Vol.2020-HCI-187 No.12
2020/3/16

意図的に埋め込まれたバグによるプログラミング学習支援

岩渕悠太[†1]　高島健太郎[†1]　西本一志[†1]

概要：企業の ICT 新人研修において，プログラミング教育が行われている．その教育に 2 つの問題が発生している．プログラミング教育における知識の定着率の格差と社内講師の本来業務に割く時間の減少である．プログラミング学習支援システムはこれまでに多く開発されてきた．しかし，この問題の根本的な原因は，従来の学習方法が「受動的な学習」となっていることにあるのではないかと我々は考えた．そこで本研究では，学習教材に誤情報を混入し，これを正しく修正する学習を行わせることで，受動的な学習を回避し，能動的な学習を促す e-ラーニングシステムを提案する．本稿では，実装したシステム BugTutor について述べると共に，これまでに実施したユーザスタディに基づく提案手法の有効性の検証結果を報告する．

キーワード：誤情報，プログラミング，アクティブラーニング

A Support Method for Learning Programming by Intentionally Embedding Bugs in Sample Codes

YUTA IWABUCHI[†1]　KENTARO TAKASHIMA[†1]　KAZUSHI NISHIMOTO[†1]

Abstract: There are two issues with programming training for new employees in ICT companies: low retention rate of knowledge once learned and wasting the instructor's time. Various learning support systems of programming have been developed so far. However, they could not solve the above-mentioned issues because of their passive learning characteristics. Therefore, in this research, we proposed a novel e-learning system that promotes active learning by embedding some bugs in learning materials and letting the learners correct it. This paper describes "BugTutor," which we developed and shows the results of user studies.

Keywords: incorrect information, programming, active learning

1. はじめに

日本では，IT 人材需要が今後拡大する一方で，供給が減少することが予想されている．そのため，需要が供給を上回り IT 人材の不足が深刻化する. 経済産業省のデータによれば，2020 年には約 37 万人，2030 年には約 79 万人の IT 人材が不足すると推測されている[1]. そのため，IT 人材育成は重要性が高く，意義があるものである．

本稿第 1 著者は，新人研修でプログラミング教育を担当した経験がある ICT 系企業 3 社の社員 7 名に対して，プログラミング教育の問題点をインタビューした．共通して指摘された問題点は，社内講師 1 人に対して複数人の研修生で学習を行った場合に知識の定着率（教授された事柄が一定期間以上記憶されている割合）が特に低いこと, 並びに, 社内講師が抱えている本来業務を実施するための時間が奪われることであった．よって，ICT 教育の職場内職業訓練（OJT）における研修生の知識定着率を効率的に向上させる教育手段の実現が求められている．

OJT 教育におけるこのような問題は，講師側から一方的に指導が行われる受動的な学習に原因があり，その解決には能動的な学習手段の導入が有効であると，筆者らは考えている．杉山ら[2]は，アクティブラーニングを導入することで，受動的な学習を能動的な学習に転換することを試みた. その結果, アクティブラーニングを導入したクラスが, テスト成績において高い得点をとることを示した．枝川ら[3]は，理科を中心にアクティブラーニングを用いた授業実施と授業改善に取り組んだ．生徒らは，自分自身の言葉で他者に説明が可能になり，他者の考えを自身の考えと比較し，検討を行えるようになった．更に，生徒の科学思考力・推論力の変容をテストした．アクティブラーニングによる授業実施クラスと実施しないクラスで比較した結果，アクティブラーニングを導入したクラスにおいて，生徒の論理的思考力・推論力がより向上することを示した．

しかしながら，プログラミング学習を支援する既存の研究では，受動的な形態での学習支援が多く，能動的な学習によって学習効率の向上を目指す提案は少ない．そこで本研究では，IT 人材育成の一環として，企業の新人研修において行われている ICT 教育に着目し，能動的な学習を導入することにより，初学者の知識定着率を向上させる手法を提案する．提案手法に基づき構築した e-learning システム BugTutor を用いた実証実験を実施し，提案手法の有効性を検証する．

2. 関連研究

多くの初学者にとってプログラミングは難度が高く，知識の定着率が低いことが問題となっている．そこでこれま

†1 北陸先端科学技術大学院大学 先端科学技術研究科
Graduate School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology

