

遠隔プログラミング演習におけるエラー解説機能の導入

大和 祐介^{1,a)} 石尾 隆^{1,b)} 嶋利 一真^{1,c)} 松本 健一^{1,d)}

概要：新型コロナウイルスの感染拡大の影響により、現在多くの教育機関が遠隔授業を取り入れている。それに伴い、プログラミング教育も遠隔で行われている。しかし、遠隔は対面と比べ、授業を受けている学生がどこで躓いているのかわかりづらい。本研究では、学生がエラーについての理解を深め、エラーを修正できるようにするために、本研究では学生の開発環境内にエラー解説機能を導入した。プログラミングの授業を受講している学生の質問チャットデータに着目し、その効果を調べた結果、全質問に占めるエラーに関する質問の割合が減少したことを確認した。

キーワード：プログラミング教育、遠隔授業

1. はじめに

近年新型コロナウイルスの感染拡大の影響により、多くの教育機関が遠隔授業を行っている。しかし、遠隔授業は教員が学生の表情や反応を確認しづらいため、学生が躓いている箇所やタイミングがわかりづらいという問題がある [1]。教員が学生を個別に指導することも難しいため、学生が遭遇した問題を自己解決できるように支援する技術が研究されている [2], [3]。

プログラミング演習において学生が遭遇する問題の1つが、コンパイルエラーや実行時エラーである。初学者にとってエラーメッセージはわかりにくいことから、既存研究 [4] では、エラーメッセージに対応する Stack Overflow の質問を自動検索し、リンクを提示する手法が提案されている。しかし、Stack Overflow 上で行われている質問や回答は、学生が使っているプログラミング環境に関係がないため、教員が行えるような、プログラミング環境に対する操作の指示のような情報を含まない。

本研究では、Python の処理系がエラーメッセージを出力した場合、それに対応する日本語の解説を表示する機能を導入する。図 1 にその動作例を示す。Python プログラムでは `input` という関数を使って標準入力の内容を 1 行読み取ることができるが、学生が何も入力を与えないとエ

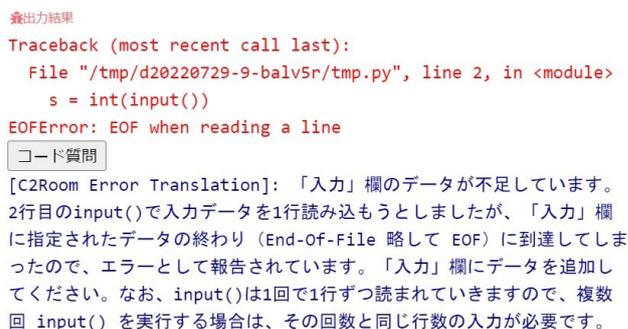


図 1 エラー解説の例

ラーとなってしまいうため、開発環境の標準入力を指定する欄に入力を与えることを指示する解説が表示されている。

解説表示機能の作成にあたっては、著者らの所属大学で実施したプログラミング演習の 2021 年度のチャットデータ、エラーデータを分析した。課題ごとに質問が多いエラーや頻繁に発生するエラーを特定し、発生頻度の高いエラーから順番に発生原因と解決策を考え、エラーメッセージの種別を検出する正規表現のルールと、対応する解説メッセージを執筆した。

本機能の導入によってプログラミングのエラーに関する質問は減少することが期待されることから、導入前後の授業での、学生の質問内容を保存したチャットデータを分析する。

2. 調査方法

本研究では、2021 年度および 2022 年度に、著者らの所属大学で実施したプログラミング演習の質問データを分析

¹ 奈良先端科学技術大学院大学
Nara Institute of Science and Technology, Ikoma, Nara, 630-0192, Japan

a) yamato.yusuke.yy7@is.naist.jp

b) ishio@is.naist.jp

c) k.shimari@is.naist.jp

d) matumoto@is.naist.jp

表 1 Python に関する質問の分類結果

	2021		2022	
エラーに関する質問	20	(16%)	16	(9%)
エラー以外の質問	105	(84%)	162	(91%)
合計	125	(100%)	178	(100%)

する。この科目は、情報科学以外の分野の出身で、博士前期課程では情報科学を専攻したい学生向けに、Python によるデータ分析のためのプログラムの書き方を教えるとともに、プログラミングの考え方を理解してもらうことを目的としている。授業は全 8 回の講義で構成されており、学生は各授業回で練習問題と課題を与えられる。授業の前半では教員が Python についての講義を行い、学生は講義を基に練習問題に取り組む。授業後半は演習の時間で、各自で課題に取り組み、わからないことがあればチャットでティーチングアシスタントに質問するという流れで行う。演習には dTosh C2Room^{*1}を使用しており、学生が質問のために送受信したチャットメッセージが記録されている。

