

# プログラミングの写経型学習の欠点を補う翻訳学習法の提案

平澤 巧望 篠埜 功

近年、高等学校でのプログラミング必修化に伴い、初学者向けのプログラミング学習支援の重要性が増している。初学者の学習法としてサンプルコードを書き写す写経型学習があるが、サンプルコードをそのまま打ち込むため思考を伴わない単調な作業となり、対象言語の理解が不十分になりがちである。この問題を解決するため、そのまま打ち込むという部分を改善し、考えながら書き写すことができるように、別の言語へ翻訳しながら写経を行う翻訳学習法を提案する。芝浦工業大学 1 年生 13 名を対象に、翻訳学習グループと写経型学習グループに分けて実験を行った結果、翻訳グループはプログラミング知識とプログラミング能力が共に有意に向上し、写経グループはプログラミング能力のみが有意に向上した。ただし、プログラミング知識とプログラミング能力のどちらにおいても、得点の上昇幅の平均値について 2 つのグループ間で有意差はなかった。

## 1 はじめに

近年、日本では、学習指導要領の改訂による高等学校でのプログラミング教育必修化[14][15] やそれに伴う大学入学共通テスト（以下では共通テストと呼ぶ）でのプログラミング出題[12] など、プログラミング学習が重要視されつつあり、プログラミングを学び始める人が増加傾向にある。

プログラミング初学者がよく行う学習法として、サンプルコードを打ち込んで実行し、結果を確認する写経型学習がある。5.1 節で述べるように、写経型学習を行うと対象言語の初学者のプログラミング能力が向上する傾向が見られるが、著者らの知る限り、写経型学習のみでプログラミング能力が有意に向上することはこれまで示されていない。写経型学習は、サ

ンプルコードをそのまま打ち込むため思考を伴わない単調な作業となり、対象言語の構文や意味についての理解が不十分になりがちである[5]。現在では写経型学習は例えば C や Java など、英語のアルファベットが主体の言語で行われることが多いと考えられ、また、写経型学習の効果に関する研究も同様である。

日本で用いられている学習用の言語では、日本語の文字が主体となる言語がいくつかあり、これらに関する学習効果やプログラム理解に関する研究がこれまで行われてきている。例えば、大部分が日本語のひらがなと漢字で書ける独自言語によるプログラミング学習環境 PEN を用いて学習した結果、JavaScript で学習した場合と比べて、日本人にとって学習効果が有意に高いことを示した研究[10] や、大部分が英語のアルファベットで書かれたプログラムと比べて、大部分が日本語のひらがなと漢字で書かれたプログラムの方が日本人にとって早く理解できることを示した研究[11] がある。また、情報に関する日本で実施される試験で用いられる言語で日本語の文字が主体となる言語もある。例えば、共通テストの科目「情報」のプログラミングに関する問題の出題において、高等学校の授業で使用されるプログラミング言語が様々であることから、ある特定の言語でプログラミングを学んでい

\* An Approach to Learning by Translation to Address the Shortcomings of Shakyo-Style Learning in Programming

This is an unrefereed paper. Copyrights belong to the Authors.

Takumi Hirasawsa, Isao Sasano, 芝浦工業大学大学院理工学研究科電気電子情報工学専攻, Department of Electrical Engineering and Computer Science, Graduate School of Engineering and Science, Shibaura Institute of Technology.

なくても全ての受験生がプログラムを読めるようにするために、日本語のひらがなと漢字を用いた共通テスト用の言語（以下では DNCL2 と呼ぶ）が用意されることになっている [13]. 高等学校の授業で用いられることがある Python 等の言語では英語のアルファベットが主体となっているが、Python 等と比べると DNCL2 は大部分が日本語のひらがなと漢字で構成されているため、日本人のプログラミング初学者が読んでも理解しやすいと考えられる。

そこで、本研究では、写経型学習が単調になることによりプログラミング能力の向上が阻害されていると考え、これを解決するために、サンプルコードをそのまま打ち込むのではなく、考えながら書き写すことができるように、別の言語へ翻訳しながら写経を行う学習法を提案する。この学習法を翻訳学習法と呼ぶことにする。本論文では、日本人を対象とし、例として DNCL2 から Python への書き換えについて扱う。

提案手法の有効性を評価するために、芝浦工業大学工学部の 1 年生およびシステム理工学部の 1 年生合計 13 名を対象として、Python のサンプルコードをそのまま写経するグループと、DNCL2 のサンプルコードを Python のコードに書き換えるグループの 2 グループに分けて演習を行い、それらの学習効果を比較する実験を行う。

