

シナリオ作成を用いたプログラミング導入教育の試み

新開純子*1, 早勢欣和*1, 宮地功*1

*1 富山高等専門学校

A Trial of Introductory Programming Education with Scenario Making Procedure

Junko Shinkai*1, Yoshikazu Hayase*1, Isao Miyaji*1

*1 National Institute of Technology, Toyama College

論理的に考え、プログラムとして記述可能なアルゴリズム構築能力を育成するためには、プログラム言語を用いたプログラミング教育の前段階の効果的なプログラミング導入教育が必要であると考えられる。本研究では、アルゴリズム構築能力を育成するためのプログラミング導入教育として、日本語のシナリオ作成を行うことを提案する。本稿では、シナリオ作成を用いたプログラミング導入教育の内容と実践後のアンケート調査の結果を報告する。

キーワード: プログラミング教育, 導入教育, アルゴリズム構築, 論理的思考

1. はじめに

情報系学科のプログラミング教育は、与えられた課題を理解・分析して、その課題の解決方法を考え、解決手順を詳細化して、プログラム言語で記述可能なアルゴリズムを組み立てる問題解決能力を育成することを目的にした重要な基礎教育である。そこで、筆者らは、アルゴリズム作成までのプロセスを重視する教育や、学生同士の評価活動やeラーニングなどをブレンドしたプログラミング教育を実践した⁽⁴⁾。さらに、手作業による処理を取り入れた体験型アルゴリズム学習も実践した⁽⁵⁾。

しかし、プログラミング教育の初期段階で難しく感じ、学習意欲を失う学習者も少なくない。これは、変数や代入の概念や、論理的にアルゴリズムを作成することも初めてのことであり、戸惑いを感じ、プログラミングを難しく感じているためであると考えた。

そこで、本研究ではプログラミングやアルゴリズム構築へのハードルを低くする導入教育として、論理的にあいまいさのない日本語のシナリオを書くことを提案する。

先行研究として、楠ら⁽¹⁾は、アルゴリズムの作成のみを構造化チャート(PAD)を用いて実践をして、そ

の効果を報告している。また、杉浦ら⁽³⁾は、手作業によるアルゴリズムをそのまま実行可能なプログラムとして記述できる言語「ことだま on Squeak」を使用した教育方法を報告している。これらの導入教育はプログラム言語を用いた教育の前に、アルゴリズムの構築に注力できる環境を提供して、アルゴリズム構築能力を育成する導入教育である。論理的に考えることに慣れていない学習者にとっては、これらの環境もまだまだハードルが高く、変数や代入という概念の導入を考える段階までに至ってはいない。

本稿では、日本語のシナリオ作成を用いたプログラミング導入教育の内容を述べる。さらに、導入教育の実践後にアンケート調査を行った結果について報告する。

2. プログラミング導入教育

2.1 プログラミング導入教育の必要性

プログラミングの教科書に最初に書かれている変数、型宣言、代入などといった概念は、コンピュータの仕組みと関連している。これらは、プログラムの書式やプログラム言語の文法と一緒に教えられるため、プログラミングの初学者にとっては、よくわからないまま

学習がスタートする。

さらに、人間が特に意識しないで行っている 2 枚のカードを交換する作業も、コンピュータで処理するときは、2 つの変数 a, b の値を交換するプログラムとなる。このとき、 $a=b; b=a;$ ではなく、別の変数 w を用意して、 $w=a; a=b; b=w;$ のように処理を行う。このように、人間が簡単に行えることとコンピュータに行わせることの違いは、プログラミングは面倒で難しいものというイメージを与えてしまう。

そこで、変数や代入というプログラムで必要となる概念とコンピュータで処理するときの手順の考え方を体験する導入教育が必要である。

2.2 プログラミング導入教育のための教材

プログラミング導入教育を実施するために、以下の 4 つの経験を行う教材が必要であると考えます。

(1) システムを開発する立場で考える経験

プログラミング教育は、システム開発への最初の一步である。そのため、導入教育では、学習者にシステム開発を行うための学習が始まることを意識させる。

(2) 答えは 1 つでないことを知る経験

問題解決方法は 1 つでなく、色々な考え方があつた。このことを知るにより、柔軟な発想力を育てる。

(3) 具体的に手順を書く経験

他人に処理手順を示す日本語のシナリオ作成を行わせる。シナリオの題材は、「カードを小さい順に並べるシナリオ」や「小学生に足し算を教えるシナリオ」のように、これまでの経験から考えることができる内容を選択する。さらに、コンピュータで実現できる内容とする。