データセットは、2021 年度および 2022 年度にプログラミング演習を受講した学生とティーチングアシスタント計 186 名のチャットデータ、実行したコード、発生したエラーデータである。学生からは授業開始前にプログラムおよび質問内容の研究への利用の許諾について質問しており、明示的に許諾を得られた学生のデータのみを取得している。2021 年度は学生 78 名、ティーチングアシスタント 6 名、2022 年度は学生 94 名、ティーチングアシスタント 8 名のデータを取得した。データには質問した時刻、質問内容、質問した人を区別するための ID が含まれている。

エラー解説の導入により、全質問に占めるエラーに関する質問数の割合がどのように変化したかを調べるため、まず、学生からの質問を Python に関するもの、授業の進行に関するもの、システムの利用法に関するものへと目視で分類し、Python に関する質問のみを取り出した。取り出した質問を、さらにエラーに関する質問とそれ以外の質問に分類し、エラーに関する質問ほどの程度減少しているか分析を行った。

3. 調査結果

表 1 に、エラーに関する質問とエラー以外の質問の個数を整理したものを示す。エラーに関する質問は、2021 年度は 78 人中 13 人から 1~3 回、合計で 20 回であった。2022 年度は 94 名のうち 8 名から 1 回ずつ、1 名から 8 回の質問があり、合計 16 回であった。合計に占めるエラーに関する質問の割合は減少している。また、エラー解説機能は合計 1563 回利用された。

表 1 に対し、「2021 年度と 2022 年度でエラーに関する質問の割合は変化していない」という仮説を設定し、フィッ

シャーの正確確率検定を行ったところ、 p 値は 0.072 となり、エラー解説を導入した場合においても有意な差は確認できなかった。この原因として、全体に占めるエラーに関する質問をした人の数が少ないことと、2022 年度で一人で多くの質問をした人がいたことが考えられる。

4. 関連研究

Fagerholm ら [5] は、プログラミング環境での学生の行動として、主に解答までの時間を分析対象としている。しかし、質問内容までは分析されていない。質問に関する研究としては、プログラミング経験が浅い学生は、質問自体を考えることができないのではないかと [6] という仮説のもとで、学生に対してプログラムの動作を質問することで理解を促す仕組みが考えられている [6], [7]。

5. まとめと今後の課題

本研究ではエラー解説機能の導入前後の、エラーに関する質問数を分析した。その結果、エラーに関する質問数の割合は減少したが、統計的な有意差はなかった。今後の課題として、質問チャットだけでなく、発生したエラーの数、種類、エラー修正時間についても分析を行うことで、エラー解説機能の効果をより詳しく調査する必要がある。

謝辞 本研究は JSPS 科研費 Grant Number JP20H05706 の支援を受けたものです。

参考文献

- [1] 中村好甫, 安藤和久, 内田圭佑, 小川直樹, 権 赫虹: 大学におけるオンライン授業の可能性と課題: 学生と教団実習を経験した大学院生へのインタビューを手がかりに, 教職課程担当教員養成プログラム報告書, pp. 3-13 (2021).
- [2] 鈴木舜也, 吉野 孝: 質疑応答のお知らせ機能をもつ授業支援チャットボットの提案, 2021 年度情報処理学会関西支部 支部大会 講演論文集 (2021).
- [3] Wu, E. H.-K., Lin, C.-H., Ou, Y.-Y., Liu, C.-Z., Wang, W.-K. and Chao, C.-Y.: Advantages and Constraints of a Hybrid Model K-12 E-Learning Assistant Chatbot, *IEEE Access*, Vol. 8, pp. 77788-77801 (2020).
- [4] Thiselton, E. and Treude, C.: Enhancing Python Compiler Error Messages via Stack, *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 1-12 (2019).
- [5] Fagerholm, F. and Hellas, A.: On the Differences in Time That Students Take to Write Solutions to Programming Problems, *2020 IEEE Frontiers in Education Conference (FIE)*, pp. 1-9 (2020).
- [6] Henley, A., Ball, J., Klein, B., Rutter, A. and Lee, D.: An Inquisitive Code Editor for Addressing Novice Programmers' Misconceptions of Program Behavior, *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, pp. 165-170 (2021).
- [7] Lehtinen, T., Santos, A. L. and Sorva, J.: Let's Ask Students About Their Programs, Automatically, *2021 IEEE/ACM 29th International Conference on Program Comprehension (ICPC)*, pp. 467-475 (2021).

^{*1} <https://c2room.jp/>