

プログラミング講義における活動種別を考慮した エディタ操作と学生パフォーマンスの分析

三浦 元喜^{1,a)}

概要: 我々はブラウザのみで利用できる Web IDE に対して、エラーチェック機能やシンタックスハイライト機能、自動補完機能を追加することにより、初学者を支援する環境を構築してきた。また学習者の状況把握や、行き詰まりの解消支援を目的として、カーソル移動やマウスによるキャレット操作を含む、詳細な編集履歴を記録する Web エディタを開発してきた。本研究では、この Web エディタの将来的な支援機能拡張の前段階として、エディタの操作種別と、学生の演習・試験の成績との関連について、時間制限のある試験、時間制限のない演習、創作の自由度が高い課題という 3 種類の活動別に分析した。その結果、自動補完機能の導入に伴う、共通のエディタ操作の傾向がみられた。また点数・成績との関連については、活動の種類によって差異があることを確認した。

Analysis of Editor Operation and Student Performance in Consideration of Activity Types on Programming Lecture

MOTOKI MIURA^{1,a)}

Abstract: We have built an environment that supports beginners by adding error checking, syntax highlighting, and auto-completion functions to the Web IDE that can be used only with a browser. In addition, we have developed a Web editor that records detailed editing history, including caret movement by cursor key and mouse click operations, for the purpose of grasping learner's situation and helping to resolve deadlocks.

In this research, as a pre-stage for future extension of the Web editor's support functions, we analyzed the relationship between the editor's operation type and the student's practice/test scores by three types of activities. The three types of activities are "time-limited exams," "time-free exercises," and "time-free creative exercises with a high degree of freedom." As a result, we found the tendency of common editor operation with the auto-completion function. In addition, we confirmed that there were differences between the types of activities in relation to scores and grades.

1. はじめに

大学生を対象としたプログラミング講義においては、ソースコードをテキストで記述しながら演習や課題に取り組む方法が用いられることが多い。しかし、初学者はプログラミング言語の文法に対する知識が不足しており、かつタイピングによってソースコードを記述することにも慣れていない。

我々は、学習者が不慣れな環境構築・設定に手間取る

ことなく、プログラミング学習に集中できるようにするため、ブラウザのみで利用できる Web IDE に対して、エラーチェック機能やシンタックスハイライト機能を追加することにより、初学者を支援する環境を構築してきた [1]。また、学習者の状況把握や、行き詰まりの解消支援を目的として、カーソル移動やマウスによるキャレット操作を含む、詳細な編集履歴を記録する Web エディタを開発してきた [2]。

本研究では、この Web エディタの将来的な支援機能拡張の前段階として、Web エディタによって記録された実際のプログラミング講義のログを分析し、エディタの操作種別と、学生の演習・試験の成績との関連について、試験や

¹ 九州工業大学 基礎科学研究系
Faculty of Basic Sciences, Kyushu Institute of Technology
^{a)} miuramo@mns.kyutech.ac.jp

演習といった活動内容別に分析・調査する。

2. 対象講義

対象とした講義は、2019年度前期に開催された「情報処理基礎」(大学2年次対象, 69名受講)である。このうち、エディタ操作ログの研究利用に同意した、24名のデータを分析対象とした。

講義は、Processing言語を中心とした内容で、一部C言語も含まれている。16回の講義のうち、2回は中間試験(30点)と期末試験(30点)である。講義の初回は、Processingの基本的な描画命令を用いて、静止画作品を作成する内容である。その後、変数、条件分岐、繰り返し、配列、マウスイベント処理と描画といった内容を導入し、中間試験を実施する。後半は2次元配列や自作関数、再帰、配列要素の並び替え、クラスとArrayListを説明し、期末試験を実施する。なお、期末試験の翌週に、ペアで作成したProcessing作品の発表会を行っている。すべての演習、試験、作品作成は、次章で説明する、Webエディタ上で実施した。