(4) プログラム言語で表現・実行・確認する経験

プログラム言語の文法を教えるのではなく、サンプルプログラムを示して、真似ればプログラムが完成するようにする。プログラミングのハードルを低くして、プログラミングへの学習意欲を高める。

3. プログラミング導入教育の実践

A 高専電子情報工学科 1 年生 40 名を対象に、日本語のシナリオ作成を用いたプログラミング導入教育を実践した。授業は 1 回 90 分、5 回実施した。学習内容を表 1 に示す。以下に、5 回の学習内容の詳細を述べ

る。

(1 回目)

プログラミング経験は、論理的思考力や順序立てて問題を解決する能力を育成することができると言われていたことを紹介した。さらに、このプログラミング導入教育は、論理的思考やコンピュータで問題解決するための体験学習であると説明した。

次に、2.2 (1) のシステムを開発する立場で考えることを経験するために、自動販売機のシステムを教材として取り入れた。購入者の行為と自動販売機の処理を時系列に記述する作業を行わせた。記述するときのルールとして、「購入者は〇〇をする」というように能動態で書くことと、いくつもの処理を一緒に書くのではなく、1 行に 1 つの処理を書くように指示した。

学習者が作成した購入者の行為と自動販売機がする処理の例を図 1 に示す。作成後、処理の順序の入れ替えが可能かどうかを学習者全員で検討した。

また、2.2 (2) の答えは 1 つでないことを知る経験として、論理的思考のトレーニングによく用いられるオオカミと羊とキャベツの川渡問題を取り上げた。この問題は、1 グループ 4 名で考えさせた。このトレーニングでは、乗せて戻ることができることに気づかせ、柔軟な発想が必要であることを示した。

表 1 学習内容

No.	学習内容
1	・プログラミング教育の目的を紹介 ・自動販売機で飲み物を買うときの、購入者と自動販売機の処理の流れを書く ・論理的思考のトレーニング(1)オオカミ、羊、キャベツの川渡り問題
2	・論理的思考のトレーニング(2)8枚のコイン ・2桁の整数の足し算のやり方を教えるシナリオを作成する(グループ作業)
3	・他のグループのシナリオ通りに手作業で作業を行い、シナリオの検証をする ・他のグループのシナリオで追加・修正すべき点を赤字で添削する ・添削されたシナリオを参考に、再度シナリオを作成する
4	・コンピュータで処理を行うことを前提としたシナリオに書き換える
5	・シナリオ通りに、Cプログラム言語でプログラムを作成・実行する

購入者	自動販売機
1. 自動販売機にお金を入れる	1. お金がいくら入ったかを計算する 2. 入金された金額で買える飲み物のボタンの色を変える
2. 飲み物を選ぶ	3. 飲み物を出す
3. 飲み物を受け取る	4. お釣りを計算する
4. お釣りを受け取る	

図 2 自動販売機の処理と購入者の行為の流れの例

(2 回目)

論理的思考のトレーニングとして、見た目がそっくりな 8 枚のコインから、1 枚の偽物で重さがわずかに

軽いコインを、天秤を使って見つける問題をグループで考えさせた。

比較回数 3 回で見つけることができるという回答が多かったが、もっと比較回数を少なくできないか、再度考えさせ、2 回でできることを発見させた。

次に 2.2 (3) の具体的に手順を書く経験を行わせる教材として、「0~9 までの 1 桁の数字が書かれたカードをたくさん用意しています。このカードを使って小学生に 2 桁の足し算を筆算で行うシナリオ (手順) を簡条書きで書きなさい。ただし、小学生は 1 桁の足し算はできるものとする」⁽²⁾ という課題を与え、グループでシナリオ作成を行わせた。

説明しやすいシナリオにするために、以下のルールを決めた。

- カードを置く場所に名前を付ける。ただし、名前は英字で始める 6 文字以内の英数字の組み合わせとする。
- シナリオの書き方は、主語と述語を明確に書く
(例)・数字 3 のカードを場所〇〇に置く。
- シナリオは、1 つの処理を簡条書きにする。
- シナリオは、能動態で書く。
- シナリオは、あいまいな表現で書かない。
- シナリオは、小学生が理解できる言葉で書く。
- 条件によって手順を変えるときは、次の例のように書く。