で，初学者のプログラミング学習を支援するためのシステムが，数多く開発されてきた．西田ら[4]は，ソースコードの入力支援や，プログラムの実行状態表示機能を持つ，DNCL を拡張したプログラム学習環境 PEN を実装した．井垣ら[5]は，各学習者のコーディング状況を記録・可視化し，進捗状況を講師に提示することで，遅れている学習者に対して講師が個別指導を行うことができる，コーディング過程可視化システム C3PV を実装した．田口ら[6]は，プログラミング教育において，学習者の理解状況，並びに学習者の学習意欲に応じて，演習科目に反映させるプログラム教育の支援を提案している．

オンライン教育についても，これまで，数多くの研究が行われてきた．穂屋下ら[7]は，大学内での講義で，各教科についてネット講義（WEB 上で授業を行うスタイル）を試みた．質問やレポート提出など，すべてネット上で処理を行った．結果，ネット講義に変更したことによって，教員が出張などのときに利用できる，不特定多数の学生がいつでも何度でも聴ける，質問がしやすいので質問が多い，などのメリットが得られた．植野ら[8]は，高等専門学校において統計学を遠隔授業（e-ラーニング）で実施している．授業コンテンツ画面では，対応する HTML 教材と映像教材が同期しながら提示される．コンテンツ作成手順は，授業担当教員が HTML 形式の教材・テストの作成を行い，映像教材を撮影し，編集するというものである．受講生は，すべてのコンテンツにいつでもアクセスが可能としている．授業担当教員は，受講生の進捗を把握することができ，特に学習が進んでいない生徒には個人メールを送る授業形態をとった．結果，授業が始まってからの授業担当教員の授業外での役務は，進捗の遅い生徒と学習プロセスが異常（コンテンツ学習時間が極端に短いなど）な生徒に指導メールを送るのみとなり，最長でも 30 分程しか要さず，教員にとっての労力を非常に軽減できたことが示されている．

3. 提案手法

本研究では，能動的な学習を行わせるため，誤った情報を含んだ教材で学習させる e-ラーニングシステムを提案する．プログラミング学習において一般的に採用されている講義形式を e ラーニング形式に変更するのは，2 章で述べた関連研究の知見に基づき，社内講師の拘束時間を少しでも削減し，講師の本来業務の妨げにならないようにするためである．また，誤った情報を含んだ教材で学習させるのは，間違いを直す作業を行わせることで能動的な学習を誘発し，知識の定着率を向上させるためである．具体的には，学習内容の説明文には正しい情報を与えるが，サンプルプログラムに誤情報（バグ）を混ぜ込んだものを提示して学習を行わせる．

初学者の学習について，誤情報を含んだ教材の有効性を示した先行研究が Adams ら[9]によってなされている．この研究では，小数点の大小を比較する問題を課題として，正しい情報に基づいて学習を進めたグループと，誤った情報を含んだ状態で学習を進めたグループを比較している．学習後，短い時間の後に実施したテストでは，両者の成績に有意な差は見られなかった．しかしながらが，数日後に再度テストを行った際には，誤情報を含んだ学習を行ったグループの成績が有意に良いという結果が得られている．このような結果となった理由として，エラー処理を行うことによって，能動的な学習が体験できた可能性が指摘されている．また，プログラミング学習を対象とした事例として，初学者が支援対象ではないが，知見ら[10]は，事前に取得したプログラミングで発生したエラーや，エラーが発生したソースコード，エラーから学んだ教訓などを失敗情報として学習中に提示することを試みた．学習修了後にテストを実施した結果，失敗情報を利用した学習グループが，失敗情報を利用しなかったグループよりも高い得点をとることを示した．これらの先行研究における誤情報の有用性に関する知見に基づき，一般的には正しい事例が示されるサンプルプログラムに最初からバグを埋め込んでおくことで，漫然とサンプルプログラムを入力して実行するだけの受動的な学習になることを防止できるとともに，知識の定着率を向上させられるのではないかと考えた．

なお，当然のこととして，初学者に誤情報だけを与えて学習を進めてもらった場合，正しい情報を覚えられないことが想定される．一定時間後に，誤りの情報について正しい情報を提供する必要がある．そのため，本研究では，学習が始まってから 15 分後に正しい情報を提供するようにする．