実験の結果、各グループにおいて学習前後でプログラミング知識およびプログラミング能力の得点が学習の前後で有意に上昇したかどうかについて Wilcoxon の符号順位検定を行った結果、プログラミング知識に関して、翻訳グループは有意に得点が増したが、写経グループについては有意差がなかった。一方で、プログラミング能力に関しては、翻訳グループと写経グループのどちらにおいても有意に得点が増していた。なお、プログラミング知識とプログラミング能力それぞれについて、各グループにおける得点の上昇幅の平均値に有意な差があるか、Mann-Whitney の U 検定を行った結果、どちらにおいても有意差はなかった。

本論文の構成は以下の通りである。まず 2 節で、翻訳学習の方法および学習をサポートするツールについて述べる。次に 3 節で評価実験の内容について述

べ、4 節でその結果を提示し、考察する。さらに 5 節において関連研究について議論する。最後に 6 節でまとめと今後の課題について述べる。

## 2 翻訳学習法および翻訳学習サポートツールの提案

本節では、翻訳学習法を提案し、翻訳学習をサポートするツールを提案する。

### 2.1 翻訳学習法

写経型学習において写経の手本となるサンプルコードに対して、異なるプログラミング言語でサンプルコードと同じ意味のコードを打ち込む学習法を本研究で新たに提案し、これを翻訳学習法と呼ぶ。翻訳学習法の手順を以下に示す。

1. サンプルコードの意味を意識しながら、サンプルコードと同じ意味のプログラムを別の言語に翻訳し、打ち込む。
2. 翻訳が終わったら、実行結果を予想する。
3. 翻訳したプログラムを実行する。プログラム中の値や実行時の入力値を変えて実行結果の変化を見てもよい。2 の予想が間違っていた場合、1 に戻る。

本研究においては、サンプルコード（書き換え元のコード）の言語を DNCL2、実際に打ち込むコード（書き換え先のコード）の言語を Python とした場合について、学習効果の検証を行った。DNCL2 と Python の表記は似ており、DNCL2 プログラムと Python プログラムの対応関係の一部を表 1 に示す。なお、入力を処理する関数について、DNCL2 では【外部からの入力】となっている。これに対し、表 1 中に示した【外部からの整数入力】は、本研究において、被験者やツールの利用者が理解しやすいと考えて関数名を DNCL2 から変更したものである。また、四則演算や比較演算、論理演算に用いる演算子は DNCL2 と Python でほとんど共通であるが、Python で割り算の商を求める演算子が `//` であるのに対し、DNCL2 では `÷` である。

表 1 DNCL2 プログラムと Python プログラムの対応関係

DNCL2	Python
表示する ()	print()
【外部からの整数入力】	int(input())
もし <条件式> ならば :   ...	if <条件式>: ...
そうでなくもし <条件式> ならば :   ...	elif <条件式>: ...
そうでなければ :   ...	else : ...
<条件式> の間繰り返す :   ...	while <条件式>: ...

## ▼ 【2】条件分岐の基本

### 偶数なら「even」、奇数なら「odd」と表示するプログラム

— 条件分岐の書き方と条件式の書き方に着目

```
x = int(input()) # ユーザの入力を整数として代入
もし x%2 == 0 ならば:
    表示する("even")
そうでなければ:
    表示する("odd")
```

```
[ ] # 以下に翻訳してみましょう
```

図 1 ノートブックの記述の 1 例

### 2.2 翻訳学習サポートツール

本研究では、実験のための環境として、Google Colaboratory [1] を使用した。Google Colaboratory は、ブラウザ上のページへ Python コードを記述し、実行できるインタラクティブな環境であり、ノートブックの 1 つである。ノートブックは、Python コードを記述し実行できるコードセル、および画像、HTML、 $\text{\LaTeX}$ などを記述するテキストセルからなる。書き換え元のサンプルコードは、テキストセル内に記述した。ノートブックの記述の 1 例を図 1 に示す。

## 3 評価実験

提案手法のプログラミング初学者に対する有効性を評価するため、芝浦工業大学工学部の 1 年生およびシステム理工学部の 1 年生合計 13 名を対象に、翻訳学習を行う GroupA の 7 名と、写経型学習を行う GroupB の 6 名に分けて、芝浦工業大学において実験を実施した。

実験は、プログラミングの基本事項に関する一斉講義を 20 分間実施した後、プログラミング知識、プログラミング能力に関する試験をした上で、グループ別の学習法により 30 分間のプログラミング演習を行い、再度プログラミング知識、プログラミング能力に関する試験を行った。試験時間は、プログラミング知識に関する試験を 10 分、プログラミング能力に関する試験を 20 分とした。