(例) ▲もし、場所〇〇のカードが 10 以上ならば
| ・場所〇〇のカードを場所××に置く
| そうでないならば
| ・場所〇〇のカードを場所△△に置く

▼

(3 回目)

他のグループが作成したシナリオにしたがって、2 桁の足し算を行うことができるかどうかの検証を行わせた。次に、間違いや分かりにくいところを赤字で記述させた。

赤字で添削されたシナリオをもとに、再度シナリオを書き直させた。

(4 回目)

2 桁の足し算を行う学習者が作成したシナリオの中に、下記のような表現があった。

▲もし、この計算結果が 10 以上ならば

- ・場所 A1 と場所 B1 の足し算をする
- ・計算結果の 1 の位を場所 C1 に置く

下線のような表現でも、人間は理解することができるが、コンピュータでは、「この計算結果」では、通じないことや、足し算しただけでは、コンピュータは人間のように計算結果を記憶できないことを説明した。また、計算結果の 1 の位は、どのようにして求めるかを明確に示さないと、コンピュータは 1 の位を求めることができないことも説明した。

このように、人間では意識することなくできることが、コンピュータでは 1 つ 1 つ具体的に指示をしないとできないということを説明した。その後、コンピュータで処理することを意識して、再度シナリオを書き直させた。

(5 回目)

グループで、シナリオにしたがって C プログラムを作成・実行して、意図通りの結果が得られるかどうかを確認させた。

この時、2 桁の足し算のシナリオと対応させた C プログラムの main 関数のサンプルを示し、そのプログラムのなかにある同様の処理をまねながらプログラムを追加して、完成させるようにした。具体的な C プログラムの書き方は次のように指示をした。

- カードの置く場所を A1 とする
⇒ int a1;
- 数値 3 のカードを場所 A1 に置く
⇒ a1=3;
- ▲場所 A1 の値が 10 以上ならば
| ・場所 w に数値 1 のカードを置く
| そうでないならば
| ・場所 w に数値 0 のカードを置く

▼

```
⇒  
if ( a1 >= 10)  
{  
  
    w=1;  
}  
else  
{  
  
    w=0;  
}
```

シナリオのカードを置く場所の名前がCプログラム言語の変数となり、カードを置く手順は、代入になる。Cプログラム言語の変数宣言、代入、加算演算、if文については、特に説明せずにプログラムの作成を行わせた。

全グループ（4人1組）が授業時間内に、プログラムを完成させ、意図した結果を得ることができた。

4. 実践結果の分析

4.1 力と意識の変化

シナリオ作成を用いたプログラミング導入教育の事前と事後に力と意識のアンケート調査を行った結果を、表2に示す。評価は5段階（5. ある、4. ややある、3. どちらとも言えない、2. ややない、1. ない）で評価させた。回答者数は、40人である。表2のm、SD、t値は、それぞれ平均、標準偏差、検定統計量の値である。検定結果の**と*は、それぞれ有意差水準1%と5%で有意差が認められたことを示す。#は有意水準10%で有意差傾向が認められることを示す。有意水準1%で有意差が認められた項目数は28項目中13項目であり、5%で有意差が認められた項目は4項目であった。あくまでも学習者の主観によるものであるが、有意差が認められた項目に着目して、次のことがわかる。

(1) 論理的に処理手順を考える力の向上

「7.問題を順序立てて解決手順を考える力」、「9.自分の考えを順序立てて処理手順を表現する力」、「10. 処理手順の流れをチェックする力」、「11.処理手順を改善する力」に対する学習者の自己評価は有意に向上した。これにより、順序立てて、論理的に処理手順を考える力が向上していると学習者が思っていることがわかった。

(2) Cプログラム言語で表現する力の向上

「12.C言語で表現する力」に対する学習者の自己評価は有意に向上した。Cプログラミング言語の文法については、特に教えていない。しかし、今回のプログラミング導入教育のなかで、Cプログラムを作成・実行することができたことで、Cプログラム言語で表現する力が向上していると学習者が思っていることがわかった。

(3) 達成感の向上

「28.達成度」に対する学習者の自己評価は有意に向上した。5回のプログラミング導入教育の中で、カードを使って小学生に2桁の足し算を筆算で行うシナリオ書きからCプログラム作成までを完成させたことが、学習者の達成感を高めたと考えられる。