4. 実験システム BugTutor

第 3 章で述べた提案手法に基づき，初学者を対象としたプログラミング学習支援システム BugTutor を構築した．BugTutor は，Python 3.6.8 を使用して構築された，Windows OS 対応の GUI ソフトウェアであり，学習・解説・演習の 3 つのフェイズで構成されている．BugTutor のシステム構成利用環境を図 1 に示す．本システムは，提案手法の有効性を検証するためのプロトタイプであるため，システム内にプログラム実行環境を搭載していない．そのため，web 上で Python3 を実行できる，PaizaCloud を利用した．ノート PC にはシステム画面を表示し，外付けのディスプレイには PaizaCloud を表示する．

BugTutor での学習を開始すると，まず図 2 に示す学習ページが提示される．このページでは，3 章で述べた提案手法に基づき，if 文などの当該ページにおける学習項目についての正しい説明文と，バグが埋め込まれたサンプルプログラムが提示される．なお，埋め込むバグとしては，たとえば，if 文の学習であれば，比較演算子の “==” が “=” になっていたりするような，当該学習項目で初心者がよく

図 1　BugTutor の利用環境
Figure 1.　Setup of BugTutor

作り出すバグを埋め込んでいる．学習者は，解説文を読んで当該ページにおける学習内容を理解するとともに，示されたサンプルプログラムを PaizaCloud 上で打ち込んで，動作を確認して学習する．しかし，サンプルプログラムにはバグが埋め込まれているため，そのまま打ち込んでもエラーを起こして動作しない．学習者には，バグをとって，サンプルプログラムが動作するように修正することが求められる．最終的にサンプルプログラムを動作するよう修正できたか確認するために，実行結果を送信する機能を用意した．

解説ページの例を図 3 に示す．このページでは，正しいサンプルプログラムとバグの説明を提供する．解説ページは，自動的に表示されるのではなく，学習ページに用意されたボタンを学習者が押すことによって表示される．ただし，学習ページでの学習開始から 15 分経過していない状態でボタンを押した場合は,「もっと考えてください」とだけ表示され，解説ページは表示されない．

学習フェイズで与えられたバグ入りのサンプルプログラムが正しく修正され,想定通りの結果が送信された場合，演習フェイズに移行する．図 4 に演習ページの例を示す．演習ページで提示された演習課題について，ヒントを参考にしながら PaizaCloud 上でプログラミングを行い，正しく動作すると思われるプログラムを作成する．できあがったプログラムのソースコードを画面左下の入力領域に貼り付けて送信ボタンを押し，またその実行結果を画面中央下の入力領域に貼り付けて回答ボタンを押す．入力された実行結果情報が正しければ，Congratulations! と表示され，次の学習項目に関する学習ページに移行する．

5.　評価実験

評価実験では，提案する学習支援システムの有用性を検証するために，プログラミング初学者に対して，BugTutor を用いる被験者 6 名（A, B, C, D, E, F）の利用グループと，BugTutor を用いない 6 名（G, H, I, J，K，L）の非利用グループに分けて比較実験を行った．非利用グループには，学習ページで提示するサンプルプログラムにバグを埋め込んでいないものを提示して学習を行わせた．第 1 章で述べた IT 人材の不足について，経済産業省より出されているデータ[1]より，AI 技術者の不足が予測される．そのため，実験で使用する学習対象言語を，AI 技術に使用されることが多い Python3 とした．

