

7ZH-05

# コンパイルエラー学習教材とソースコード編集過程を 振り返ることによる修正支援\*

金澤 勇輝†

静岡大学大学院 総合科学技術研究科

酒井 三四郎‡

静岡大学 情報学部

## 1 はじめに

コンパイルエラーが発生した際、プログラミング学習者はソースコードの修正が困難となってしまう場合がある。これは事前にコンパイルエラーの修正方法について学ぶ機会がないこと、過去のコンパイルエラーの修正履歴を学習者が確認し辛いことの2つが原因であると考えた。

本研究では前者の問題を解決するために学習者にコンパイルエラーが発生するソースコードを提示しそれを修正してもらう課題(コンパイルエラー学習教材)を作成し、後者の問題については学習者が自身のエディタ内の編集履歴を振り返ることのできるツールを作成した。事前に学習し、振り返ることが出来るようにすることでコンパイルエラーの発生数を減らし、学習者のコンパイルエラーに対する抵抗感を軽減させることを目指す。

## 2 関連研究

平尾らはコンパイルエラーの修正履歴を観察できるツール [1] を開発した。コンパイルエラーそれぞれの修正時間の推移グラフを学習者に提示することによってコンパイルエラーに対する恐怖感が軽減されるという効果が得られている。榎原らは探索的なプログラミング活動を支援する環境を提供するツール [2] を開発した。学習者が自身のプログラミング編集過程を確認することでプログラミングの複雑さを回避し、課題が解きやすくなったという声が得られている。

本研究では事前にコンパイルエラーについて触れ、その履歴も振り返ることにより抵抗感が軽減されるのではないかと考えた。榎原らの研究は同じファイルについての編集過程を確認しながら課題を解くのにに対し、本研究では同じエラーがあった別のファイルを振り返りながら課題を解いてもらう。作成したツールは以前提案したツール [3] をベースにし、ユーザインタフェースを変更し、学生にとって大きな負担となる可能性のある整理する機能が取り除かれている。加えて学習教材を用意し、それを学習した履歴も振り返ることが出来るようにした。

\*Code correction support using compile error learning materials and source code editing process

†Yuki Kanazawa, Graduate School of Integrated Science and Technology, Shizuoka University

‡Sanshiro Sakai, Faculty of Informatics, Shizuoka University

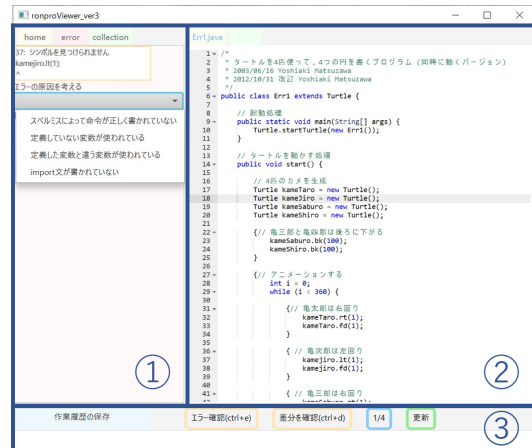


図 1: ツールの実行例

## 3 提案手法

### 3.1 コンパイルエラー学習教材

教材を作成した目的は学習者にどのようなコンパイルエラーが存在するか、それが発生した際の修正方法について学んでもらうことである。教材の対象エラーは昨年度の学生のコンパイルエラーの履歴を基に数が多かったもの、修正に苦勞していたものとした。1つあたり50行程度のソースコードで数行の改編で修正が出来る。これは後にツールで再び確認することが出来る。教材を解く際にはそのエラー文に対する複数の原因の候補に対応したタグが与えられ、それを参考にしながら解いていく。

### 3.2 提案ツール

作成したツールは次の機能を持つ。

- 編集過程のソースコードそれぞれについてどのようなコンパイルエラーが発生していたかの表示
- ソースコード間の差分の表示
- コンパイルエラーごとの検索

ツールの役割は学習者が自身のプログラミングの編集過程を確認出来るようにすることで過去に経験したコンパイルエラーに再び遭遇した際に参考に出来るようにすることである。ツールの実行例を図1に示す。

ツールは3つの要素から構成されている。①のエリアはタブ構造になっており図1では赤色のerrorタブを表示している。②で表示しているソースコードをコンパイルした際に発生したコンパイルエラーが表示される。灰色