実験では、Google Colaboratory のノートブックを実験協力者の人数分コピーして配布したり試験の答案を回収したりする作業を簡単にするために、Google が提供している Learning Management System の Google Classroom を使用した。実験協力者には、実験実施前に Google Classroom に作成された実験用のクラスに参加してもらった。実験当日は、各自が持参したノート PC で Google Classroom にアクセスしてプログラミング演習や試験の回答を行ってもらった。

### 3.1 プログラミングの基本事項に関する一斉講義

実験開始時に、一斉講義で使用する講義資料を紙で配布した。この資料に沿って、アルゴリズムの3要素である順次、分岐、反復を実現するための基本的な Python プログラミングの手法について、DNCL2 の表記を使った意味の解説を交えながら、20 分間の一斉講義を行った。講義の内容を以下に示す。なお、「Python プログラムを実行してみる」の部分では、Google Classroom の資料として Google Colaboratory のノートブックを配布し、Python プログラム `print("Hello, world!")` をノートブック上で実際に実行してもらった。

- アルゴリズムとは？
- 順次処理と変数・代入
  - Python における四則演算の書き方
- 条件分岐 `if`
  - Python における条件式の書き方
- 繰り返し `while`
- Python プログラムを実行してみる

### 3.2 学習前試験および学習後試験

本節冒頭で述べたプログラミング知識やプログラミング能力に関する試験を学習前および学習後に同じ方式で行った。なお、各試験問題は、3.1 節で挙げた知識で回答できるように作成し、講義で扱っていない知識を要する場合には注釈を加えた。また、各試験では試験開始直前に問題用紙を配布し、試験時間中は講義資料等の参照を不可とした。

#### 3.2.1 プログラミング知識

プログラミング知識の試験問題には両グループで DNCL2 を用いた。試験問題の解答は、試験開始直前に Google Classroom の課題として配布した Google Forms に入力し、送信してもらった。Google Forms には問題用紙と同じ問題が記載されている。プログラミング知識の試験では、Google Forms および問題用紙に記載されている DNCL2 で書かれたプログラムを読み、その実行結果に関する設問の解答を Google Forms から入力してもらった。プログラミング知識に関する問題のうち2つを図2に示す。試験は7問から

次のプログラムを実行したとき、表示される数字を半角で入力してください。

```
a = 3
b = a * 2
a = b + a * 2
表示する (a)
```

1 から 10 までの整数を全て表示するプログラムを作りたい。次のコードは正しいでしょうか。

```
c = 0
c < 10 の間繰り返す:
| 表示する (c + 1)
```

正しい  
 正しくない

図2 プログラミング知識に関する問題の例

なり、試験時間は10分とし、12点満点で採点した。

#### 3.2.2 プログラミング能力

プログラミング能力の試験問題の回答は、試験開始直前に Google Classroom の課題として各実験協力者にコピーを配布した Google Colaboratory のノートブックに入力してもらい、Google Classroom から提出してもらった。ノートブックには問題用紙と同じ問題が記載されている。プログラミング能力について、Google Colaboratory のノートブックおよび配布した問題用紙に記載されている問題文で指定された Python プログラムを作成してもらった。プログラミング能力に関する問題のうち1つを図3に示す。試験は6問からなり、試験時間は20分とし、10点満点で採点した。

#### 3.3 プログラミング演習

学習前試験の次に、プログラミング演習を行った。プログラミング演習は、Google Classroom の課題として各実験協力者にコピーを配布した Google Colaboratory のノートブックを使用して30分間行った。

### ▼ 問3

ユーザが入力した整数nについて、2のn乗の計算結果を表示するプログラムを作成してください。

```
[ ] n = int(input())
    # ここまで雛形
    # 以下に解答を入力してください
```

図3 プログラミング能力に関する問題の例

GroupAの実験協力者には翻訳学習用、GroupBの実験協力者には写経型学習用のノートブックを配布した。また、プログラミング演習の開始前には、学習の手順およびノートブックと同じサンプルコードが印刷されている演習用資料を紙で配布した。

また、実験協力者には、プログラミング演習開始前に2.1節で述べた翻訳学習の手順もしくは写経型学習の手順について説明し、この手順に沿ってプログラミング演習を行うように指示した。実行結果の予想が合っていた場合は、以下のリストの順番で次のサンプルコードへと進んでいく。