3. Webエディタ

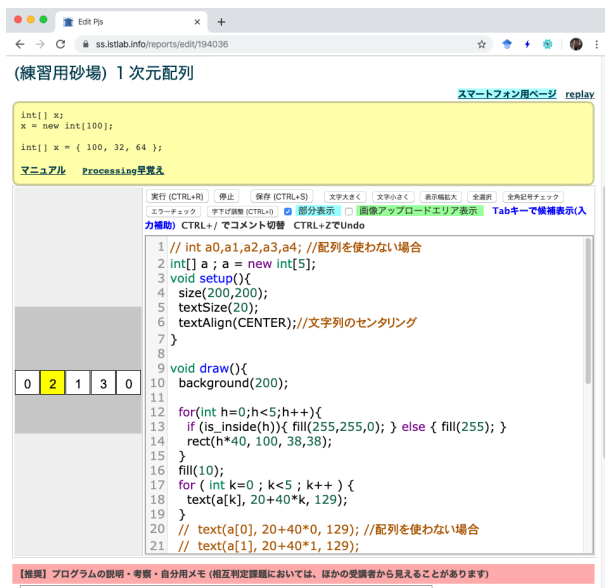


図 1 Webエディタ
Fig. 1 Web Editor

図 1 に、本研究で使った Web エディタ画面を示す。エディタ部分は CodeMirror を利用している。学習者がソースコードを編集し、CTRL+S を押すと、Web エディタはソースコードをサーバに保存し、エラーがなければエディタの左側でプログラムを実行する。

自動補完機能の動作を説明するため、[m] キーに続けて [Tab] キーを押下したときの画面を図 2 に示す。学習者はカーソルキー上下で候補を選択(図 3(左))するか、または続けてタイプすることで候補を絞り込む(図 3(中央))こ

とができる。候補を選択している状態で、Tab キーまたは Enter キーを押下すると確定となり、図 3(右)に示すように補完入力完了する。[Tab] キーを押さなければ、補完機能は起動せず、従来のエディタと同じ振る舞いとなる。また、[Tab] キーを行頭や、半角スペースのあとに入力した場合は、すべての候補が表示される。

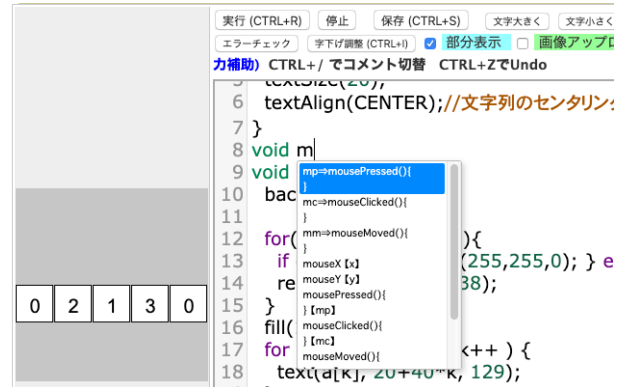


図 2 自動補完機能の呼び出し ([m] キーに続けて [Tab] キーを入力)
Fig. 2 Auto-complete function (Press [m] [Tab])

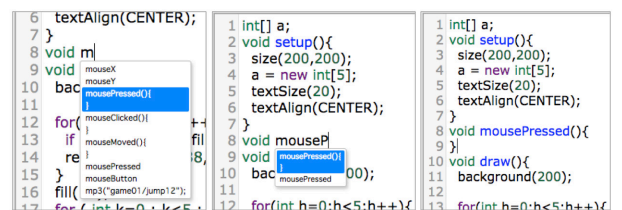


図 3 (左) カーソルキーで選択 (中央) 続くキーで絞り込み (右) Enter キーで確定
Fig. 3 (left) Select by cursor (center) Search by typing (right) Fix by pressing Enter

操作ログについては、編集操作イベントが 100 個蓄積されたとき、学習者が保存操作を行ったときに、JSON形式でサーバに送信した。編集操作イベントには、通常のキー押下のほか、カーソルキーによる移動やマウスによるカーソル位置設定、貼り付け操作によるコードの変更と、それらの操作時刻が含まれており、学習者のソースコード編集過程を再現することができる。