4.2 プログラミング導入教育への評価

プログラミング導入教育実践後に、今回の導入教育に対する評価アンケート調査を行った。回答は、5段階（5.思う、4.やや思う、3.どちらとも言えない、4.やや思わない、1.思わない）で評価させた。「3.どちらとも言えない」に対して、平均評定値が肯定側ないし否定側に偏っているかをt検定した結果を表3に示す。

あくまでの学習者の主観によるものであるが、表3の結果より、次のことが分かった。

(1) 「1.プログラミングの導入教育として良い経験になった」、「2.プログラミングの導入教育は楽しかった」、「21.問題解決のための処理手順を考えることが楽しい」の項目は、肯定側に有意に偏っている。これにより、学習者は、シナリオを書く作業からCプログラム完成までの導入教育を肯定的に思っている。

(2) 「3.頭の体操のような課題は、論理的思考を鍛えると思う」、「20.プログラミングの授業で論理的考え方が身につくと思う」の項目は、肯定側に有意に偏っている。これにより、プログラミング導入教育で実施した教材は、論理的思考の向上に役立つと学習者が思っている。

(3) 「7.グループで考えることができた」、「14.他のグループの手順書を確認することは良い経験になった」、「23.互いに教えあうことで論理的に考える力が向上すると思う」の項目は、肯定側に有意に偏っている。これにより、プログラミング導入教育で取り入れたグループ活動や他者評価活動を学習者は肯定的に思っている。

(4) 「3.曖昧さがないように手順を書く必要が有ることがわかった」、「13.プログラミングには論理的に手順を考えることが大切であると思った」、「19.アルゴリズム（処理手順）を考えることは重要だと思う」の項目は、肯定側に有意に偏っている。これにより、学習者がプログラミングにおけるアルゴリズムの重要性を認

識したと考えられる。

(5)「11.プログラミング言語で表現することは思ったより簡単であった」の項目だけは、肯定側に有意に偏っていない。これにより、今回のプログラミング導入教育で実施したCプログラムの作成・実行は、学習者にとってまだまだハードルが高かったと言える。

(6)「16.Cプログラム言語の文法をもっと知りたいと思う」、「17.プログラミングをもっと学びたいと思う」の項目は、肯定側に有意に偏っている。これにより、プログラミング導入教育を終えた学習者の学習意欲が向上していると言える。

4.3 プログラミング導入教育への学習者の感想

アルゴリズム導入教育に対する感想を自由記述で書かせた。その内容を以下に示す。

- プログラミングは難しく複雑なものだと思い不安だった。しかし、実際にやってみると思ったより難しくなく、アルゴリズムのおかげでスムーズにできた。
- 頭の体操があり驚いた。解いている中で楽しく解くことができ、プログラミングが難しそうという先入観がなくなった。
- 頭の体操がプログラミングに何の関係があるのだろうと思っていた。しかし、手順は1つではないことを気づかせてくれた。
- 人間にはあいまいな表現でも伝わるが、コンピュータには伝わらないことがわかった。
- 純粹に楽しかった
- プログラムに関する知識が全くなく、不安だったが、自動販売機など自分の身近な例で考えたので想像しやすかった。
- 筆算を小学生にわかるように説明するのは、想像以上に難しかった。
- 2桁の足し算のシナリオを作成するときに、計算結果をどこかの場所に置くように先に指導した方が良いと思います。

自由記述の内容には、2桁の足し算を小学生に教えるためのシナリオ作成は想像以上に難しかったという意見があった。初めての論理的で曖昧さがない表現のシナリオ作成は、2桁の足し算の仕方を理解

している学生にとっても難しい教材であることが分かった。

5. おわりに

プログラミング教育へのハードルを低くするために、初学者を対象に論理的に曖昧さのない日本語のシナリオを作成するプログラミング導入教育を試みた。実践後のアンケート調査より、曖昧さのない手順書を書く重要性(アルゴリズムの重要性)を認識させることができた。さらに、プログラミングへの学習意欲を向上させることもできた。また、論理的に処理手順を考える力が向上したと学習者が思っていることもわかった。

しかし、カードを使って2桁の足し算を筆算で行うシナリオ作成は、学習者にとって難しい課題であった。今後は、学習者にとって身近な課題であり、容易にシナリオを作成することができ、さらに最終的にコンピュータで処理できる教材を検討する必要がある。