1. 順次処理と変数
2. 条件分岐の基本
3. 繰り返しの基本
4. 条件分岐の応用
5. 繰り返しの応用
6. 順次、分岐、反復の総理解

サンプルコードは、翻訳グループはDNCL2、写経グループはPythonで各項目ごとに1つずつ用意した。

#### 3.4 学習後アンケート

学習後試験の後、4.1.3節で示す設問について、全ての実験協力者にGoogle Classroomの課題として配布したGoogle Formsから回答してもらった。

## 4 実験結果および考察

本節では、実験結果および学習後アンケートの結果を示し、考察を行う。

### 4.1 実験結果

ここでは、プログラミング知識、プログラミング能力に関する実験結果および学習後アンケートの結果を示す。

#### 4.1.1 プログラミング知識

図4にプログラミング知識に関する試験問題の学習前後での各実験協力者の得点の変化を示す。また、表2に各実験協力者における得点の上昇幅および各グループにおける平均上昇幅を示す。

図4および表2に示す通り、得点が下がった実験協力者、得点が変わらなかった実験協力者がそれぞれ見受けられるが、得点の平均上昇幅は両グループともに正の値となった。この得点の上昇が有意であるかを確認するため、GroupAとGroupBそれぞれの実験協力者の学習前と学習後の得点に対して有意水準5%でWilcoxonの符号順位和検定を行ったところ、p値はGroupAにおいて0.0292、GroupBにおいて0.2071であり、翻訳学習を行ったGroupAは有意差があった。

また、GroupBよりもGroupAの方が得点の平均上昇幅が大きく、この差が有意なものであるかを確認するため、表2のデータに対して有意水準5%でMann-WhitneyのU検定を行ったところ、p値は

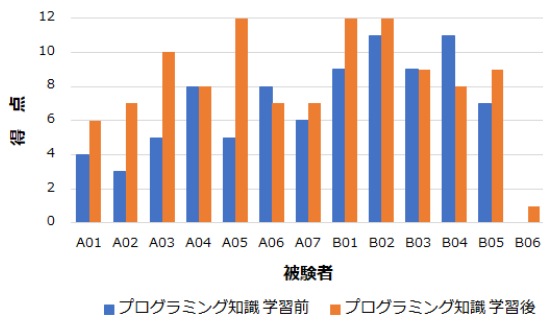


図4 実験協力者ごとの学習前後のプログラミング知識に関する得点の比較

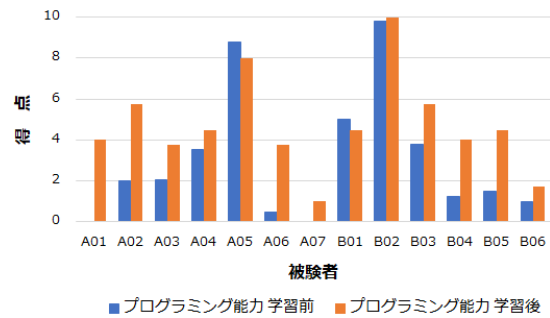


図5 実験協力者ごとの学習前後のプログラミング能力に関する得点の比較

0.1746 であり、有意差はなかった。

#### 4.1.2 プログラミング能力

図5にプログラミング能力に関する試験問題の学習前後での各実験協力者の得点の変化を示す。また、表3に各実験協力者における得点の上昇幅および各グループにおける平均上昇幅を示す。

図5および表3に示す通り、得点が下がった実験協力者や得点が変わらなかった実験協力者がそれぞれ見受けられるが、得点の平均上昇幅は両グループともに正の値となった。この得点の上昇が有意なものであるかを確認するため、GroupA と GroupB それぞれの実験協力者の学習前と学習後の得点に関して有意水準5%で Wilcoxon の符号順位和検定を行ったところ、p 値は GroupA において 0.0140、GroupB において 0.0374 であり、両グループともに有意差があった。

また、GroupB よりも GroupA の方が得点の平均上昇幅が大きく、この差が有意なものであるかを確認するため、表3のデータに対して有意水準5%で Mann-Whitney の U 検定を行ったところ、p 値は 0.2226 であり、有意差はなかった。

#### 4.1.3 アンケート結果

3.4 節で述べたアンケートについて、No.1~No.8 の質問項目および各グループにおける回答の平均値と GroupA の平均値 - GroupB の平均値 によって求められる差を表4に示す。また、設問 No.9 「サンプルコードを基に自分でプログラムを打ち込む学習法におけるサンプルコードの内容について、学習に役立つと思うのはどちらですか？」に対する回答をグループ毎、および実験協力者全体について図6に