4. 分析

操作ログについて、(1) 試験、(2) 初期創造課題、(3) 演習の活動別に分析を行った。今回抽出した操作は、(1) 総イベント数 (total), (2) タブキー押下数/total (r_tab), (3) 文字キー押下数/total (r_chr), (4) カーソルキー押下数/total (r_cursor), (5) 括弧キー押下数/total (r_kakko), (6) Enter キー押下数/total (r_enter) の 6 種類で、各学習者の試験・課題・演習ごとに集計した。なお、total には、マウスによるキャレット移動も含まれている。これに (7) 当該試験・

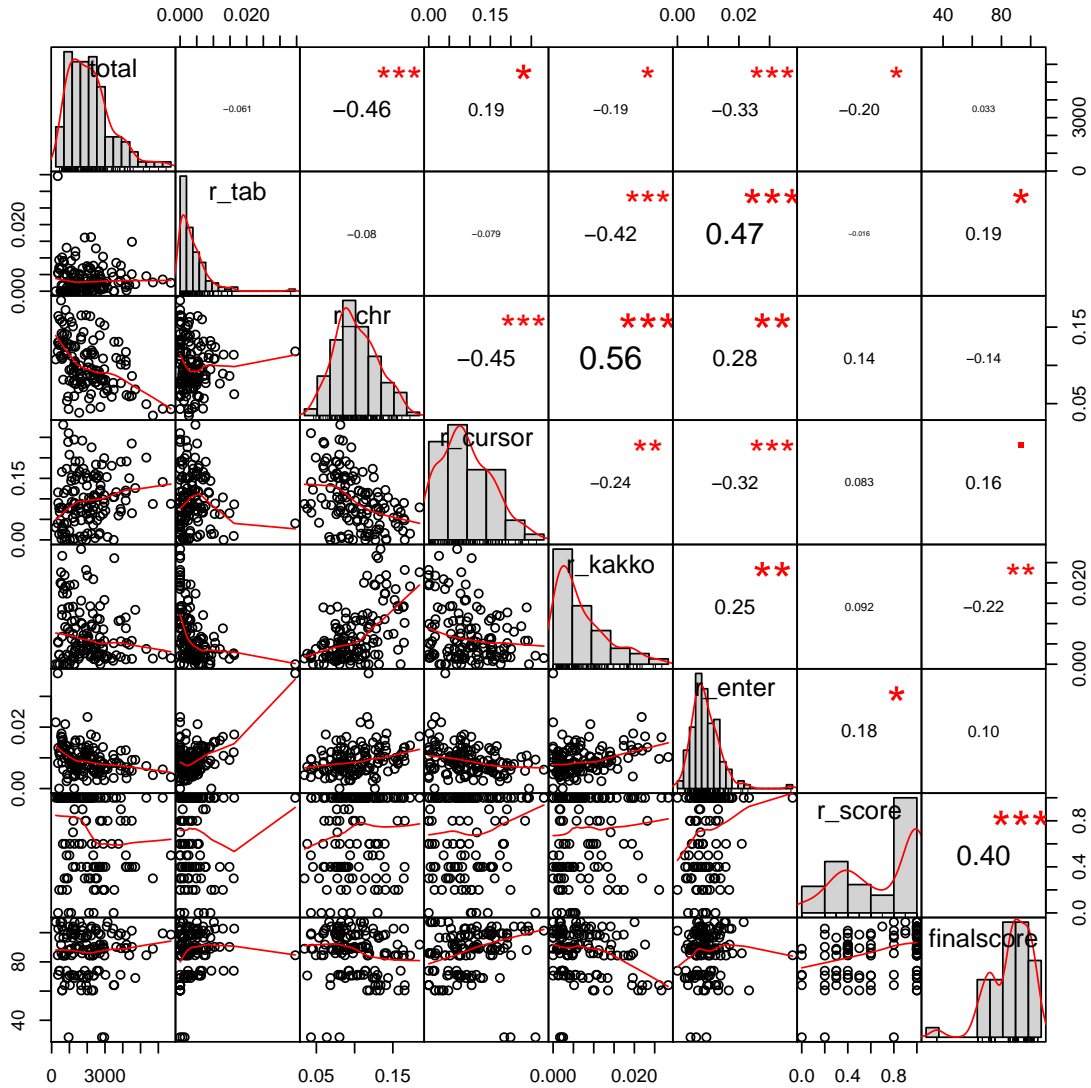


図 4 試験の分析表

Fig. 4 Analysis Table of Examination

課題・演習の獲得点/満点 (r_score), (8) 当該学習者の最終得点 (finalscore) を加え, 相関表を作成した.

試験の結果を図 4, 初期創造課題の結果を図 5, 演習の結果を図 6 に示す. マトリックスの対角成分はヒストグラム, 左下は散布図, 右上は相関を示している. 相関に付与された赤字のドットおよび*, **, ***は, それぞれ p 値が 10%, 5%, 1%, 0.1% 以下であることを示している.

4.1 試験