表 1: 2018 年度と比較したコンパイルエラー平均発生数の増減

対象エラー	E	0	-0.8				-0.1	0.1
	D	0.2	-0.7	-0.3	-0.1	-0.5	-0.6	-0.4
	C	2	-0.3	1.2	0.6	-0.2	3	0.9
	B	-0.1	-2.3	0.4	0.6	-0.1	1.3	-0.8
	A	-0.1	-0.6	0.3	0	2.7	0.4	1.2
		5	6	7	8	9	10	11
		授業回						

のプルダウンメニューをクリックすると、そのエラー文に対する、予め用意された複数の原因が記されたタグを確認、登録することが出来る。黄色の home タブはファイルツリーを表示し、ここで表示させたいファイルを選ぶ。緑色の collection タブは今までに学習者が発生させたコンパイルエラーの一覧が表示され、同じコンパイルエラーを起こしたファイルを検索できる。それらのファイルは前述したタグごとに分類されている。③はツールバーを表示するエリアで①の error タブに切り替えるボタンや編集過程の中で任意のソースコード間の差分を表示させるボタンがある。

#### 4 実験・結果

仮説として「①事前にコンパイルエラーについて学習することで発生率が減る」「②ツールを用いることで修正に対する抵抗感が軽減される」を設定し、検証するために2019年度の1年生向けのプログラミング講義において約70人を対象に実験を実施した。実施した期間は第5回から第11回の講義である。初めに学生にツールを配布、使い方の説明を行い、各回では5問ずつコンパイルエラー学習教材を配布した。実験課題の具体的な内容として、通常の講義の課題に取り組む前に、配布したソースコードを修正してコンパイルエラーが発生しない状態にすることを条件とした。教師が作成した模範解答も同時に配布しそれを見ながら課題を解いても良いこととした。各回で条件を達成した学生の割合は81%であった。仮説の検証方法について、①については課題の対象としたコンパイルエラーが通常の課題を解く際に減少したか2018年度の学生と比較を行った。②については実験が終了した次の週の第12回講義時にアンケートを実施した。

実験結果を表1に示す。横軸は授業回、縦軸は対象のコンパイルエラーの種類を表す。コンパイルエラーそれぞれについて発生回数の平均を求め、2018年度の発生回数の平均との差を小数第2位を四捨五入した。青色のマスは2019年度の方が減少し、赤色のマスは増加したことを表す。わずかでも減少したコンパイルエラーは16件、増加したコンパイルエラーは14件であった。

次にアンケートの結果を表2に示す。アンケートの有効回答数は53件であった。ツールを使用したことでコンパイルエラーが修正しやすくなったと感じた人が全体の

表 2: ツールの評価アンケート

ツールの使用によってコンパイルエラーが修正しやすくなったと感じたか	人数
そう思った	11
少しそう思った	26
どちらとも言えない	5
あまりそう思わなかった	4
そう思わなかった	1
使わなかった	6

約60%であった。

#### 5 考察

仮説の①について、対象エラーそれぞれについて平均の発生回数がわずかでも減少したものと増加したものにほとんど差がないため、事前に問題を解くことでコンパイルエラーの発生回数が低くなるとは言えなかった。原因として問題を解いたあとで対象エラーに遭遇した際に同類だと思えなかったこと、一つのエラーメッセージに対して原因が複数あることが多く、教材だけではカバーすることが出来なかったことが考えられる。仮説②について、ツールを使うことがコード修正に対して有効であるという意見が多く得られた。有効と感じなかった原因について、ツールの使い方に関する説明が足りなかった、ツールの使い勝手が悪く自分で考えた方が速かったという意見がアンケートから得られた。

#### 6 おわりに

本研究では自身のソースコード編集過程を振り返ることの出来るツールの開発と学習教材の作成を行った。約70名を対象に実験を行った。教材の対象としたコンパイルエラーについて、解いた方がコンパイルエラーの発生回数が減少するとは言えなかったがアンケートからツールを使うことで課題が解きやすくなったという意見が多く得られた。今後の課題はツールの使いやすさの向上である。

#### 参考文献

- [1] 平尾元紀, 松澤芳昭, 酒井三四郎: Compile Error Collection Viewer:修正履歴分析によるコンパイルエラー学習支援システム, 情報教育シンポジウム 2014 論文集,2014(2),pp.151-158,2014
- [2] 槇原絵里奈, 藤原賢二, 井垣宏, 吉田則裕, 飯田元:初学者向けプログラミング演習のための探索的プログラミング支援環境 Pockets の提案. 情報処理学会論文誌,57(1),pp.236-247,2016
- [3] 金澤勇輝, 酒井三四郎:プログラミング編集過程を学習者が整理することによるコンパイルエラー修正支援. 情報処理学会第81回全国大会講演論文集,2019.1:pp547-548,2019