示す。なお、No.1, No.2, No.5~No.8 は「とてもそう思わない」を1、「とてもそう思う」を4とした1~4の4段階、No.3 と No.4 は「とても難しかった」を1、「とても簡単だった」を4とした1~4の4段階、No.9 は「～の間繰り返すように日本語で書かれている」を1、「while のようにローマ字でプログラミング言語で書かれている」を4とした1~4の4段階で回答してもらった。

表4を見ると、設問7「この学習法ならプログラミング学習を積極的に実施できそうだ」に対して差が大きく開いており、翻訳学習を行った GroupA の方が数値が高い。この差が有意なものであるかを確認するため、回答データに対して有意水準5%で Mann-Whitney の U 検定を行ったところ、p 値は 0.0380 であり、有意差があった。なお、No.1~No.9 の他の質問項目についても同様の検定を行ったところ、どれも有意差がなかった。

以上で示した選択式アンケートに加えて、自由記述式の設問において、学習法の良かった点と良くなかった点および改善すべき点について回答してもらった。翻訳学習を行った GroupA の実験協力者からは、コードの意味を理解しながら演習を進めやすい点が優れているが、演習時間がコードを覚えきるのに不十分であったという意見が寄せられた。一方で、写経型学習を行った GroupB の実験協力者からは、写経する過程で自分が勘違いしているところに気づいたり修正できたりする点が優れているが、サンプルコードを書き写すだけの作業になってしまい分からなくても実行できてしまうことでできた気になってしまう点が

表 2 実験協力者ごとのプログラミング知識の得点の上昇幅

グループ	各実験協力者における得点の上昇幅							平均値
GroupA	2	4	5	0	7	-1	1	2.57
GroupB	3	1	0	-3	2	1		0.67

表 3 実験協力者ごとのプログラミング能力の得点の上昇幅

グループ	各実験協力者における得点の上昇幅							平均値
GroupA	4	3.75	1.7	0.95	-0.8	3.25	1	1.98
GroupB	-0.5	0.2	1.95	2.75	3	0.75		1.36

表 4 アンケートの質問項目および各グループの回答の平均値

No.	質問項目	GroupA	GroupB	平均値の差
1	プログラミングへの自信がつきそうですか？	2.86	3.33	-0.48
2	本日使用した学習ツールによってプログラミングに対する以下の諸課題は解決しそうですか？			
2a	自分が解決したい問題をどのようなアルゴリズムで解決できるか分からない	2.71	3.17	-0.45
2b	アルゴリズムが与えられていてもそれをプログラムで記述するのが難しい	2.57	3.00	-0.43
2c	プログラミングに触れる量が少なく、上達できない	2.29	2.83	-0.55
2d	print や while など、ローマ字で書かれた命令を覚えられない	2.86	3.00	-0.14
3	プログラミング演習の学習法が難しかった	2.29	2.33	-0.05
4	プログラミング演習の際、サンプルコードを読むのが難しかった	3.29	3.00	0.29
5	打ち込んでいるコードの意味を理解しながらプログラミングできた	3.43	2.83	0.60
6	この学習法なら学習がスムーズに進みそうだ	3.43	3.00	0.43
7	この学習法ならプログラミング学習を積極的に実施できそうだ	3.86	3.33	0.52
8	このツールを使うことはプログラミングへの苦手意識の克服に役立ちそうだ	3.29	3.50	-0.21

問題だという意見が寄せられた。

#### 4.2 考察

4.1 節で示したように、プログラミング知識の得点に関して、翻訳グループは学習前と比べ学習後は有意に上昇したが、写経グループには学習前後で有意差はなかった。一方で、プログラミング能力の得点に関しては、翻訳グループと写経グループの双方で有意に上昇した。また、プログラミング知識とプログラミング能力のどちらにおいても、各グループにおける得点の上昇幅の平均値には有意差はなかった。また、アンケート結果からは、翻訳グループの方が学習に積極的に取り組めそうだと有意に感じていることがわかった。

今後の課題として、今回の実験では少数の実験協力者に対して、30 分のみ学習時間で学習効果の評価を行ったため、実際の授業やプログラミング講座のように時間をかけて長期的に学習を行った場合の効果に違いがあるかを調べる事が挙げられる。

#### 5 関連研究

これまで、初学者に対するプログラミング学習について、写経型学習の学習効果や日本語文字を取り入れたプログラミング言語による日本人への学習効果、別のプログラミング言語へ移行する際の課題などについて様々な研究が行われてきた。ここでは、これらの観点から主要な研究について概観する。まず 5.1 節において、写経型学習の有効性と問題点およびそ