図 4 をみると, 文字と括弧の正の相関 (0.56***) が最も顕著であり, 次いで Tab と Enter の正の相関 (0.47***), 総数と文字の負の相関 (-0.46***), 文字とカーソルの負の相関 (-0.45***) が顕著であった. Tab と括弧の負の相関 (-0.42***) が現れる原因として, 自動補完機能で for 文や if 文の補完を行うと, 必要な括弧も含めて挿入されることが考

えられる. ちなみに, for 文をタブ補完した場合 `for (int k=0 ; k<10; k++) { }` が挿入される. Tab と Enter の正の相関 (0.47***) は, 自動補完の確定操作や, その後のブロック内への改行挿入が影響していると考えられる. 成績との関連では, 総数と得点の負の相関 (-0.20*), Tab と最終得点の正の相関 (0.19*) や, 括弧と最終得点の負の相関 (-0.22**), Enter と得点の正の相関 (0.18*) がでていますが, あまり顕著ではない. 総数と得点の負の相関は, 回答時の試行錯誤が影響している可能性がある.

4.2 初期創造課題

図 5 からは, 試験のときと同様の傾向として, Tab と Enter (0.62***), 文字と括弧 (0.59***), 総数と文字の負の相関 (-0.23**), 文字とカーソルの負の相関 (-0.17*) が確認できた. ただし, これらのうちの負の相関の 2 つは試