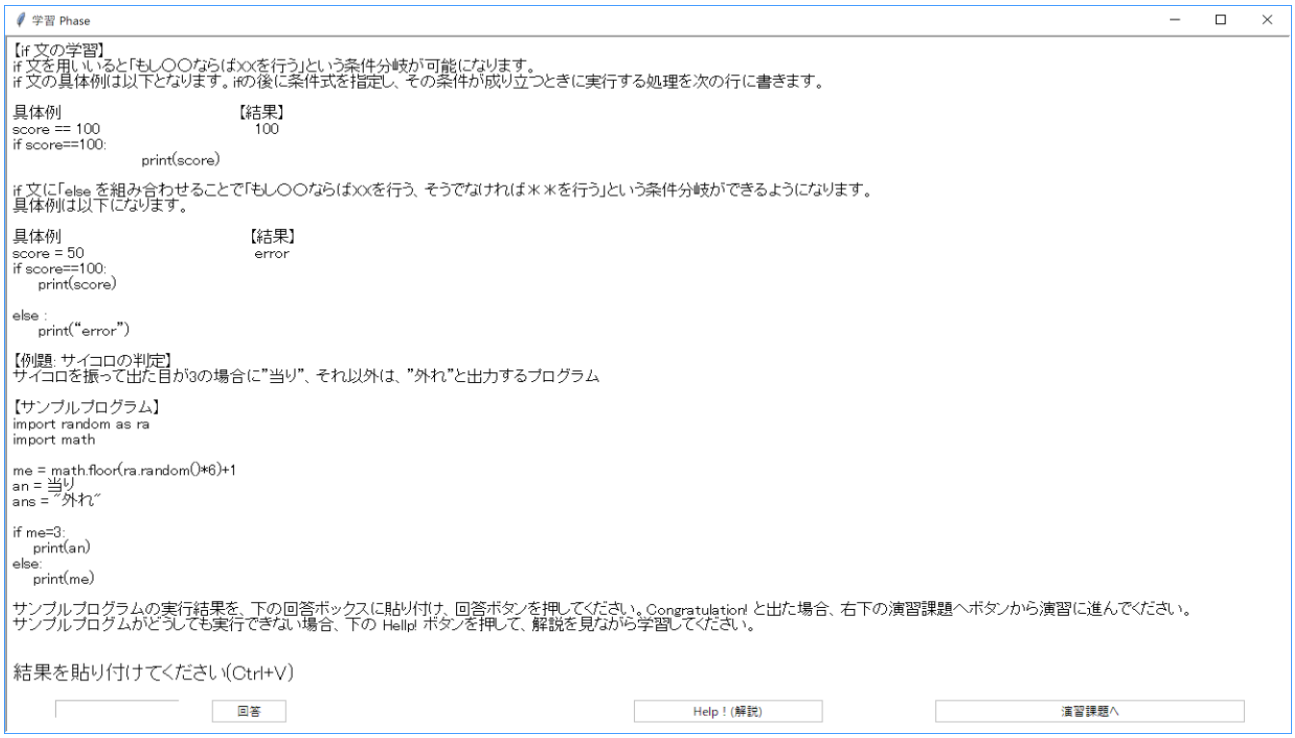

図 2　学習ページの例
Figure 2.　An example of a tutorial page

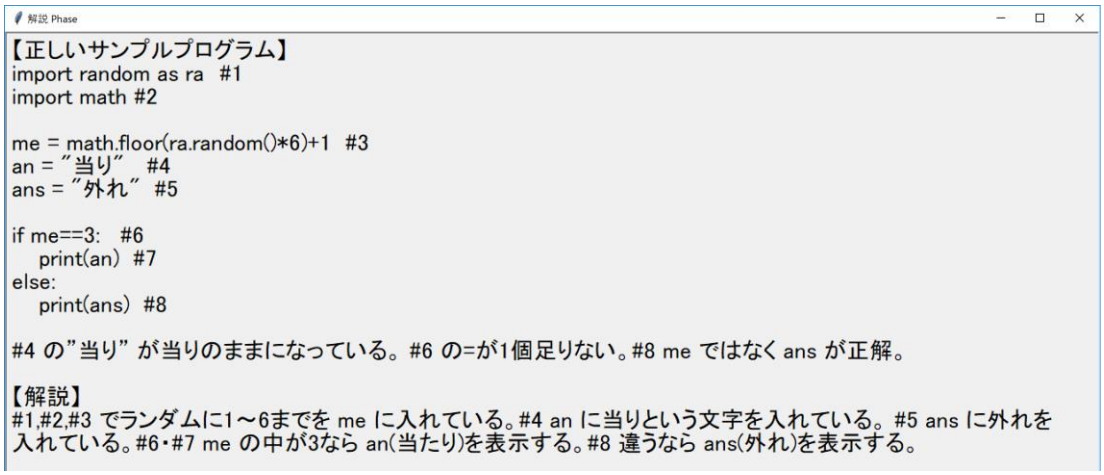

図 3　解説ページの例

Figure 3.　An example of a commentary page

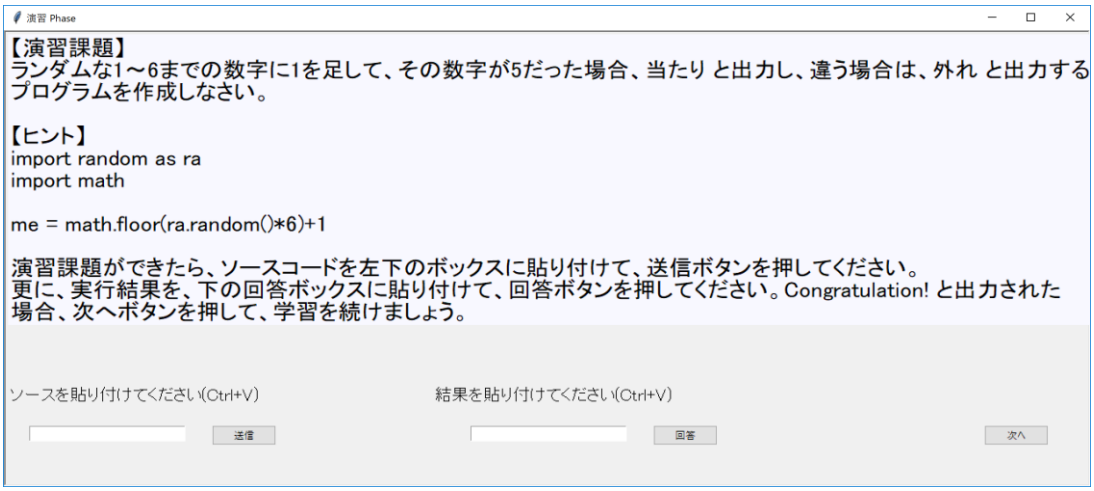

図 4　演習ページの例

Figure 4.　An example of an exercise page

5.1 実験手順

両グループとも，事前準備として，Python3 でのプログラミング経験のアンケートを行った．結果，被験者全員，Python3 でのプログラミング経験はないと述べた．アンケート終了後，BugTutor および比較実験用システムの利用手順を説明し，学習を行ってもらうよう求めた．学習項目は，変数の計算・if 文・for 文の 3 つである．いずれのグループについても，サンプルプログラムと課題を正しく実行できなければ，次の項目の学習に移れないようにした．学習フェイズでは，時間制限を設けず自由に学習できることとした．被験者には，以下の教示を与えた：

- インターネットでの検索は行わないこと，
- 演習フェイズに移行する前に，PaizaCloud のエディタ内のソースコードは消去すること，
- 演習フェイズで，30 分経過しても課題が解けない場合，自己申告すること．

学習フェイズ終了後，演習フェイズ移行する．演習フェイズでは，演習課題を行ってもらい，ソースコード，並びに，正しい実行結果を入力することで，次の学習フェイズに移るよう設定した．

以上の学習が終了した 5 日後に，記憶の定着率を測るためのテストを実施した（以降，Delay テスト略す）．Delay テストの内容を図 5 に示す．

2重for文を使用して以下の実行結果と同じになるプログラムを作成しなさい

【実行結果】
1
12
123