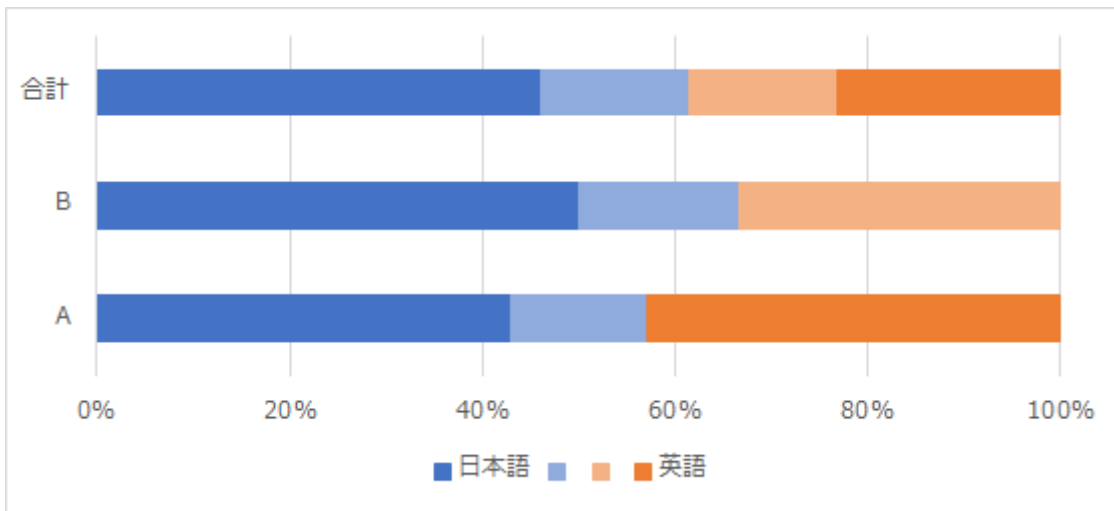


図6 設問 No.9「サンプルコードを基に自分でプログラムを打ち込む学習法におけるサンプルコードの内容について、学習に役立ちそうだと思うのはどちらですか？」に対する回答

の解決策について述べる。次に、5.2節では、日本語文字を使用して記述したプログラムによるプログラミング学習を提案した研究について述べる。最後に、5.3節では、あるプログラミング言語のプログラムを別のプログラミング言語のプログラムに書き換える学習法あるいは言語の移行に関し、その効果や課題について述べる。

### 5.1 写経型学習の有効性と問題点

菊池らによる研究[6]では、普段プログラミングを行うグループと行わないグループそれぞれにアルゴリズム理解課題とアルゴリズム実装課題を行わせ、プログラミングの経験量と開発力の関連について調べた。この結果、普段のプログラミング経験は開発力への影響が明らかに見られるが、アルゴリズム自体の理解力にはプログラミング経験での有意差は認められず、一定以上のプログラミング経験があるとアルゴリズムを理解して実装することが可能である。

岡本らによる研究[4]では、プログラミング初学者の写経型学習における問題点を作業遂行上の躓きと作業を介した理解に関する躓きに分類し、それぞれに対してコンパイル手順やエラー発見方法の手順が書かれており、さらに学習目的や学習方法を明示されている教材を開発した。この教材を用いた実験の結果、

写経型学習において、学習目的を明示することと実行に至るまでの作業が滞りなく進むよう丁寧な支援を行うことによって、躓きが多発するfor文に関して、8割以上の学生が理解した。

亀井らによる研究[5]では、写経型学習において構文や意味といったプログラミング知識の理解が不十分になりがちであるという欠点を補うために、写経型学習においてコードに微小な変更を加える摂動による学習法を提案した。彼らは、提案手法の例として、二項演算子に摂動を加えるプログラミング学習ツールを開発し、初学者を対象とした評価実験を行った。この結果、演習問題を解く学習法を行ったグループよりも、写経型学習に加えて摂動に基づく学習法を行ったグループの方が、学習効果が向上したことを示した。

我々が過去に行った研究[16]では、プログラミング初学者である高校生の多くはキーの位置を覚えていないことに着目し、プログラミング初学者の写経型学習の効果向上を目的として、次に押すべきキーを光らせながら写経を行う学習法を提案した。この学習法を実装したツールによる実験の結果、次に押すべきキーを光らせることで、光らせない場合と比べ、写経型学習のプログラミング能力向上の効果が有意に高いことが示された。このことから、初学者がタイピングに対する課題を抱えていることが写経型学習



のプログラミング能力を向上させる効果を抑制しており、写経型学習には一定のタイピング能力が必要であった。しかし、亀井らの研究[5]および我々の過去の研究[16]において、写経型学習自体の有効性については調査されていない。