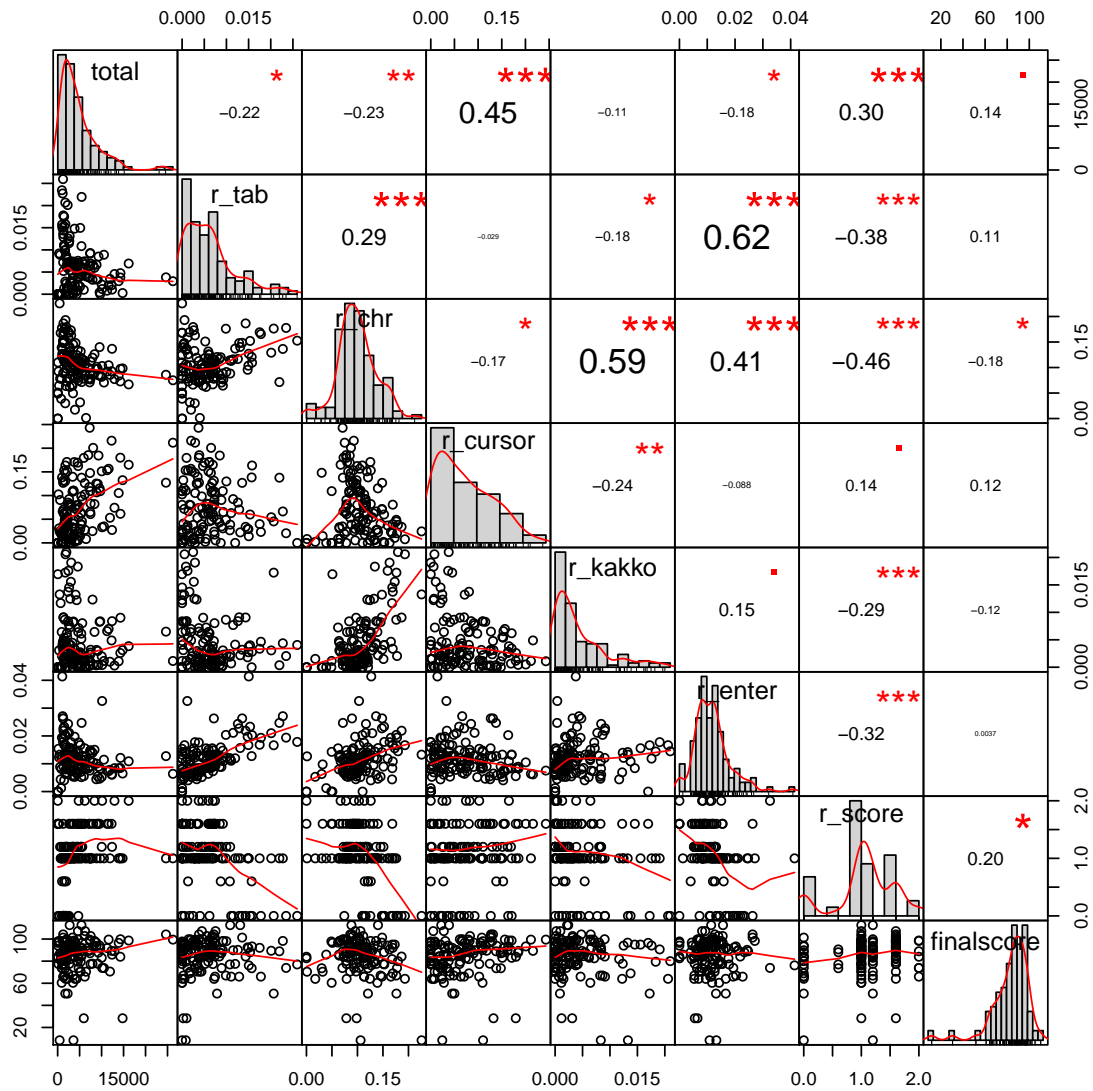


図 5 初期創造課題の分析表

Fig. 5 Analysis Table of First Drawing Assignment

験のときほど顕著ではなかった。独自の傾向として、総数とカーソル (0.45***) が顕著であった。成績との相関では、文字 (-0.46***), Tab (-0.38***), 総数 (0.30***), 括弧 (-0.29***), Enter (-0.32***) が顕著であった。初期創造課題では、基本的に多くの描画命令を駆使した作品が評価されるため、総数と得点の正の相関は理解しやすい。ちなみに、r_score が 1.0 を越えているのは、基本点に加えて努力点・ボーナス点を付与したことが影響している。それ以外の項目との負の相関を説明するには、最終的なソースコードの内容や、文字数、文字を削除した回数との関連も含めた追加調査が必要であると考えている。

4.3 演習

図 6 からは、前述の活動と同様、Tab と Enter (0.48***), 文字と括弧 (0.18***) の顕著な正の相関が確認できた。し

かし、初期創造課題と比較すると、成績との相関については Enter との弱い負の相関 (-0.14***) 以外現れてなかった。これは、演習の内容が、問題文で書かれた指示に従うよう機械的にソースコードを修正するものであり、自由な記述や柔軟性があまり要求されなかったことが影響していると考えられる。

