときである．エディタ部にキー入力があった後，1 秒間キー入力がなかった場合，プログラムが自動で実行される．2 つ目は，入力部を編集したときである．入力部のテキストエリアにキー入力があった後，1 秒間キー入力がなかった場合，プログラムが自動で実行される．また自動実行の処理は，プログラムの保存，コンパイル，実行，入力の順で行われる．コンパイル時にエラーが発生した場合，実行はされず，出力の表示も変化しない．実行時のエラーは出力部にエラーメッセージが表示される．

次に，機能（2）の実装について述べる．ACE ではメニューバーの「入出力」から入出力部を追加することができる．また，不要な入出力部は各出力部の下に配置されたバツボタンで削除することができる．図 2 に入出力部が 4 つある場合の実行結果を示す．入力部にそれぞれ 1，2，3，4 と入力しているため，出力部にそれぞれ 1，2，3，4 と表示されている．入出力部が複数ある場合，ソースコード編集時は全ての入出力について自動実行されるが，入力部の編集時は，その入力だけが再実行されるようにした．機能（2）を利用しやすくするため，入出力部は起動時にデフォルトで 2 つ表示されている．

5. プログラミング学習へ適用

本科 2 年生の授業で ACE を使用してもらい， ACE の感想についてアンケートを行ったところ，以下のようなことが分かった．

「授業で Eclipse と ACE のどちらか一方を使用するならばどちらが良いか」という質問に対しては，全ての受講者が Eclipse と回答した．回答に対する理由には「間違えている場所がわかるから」や「使い慣れているから」という意見があった．前者の意見は Eclipse の構文エラーを視覚的に表示する機能に対するものである．ACE では構文エラーの場合，出力が変化しないため，受講者が構文エラーに気づかずにプログラミングを進めてしまうケースがあった．よって，受講者に ACE を使用してもらうには，まず，構文エラーの明示が必要であることが分かった．例えば，Eclipse を参考とし，エラー箇所をソースコード上で指摘し，具体的なエラー内容を日本語で表示する機能が有効であると考える．後者の意見には，Eclipse を授業で約半年間使用しているという背景がある．この意見から，実際の開発環境の導入は年度当初となることから，初学者に対する導入がスムーズになるためのシンプルさとユーザビリティの向上が重要であると考える．

また，ACE の想定していない使われ方として，複数の入力パターンをテストする場合に，入力部がデフォルトで 2 つあるにもかかわらず，片方の入力部に入力パターンを打ち込む学生が存在した．この行動の原因として，受講者のテスト方法の固定化が原因だと推察する．受講者は別の入力パターンを試すとき，Eclipse では，実行しなおして，再び入力しなおす．ACE 上では，この作業を入力部に入力しなおすという単純な作業で実行できるため，普段の手順で別の入力パターンをテストする受講者がいたと考える．この問題に対し，過去の入力を記録しておき，過去と同じ入力した場合に，別の入出力部として分離するかを提案する機能によって改善できると考える．

「Eclipse と比べて，ここが良いと思った点はありますか？」という質問に対しては，「実行を毎回押す必要がないのでデバッグがしやすい」や「テストするための入力を何度も打たなくてもいい」という意見があった．これらの意見から，提案手法の恩恵が感じられていることが分かる．この効果を顕著にするためには，前述の改善が必要不可欠である．

6. おわりに

本研究では，プログラミング初学者が論理エラーを含むプログラムの提出を防ぐことを目的とし，プログラムのテストを支援するツール ACE を試作した．ACE にはプログラムの自動実行機能と複数入力の並列実行の機能を実装した．学生に参加してもらった試用結果から，ACE は授業の受講者がテストで自主的に使用するほど完成されていないことが分かった．改善点として，構文エラーの表示，ユーザビリティ向上のための UI の改善，過去の入力履歴に基づく提案を考察した．

今後の課題として，前述の改善点の実装と，機能（1），機能（2）の有効性を確かめる被験者実験を行うことが挙げられる．

参考文献

[1]稗方和夫：“プログラミング学習支援システム JavaEP の開発と導入”．*SIG-KST* 14，pp. 1-4．2011.
[2]片桐，酒井：“仕様をもとに入力と期待される結果を整理する機能をもった初学者向けプログラミング学習支援”．第 81 回全国大会講演論文集，pp. 549-550，2019.