上記のように、摂動や光るキーボードなど、写経と何かを組み合わせることで単なる写経と比べて学習効果が向上することがこれまで示されてきた。本研究は、翻訳しながら写経する新たな学習法を提案したものである。

## 5.2 日本語文字を用いたプログラミング学習に関してさまざまな観点から検討を行った研究

中川ら[11]は、C言語において識別子等に日本語を使用する場合と使用しない場合で読解力を比較する実験を行った。この実験の結果、日本語の方が英語の2/3程度の所要時間で読解できることが示された。

次に、西田ら[10]は、DNCLに拡張を加えた日本語ベースの独自言語によるプログラミング学習環境PENを作成した。この環境を大学の授業で用いた結果、JavaScriptで学習した場合と比べ、プログラミングの理解度の自己評価と試験の成績の双方で有意に高くなった。

また、高野らの研究[8]では、中高生にプログラミングの楽しさを学んでもらうために、自然言語を用いたプログラミング言語Samoyedを提案し、プログラムの動作原理が理解できるような言語処理系の構想を示したが、自然言語の曖昧性による問題が解決されていない。

さらに、西村らの研究[9]では、中学、高校の情報分野の授業で記述させるフローチャートだけではC言語のプログラムを記述するための情報が不足していると考え、C言語のプログラム断片を選択肢から選択することでプログラムを書き、さらに日本語の文との対応が見ることができるとする学習支援システムを開発した。

上記の研究においては、日本語文字を使用して記述するプログラミング言語の検討や開発がなされており、また、従来のプログラミング言語による学習と比べて学習効果が高いことも示されているが、大部分が

英字で書かれている書かれている別のプログラミング言語から書き換えることの学習効果については調べられていない。これに対し、本研究では、DNCL2からPythonへ書き換える学習を例として学習効果を調査しており、この点においてこれらの研究と異なる。

## 5.3 別のプログラミング言語への書き換えや移行に関して検討を行った研究

間辺らの研究[7]では、学習指導要領の改訂によって、生徒がプログラミングの学習経験を持った状態で高等学校に入学してくることを考慮し、Cの学習経験を持つ生徒に対してPythonを導入する授業の例として、CからPythonへのプログラム書き換えによる方法を示した。この方式で授業を行った結果、約8割の生徒が難しさを感じた。間辺らは、その要因として、既習のアルゴリズムの理解が不十分であることや、CからPythonに書き換えるための思考とアルゴリズムをPythonで記述するための思考とが混在してしまったことを挙げている。一方で、この学習によって、PythonとCでは表記が異なるもののアルゴリズムは同じであることに気づかせる効果が期待できると述べている。

また、Nischalらの研究[2]では、ある言語を理解している職業プログラマーが既習の知識を活用することで、別の言語を簡単に習得することができるのかを明らかにするために、既習の知識が新しい言語の学習を妨げるかどうかについて調査した。この調査の結果、既習の言語の構文などが新しい言語における学習内容に干渉し、混同してしまうなどの事例が多数見られた。

さらに、Ethelらの研究[3]では、学生がプログラミング教育の過程において言語の移行に関して課題を抱えているということに着目し、プログラミング初学者の学生が新しい言語を学ぶ際に、変数への代入やwhileループ、メソッドといった構成要素の理解がどのように深まるかを明らかにするために、命令型のPythonからオブジェクト指向型のJavaに移行する学生と、オブジェクト指向型のJavaから命令型のPythonに移行する学生の学習について比較した。この結果、学生は、JavaとPythonで構文が似ている

が意味が違う場合や、構文は異なるが意味は同じ場合に苦勞していた。

上記の研究は、ある言語のプログラムを別の言語のプログラムに書き換える際に、CやJavaからPythonといったような言語の構成要素が少し異なる言語間での移行について調査した。これに対し本研究では、言語の構成要素がほぼ同じであるDNCL2からPythonへの書き換えを対象としており、上記の研究で指摘された表記の違いによる混乱が起りにくくなると考える。