4.4 考察

前述したように、定められた制限時間内での高度な回答が求められる試験と、時間制限のない演習、および自由度の高い初期創造課題の 3 つの活動に分けて分析を行ったところ、共通の傾向 (Tab による自動補完の効果としての、文字や括弧の入力減少) がみられた。ただし、学習者のエディタ操作の傾向の違いや、得点との相関の現れ方に差異が生じる項目も確認できた。それらのうちの一部について

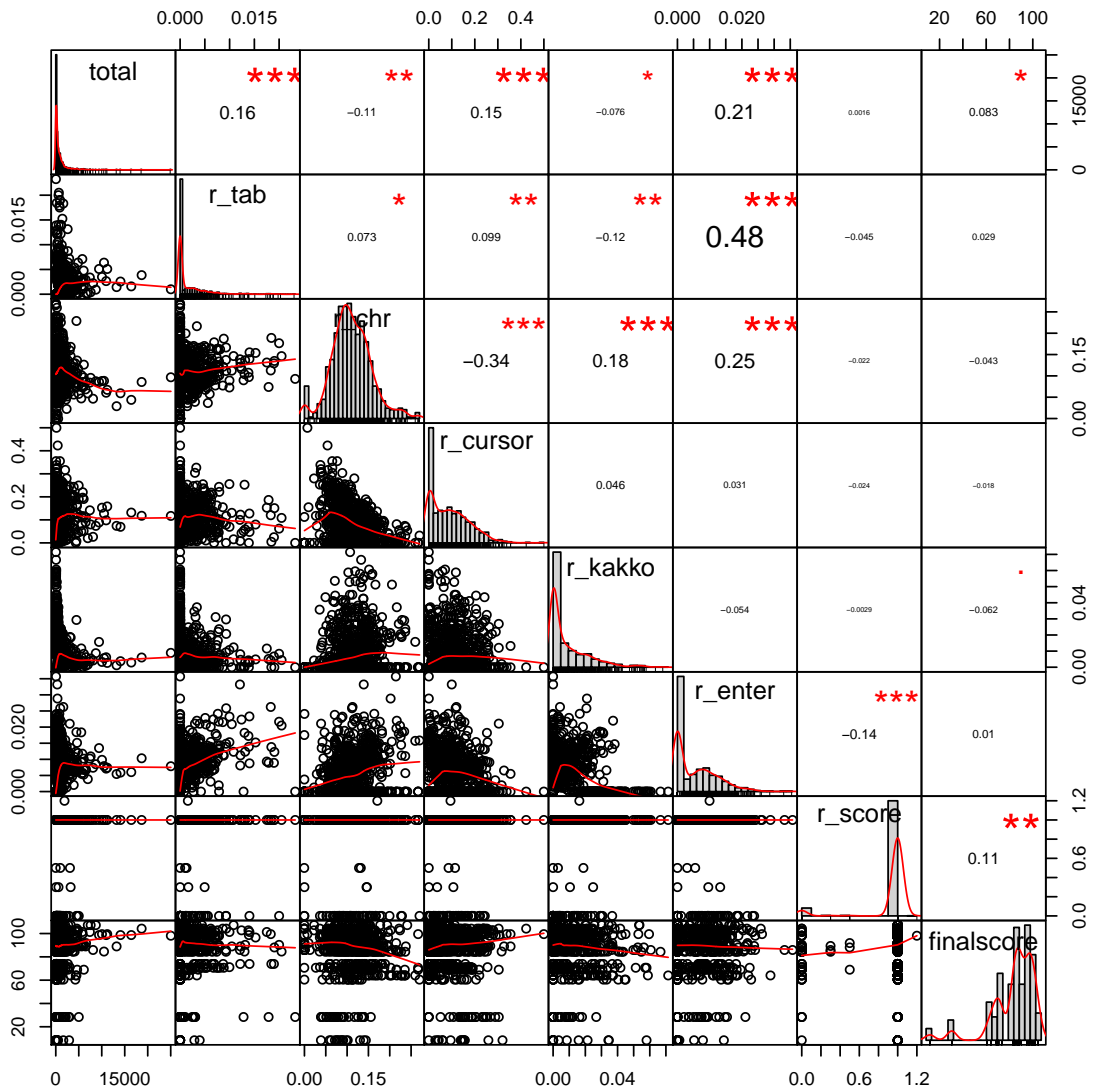


図 6 演習の分析表

Fig. 6 Analysis Table of Exercise

は、現時点で原因が不明確であるため、今後の追加調査が必要である。

5. 関連研究

市村らは、つまづいている学生の早期発見と、多くの学生が共通にかかえる問題の発見を可能にするための、Webブラウザ上で動作するプログラミング演習支援システムを構築している [3]。学生の操作ログ・エラーログを収集し、解析をすることで、いつまでもエラーを解消できない学生を検知したり、手を挙げる前に発見したりすることができるようになったと報告している。加藤らも、Webベースのプログラミング授業支援システムにおいて、コンパイルエラーと実行時の出力に基づいて学生の学習状況を判別し、教員が把握できる学習状況把握機能を構築している [4]。市村らの手法 [3] ではコンパイルエラーの種類を分類するの

に対し、加藤らの手法ではコンパイルエラーを模範解答プログラムの行に対応させて分類する点が異なると述べている。中澤らは、括弧やコンマ、セミコロンなどのプログラム構成要素の区切りを表す文字と、スペースキー、リターンキー、バックスペースキーなどの体裁調整や訂正に関わるキーが入力されたときに記録する粒度の高い方式を採用した、編集履歴可視化システムを構築している [5]。我々は、学生をつまづきの早期発見・検知・支援を目的として、Webエディタにおける学習者の操作について、カーソル移動やマウス操作を含む、ソースコードに直接反映されない操作履歴データを含めて記録するシステムを構築した。また、実際の講義で運用を行い、活動の種類別に分析を行うことによって、エディタ操作と点数・成績との関連を調査した。

6. おわりに