1234
12345

【ヒント】
print(変数,end="")を使ってもよい

図 5　Delay テスト内容

Figure 5.　Delay test

Vol.2020-HCI-187 No.12
2020/3/16

5.2 実験結果

利用グループおよび非利用グループのそれぞれについて学習フェイズに要した総時間と演習フェイズに要した総時間の平均と標準偏差を，図 6 に示す．縦軸の単位は分である．学習フェイズに要した総時間の平均について，F 検定を実施した結果，等分散であったため，Student の t 検定（対応なし）をおこなった．結果，利用グループの学習フェイズに要した総時間の平均は，非利用グループよりも有意に長い（$t(10) = 2.3$，$p < .05$）ことが示された．また，演習フェイズ に要した時間について，F 検定を実施した結果，非等分散であったため，Welch の t 検定（対応なし）をおこなった．結果，利用グループの演習フェイズに要した総時間の平均は，非利用グループよりも有意に短いことが示された（$t(7) = 2.9$，$p < .05$）．よって学習フェイズに要した時間は利用グループが非利用グループよりも有意に長く，逆に演習フェイズに要した時間は利用グループの方が非利用グループよりも有意に短いという結果が得られた．

なお，演習フェイズで，演習課題を達成できず自己申告した被験者は，利用グループでは確認されず，非利用グループでは被験者 H，L の 2 名であった．被験者 H が達成できなかった課題は，変数の計算と for 文で，被験者 L が達成できなかった課題は，for 文であった．実験後，両名にインタビューを行った．被験者 H は，変数計算のサンプルプログラムを動かし，すぐ演習課題に進んだ結果，書き方を覚えていなかったと述べた．また for 文の課題については，for 文の使い方を覚えたと思い課題に臨んだが，for 文で使用する range の書き方を忘れてしまったと述べた．被験者 L は，for 文の in を思い出せなかったと述べた．