## 6 まとめと今後の課題

本研究では、プログラミング初学者を対象に、写経型学習の単調性を改善するために、別の言語へ翻訳しながら写経を行う翻訳学習法を提案した。提案手法の有効性を評価するため、書き換え元のプログラムがDNCL2、書き換え先のプログラムがPythonである場合を例として、翻訳学習を行う場合と、Pythonの写経型学習を行う場合でプログラミング知識やプログラミング能力について比較実験を行った。その結果、プログラミング能力の得点に関して、翻訳グループと写経グループともに有意に上昇した。プログラミング能力の得点に関しては、翻訳グループは有意に上昇したが、写経グループには有意差がなかった。また、プログラミング知識とプログラミング能力のどちらにおいても、各グループにおける得点の上昇幅の平均値には有意差はなかった。これらのことから、翻訳学習と写経型学習のどちらが学習効果が高いかについて示すことはできなかった。今回は短時間の実験であったが、長期的な学習で効果に違いがあるかを調査することが今後の課題として考えられる。

### 謝辞

本研究を進めるにあたり、評価実験にご協力いただいた芝浦工業大学工学部およびシステム理工学部の2024年度入学の1年生13名の学生の皆様に感謝申し上げます。

### 参考文献

[1] Google: Colaboratoryへようこそ - Colab, <https://colab.research.google.com/>. 2024年6月21日

参照。

- [2] Shrestha, N., Botta, C., Barik, T., and Parnin, C.: Here we go again: why is it difficult for developers to learn another programming language?, *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, ICSE '20*, 2020, pp. 691–701.
- [3] Tshukudu, E. and Cutts, Q.: Understanding Conceptual Transfer for Students Learning New Programming Languages, *Proceedings of the 2020 ACM Conference on International Computing Education Research, ICER '20*, 2020, pp. 227–237.
- [4] 岡本雅子, 村上正行, 吉川直人, 喜多一: プログラミングの写経型学習過程を対象としたつまづきの分析とテキスト教材の改善: 作業の自立的遂行と作業を介した理解のための支援と工夫, *京都大学高等教育研究*, Vol. 19(2013), pp. 47–57.
- [5] 亀井亮汰, 吉塚大浩, 篠埜功, 古宮誠一: 写経型学習の欠点を補う摂動を用いた理解度確認問題生成手法 — 二項演算子の事例に基づく有効性評価, *コンピュータソフトウェア*, Vol. 38, No. 1(2021), pp. 111–139.
- [6] 菊池智, 濱本和彦: プログラミングの経験量が開発力に与える影響に関する研究, *人工知能学会研究会 第73回 先進的学習科学と工学研究会*, 2015, pp. 31–35.
- [7] 間辺広樹, 長島和平, 並木美太郎, 長慎也, 兼宗進: Cの学習経験を持つ高校生へのPythonの授業導入事例, *情報教育シンポジウム 論文集 2019*, 2019, pp. 256–262.
- [8] 高野志歩, 田村みゆ, 富岡真由, 秋信有花, 倉光君郎: 擬似コードから考える自然言語を活かしたプログラミング言語, *情報教育シンポジウム 論文集 2021*, 2021, pp. 147–151.
- [9] 西村宗一郎, 花田理, 倉山めぐみ: 日本語とC言語をつなぐプログラミング学習支援システムの提案, *電子情報通信学会 教育工学研究会*, 2018, pp. 39–42. 2018-11-ET.
- [10] 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), pp. 2736–2747.
- [11] 中川正樹, 早川栄, 玉木裕一, 曾谷俊男: 日本語プログラミングの実践とその効果, *情報処理学会論文誌*, Vol. 35, No. 10(1994), pp. 2170–2179.
- [12] 独立行政法人大学入試センター: 平成30年告示高等学校学習指導要領に対応した令和7年度大学入学共通テストからの出題教科・科目について, <https://www.dnc.ac.jp/albums/abm.php?d=510&f=abm00000301.pdf>, 2021. 2024年6月17日参照.
- [13] 独立行政法人大学入試センター: 令和7年度大学入学共通テスト 試作問題「情報」の概要, <https://www.dnc.ac.jp/albums/abm.php?d=511&f=abm000003141.pdf>, 2021. 2024年6月17日参照.
- [14] 文部科学省: 高等学校学習指導要領(平成30年告示)解説 情報編, [https://www.mext.go.jp/content/1407073\\_11\\_1\\_2.pdf](https://www.mext.go.jp/content/1407073_11_1_2.pdf), 2018. 2023年5月16日参照.
- [15] 文部科学省: 小学校プログラミング教育に関する概要資料, [https://www.mext.go.jp/a\\_menu/shotou/zyuhou/detail/1416328.htm](https://www.mext.go.jp/a_menu/shotou/zyuhou/detail/1416328.htm), 2018. 2023年5月

16 日参照.  
[16] 平澤巧望, 篠埜功: 光るキーボードを用いた写経型

プログラミング学習の支援, 情報処理学会論文誌 プログ  
ラミング, Vol. 16, No. 2(2023), pp. 1-27.