自動補完機能を含む、高度な入力支援機能を備えたプログラミング用 Web エディタにおいて、学習者の操作量と点数・成績との関連を、活動の種類別（試験、演習、初期創造課題）に分析した。その結果、Web エディタとしての共通の傾向が確認できた。また点数・成績との関連においては、活動の種類によって差異が生じることをあきらかにした。

我々はキャレット移動を含む、詳細な編集活動を考慮した、プログラミング学習支援環境の構築と有効性の確認を最終目標としている。今回の分析は、そのための基礎的な検討の一部である。今後は、カーソルキー以外の、マウス操作によるキャレット移動についても分析対象に加えることや、キャレット操作の方向や回数などから、操作の目的・意図といった高次の操作内容を抽出したうえで、つまずきの早期発見と支援に貢献できるかどうかを確認していきたいと考えている。

謝辞 本研究の一部は JSPS 科研費(基盤(C), 19K03056, 課題名「プログラミング教育における細粒度活動履歴に基づくラーニングアナリティクス」)の支援によるものです。

参考文献

- [1] 三浦元喜. Processing Web IDE を用いたプログラミング基礎教育の試み. 情報処理学会情報教育シンポジウム (SSS2013) 予稿集, pp. 225-231, August 2013.
- [2] 三浦元喜. 初学者向け Processing プログラミング環境におけるコード補完機能の導入と実践. 情報処理学会情報教育シンポジウム (SSS2018) 予稿集, pp. 142-149, August 2018.
- [3] 市村哲, 梶並知記, 平野洋行. プログラミング演習授業における学習状況把握支援の試み. 情報処理学会論文誌, Vol. 54, No. 12, pp. 2518-2527, 2013.
- [4] 加藤利康, 石川孝. プログラミング演習のための授業支援システムにおける学習状況把握機能の実現. 情報処理学会論文誌, Vol. 55, No. 8, pp. 1918-1930, 2014.
- [5] 中澤真, 梅澤克之, 平澤茂一. 粒度の高いコンテキストウェアな学習履歴を組み合わせた学習者分析. PACIS2018 主催記念特別全国研究発表大会, pp. 43-46, 2018.