Delay テストの結果を表 1 に示す．利用グループの被験者 6 名のうち 5 名が課題を達成することができたのに対し，非利用グループの被験者は 6 名のうち 1 名しか課題を達成することができなかった．システム利用に Delay テストの結果が関係しているかについてカイ 2 乗検定で検証したところ，システム利用の関係性に有意な差が示された（$\chi^2(1) = 5.3$，$p < .05$）．

変数の計算学習以降，利用グループの被験者には，次のような同一の傾向が多くみられた．学習フェイズ中，サンプルプログラムにエラーが発生した際，解説の閲覧に頼らず，自身で試行錯誤し，プログラムを作成した．試行錯誤してもサンプルプログラムが動かない場合のみ，解説を閲覧していた．

学習フェイズ中に，与えられたサンプルプログラム以外のコードを書き，自発的に追加学習した被験者は

- 利用グループ： 被験者 A，B，D，E，F
- 非利用グループ： 被験者 G

であった．具体的に作成したプログラムは，変数の計算学習中に，サンプルにはない足し算，引き算の実行などであった．

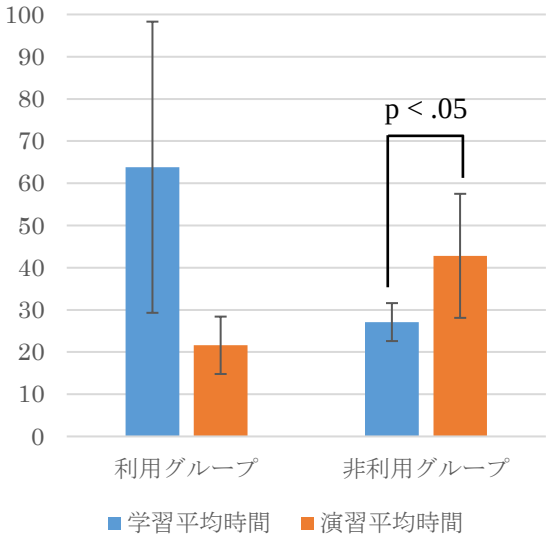

図 6 実験結果
Figure 6. Results of the experiments

表 1 Delay テスト結果
Table 1. Results of the delay test

	利用グループ	非利用グループ
達成人数	5	1
非達成人数	1	5
合計	6	6

6. 考察

非利用グループと比較して，利用グループにおいて学習時間が長い理由として，学習フェイズにおいてサンプルプログラムが実行できない場合，演習フェイズに移れないため，サンプルプログラムの修正に時間を取られたためと考えられる．さらに，規定のサンプルプログラム以外にコードを実行する行為があったため，学習時間がより長くなったと推測する．次に，演習時間が，非利用グループに対して利用グループの方が短い理由として，学習フェイズにおいて，サンプルプログラムの修正を実行することによって，能動的な学習に転換され，学習項目（if 文や for 文など）の理解が深まり，演習課題を解く効率が向上したと考えられる．表 1 に示した Delay テストの結果より，BugTutor の利用者は，プログラミング学習において，記憶定着率が向上したと考えられる．この結果については，前述した学習フェイズの学習方法によって差が出たと推測する．

非利用グループの G，I，J，K は学習フェイズにおいて，サンプルプログラムの入力を誤って，インデントやスペルミス，スペル忘れ（if 文で使用する，コロン忘れなど）な

どのエラーを発生させていた．このエラーを修正する作業を通じて，この 4 名の被験者はある程度の能動的学習を達成できていたものと思われる．ただし，その誤りは，必ずしも各学習課題における本質的な内容に関わるものではない．このため，学習課題における本質的な内容に関する誤りが埋め込まれていた利用グループに比べて，知識の定着率が低くなったものと思われる．さらに，演習課題を達成できなかった非利用グループの被験者 H，L は，どちらもサンプルプログラムを正しく入力できため，エラーを発生させなかった．この結果，被験者 H，L は，他の被験者よりも学習形態が受動的なものとなってしまい，記憶が十分定着せず，演習課題を達成することができなかったものと推測される．

なお，非利用グループの被験者 G は，学習時間と，演習課題において利用グループと同等の結果を残している．この被験者は，自発的にサンプルコード以外のプログラムを作成して動かし，声に出しながら学習を行っていたことが観察された．つまりこの被験者は，自発的に能動的な学習を行うことができていたと考えられる．

実験終了後，実験を行っての感想，もしくはシステムを使ってみての感想をお聞かせください，というアンケートに対して，利用グループの被験者 E は，「この実験に対しての感想になるかわからないが，これから仕事で使う可能性があるので，プログラムの学習をする際には，答えをみないで間違えながら勉強しようと思った．理由は，5 日後のテストを始める際には，覚えているわけがないと思ったが，手を動かしているうちに内容を思い出すことができ，身についているのではないかと思ったから.」と回答した．間違えながら勉強しようという学習方法は，能動的な学習ととらえることができる．そのため，被験者 E は能動的な学習を実行できたため，Delay テストが達成できたのではないかと主張している．

以上の結果から，従来は正しいサンプルプログラムを提供するのが当たり前であったプログラミング学習教材において，あえてサンプルプログラムに学習内容の本質に関わるバグを埋め込んでおくことによって能動的学習を誘発でき，それによって知識の定着率を向上させられることが示された．

7. おわりに

本研究では，既存のプログラミング学習の問題である受動的な学習を，サンプルプログラムに最初からバグを埋め込んでおくことによって能動的な学習に転換するプログラミング学習支援手法を提案した．この手法に基づいて構築した e-learning システム BugTutor を用いたユーザスタディを利用グループ 6 名，非利用グループ 6 名，合計 12 名で実施し，その効果や影響を調査した．結果，提案手法を利用することで，プログラミング初学者の学習が能動的なものになると共に学習量が増加し，記憶定着率が向上する可能性が示された．

今後は，更に被験者を増やし，実験を行うことで，提案手法の有効性に関するより精密な検証を進めたい．また BugTutor については，エディタと実行環境を追加した構築が考えられる．前述したエディタを搭載することにより，PaizaCloud を使用しないため，システム内で学習が完結する．これにより，ユーザへの負担を軽減できると考えられる．さらに，BugTutor について，被験者から GUI デザインの改善や，ソースコードの送信後に送信されたかどうかを判断できるフィードバック機能の追加などが求められた．これらの機能を追加し，より実用的な学習支援システムとしていきたい．最後に，エラーの難易度を変更し，誤情報の質の関係性に関して検証を行いたい．

謝辞 本研究関わってくださったすべての方々に，心より感謝いたします．本研究は科研費（18H03483）の助成を受けたものです．

参考文献

[1] 経済産業省，商務情報政策局，情報処理振興課．IT 分野について．2016
[2] 杉山　成，辻　義人．アクティブラーニングの学習効果に関する研究．小樽商科大学人文化研究 第 127 輯　p61-74，2014.
[3] 枝川　義邦，谷　益美，佐藤　哲也．アクティブラーニングが知識学習に与える影響と実施に向けた課題．早稲田大学高等研究所紀要　第 8 号，p129-140，2015.
[4] 西田　知博，原田　章，中村　亮太，宮本　友介，松浦　敏雄．初学者用プログラミング学習環境 PEN の実装と評価．情報処理学会論文誌，Vol 48，No.8，2007.
[5] 井垣　宏，斎藤　俊，井上　亮文，中井　亮太，楠本　真二．プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PV の提案．情報処理学会論文誌，Vol 54，No.1，330-339，2013.
[6] 田口　浩，糸賀　裕弥，毛利　公一，山本　哲男，島川　博光．個々の学習者の理解状況と学習意欲に合わせたプログラミング教育支援．情報処理学会論文，Vol48，No.2，p958-968，2007.
[7] 穂屋下　茂，学部教育における e ラーニングの利用と価値，メディア教育研究，第 1 巻，第 1 号，p31-43，2004.
[8] 植野　真臣，大学-高専における e-ラーニングによる授業実践，日本教育工学会論文誌，p417-426，2003.
[9] Deanne Adams，Bruce M. McLaren，Kelley Durkin，Richard E. Mayer，Bethany Rittle-Johnson，Seiji Isotani，Martin van Velsen，Erroneous Examples Versus Problem Solving: Can We Improve How Middle School Students Learn Decimals，Proceeding of the 34th Meeting of the Cognitive Science Society，p1260-1265，2012.
[10] 知見　邦彦，櫨山　淳雄，宮寺　庸造，失敗知識を利用したプログラミング学習環境の構築，電子情報通信学会論文誌，D-1，J88_DI（1）:66-75，2005.