謝辞

本研究の一部は、JSPS 科学研究費補助金基盤研究(C)(課題番号:16K01153)の助成を受けて行われた。

参考文献

- (1) 楠房子, 宮内新, 小沢慎治: “アルゴリズムスタイルを重視した情報処理教育”, 電子情報通信学会論文誌 A, Vol.J75-A, No.2, pp.441-448 (1992)
- (2) 大岩元: “日本語プログラミングによる算数教育”, FIT2016 第15回情報科学技術フォーラム講演論文集, pp.481-482 (2016)
- (3) 杉浦学, 松澤芳昭, 岡田健, 大岩元: “アルゴリズム構築能力育成の導入教育: 実作業による概念理解に基づくアルゴリズム構築体験とその効果”, 情報処理学会論文誌, Vol.49, No.10, pp.3409-3427 (2008)
- (4) 新開純子, 宮地功: “ブレンド型授業によるプログラミング教育の効果”, 教育システム情報学会誌, Vol.28, No.2, pp.151-162 (2011)
- (5) 新開純子, 宮地功: “手作業による体験的アルゴリズム学習の実践”, 日本教育工学会論文誌, Vol.35, Suppl., pp.129-132 (2011)
- (6) 山本樹, 國宗永佳: “アルゴリズム的思考における問題解決プロセスの検討”, 電子情報通信学会技術研究報告,

表2 カと意識などに関するアンケート

No.	カと意識などに関する評価項目	実施前		実施後		有意差検定	
		m	SD	m	SD	t値	結果
1	プログラミングに関する興味・関心	4.13	0.79	4.45	0.64	2.31	*
2	プログラミングに関する知識	2.05	1.08	2.45	1.11	2.34	*
3	プログラミングに関する学習意欲	4.03	0.80	4.23	0.58	1.84	#
4	試行錯誤しながら問題を解決する力	3.35	0.77	3.38	0.70	0.20	
5	筋道を立てて考える力	3.20	0.72	3.43	0.71	1.94	#
6	問題の大まかな解決方法(アイデア等)を考える力	2.98	0.89	3.38	0.81	3.12	**
7	問題を順序立てて解決手順を考える力	2.93	0.80	3.35	0.66	3.60	**
8	相手にわかりやすく順序立てて話しをする力	2.85	0.92	3.15	0.58	2.22	*
9	自分の考えを順序立てて処理手順で表現する力	2.88	0.79	3.33	0.66	3.15	**
10	処理手順の流れをチェックする力	2.83	0.87	3.40	0.71	4.16	**
11	処理手順を改善する力	2.70	0.85	3.35	0.74	4.33	**
12	Cプログラム言語で表現する力	1.75	1.01	2.60	1.03	3.82	**
13	自分の考えを他人にわかりやすく説明する力	2.75	0.95	3.13	0.82	2.15	*
14	相手の考えを理解する力	3.43	0.68	3.48	0.82	0.35	
15	相手に質問する力	3.28	0.96	3.50	0.68	1.71	#
16	他の人のプログラムを読む力	2.00	1.01	3.28	0.75	7.59	**
17	他の人の考えた処理手順を書いた報告書を読む力	2.48	1.06	3.13	0.76	3.59	**
18	問題に対して協調して取り組む力	3.60	0.93	3.80	0.82	1.35	
19	問題に対して積極的に取り組む力	3.78	0.83	3.90	0.78	1.15	
20	問題に対して最後までやり遂げる力	3.85	0.83	4.00	0.78	1.10	
21	問題に取り組む意欲	3.98	0.73	4.03	0.70	0.39	
22	プレゼンテーションする力	2.65	1.00	3.03	0.83	2.83	**
23	コミュニケーションする力	3.10	0.98	3.53	0.91	3.19	**
24	Cプログラム言語の文法知識	1.68	0.97	2.30	1.11	3.66	**
25	逐次・選択・反復の知識	2.05	0.96	2.55	1.01	2.98	**
26	コンピュータに関する興味・関心	3.93	0.62	3.93	0.80	0.00	
27	充実感, 満足度	3.78	0.97	4.13	0.72	1.90	#
28	達成度	3.63	1.00	4.33	0.73	3.82	**

** : p < .01, * : p < .05, # : p < .1

表3 プログラミング導入教育に関する評価アンケート

No.	導入教育の評価項目	m	SD	t値	結果
1	プログラミングの導入教育として良い経験になった	4.55	0.55	17.75	**
2	プログラミングの導入教育は楽しかった	4.25	0.74	10.65	**
3	曖昧さがないように手順を書く必要があることがわかった	4.55	0.64	15.35	**
4	頭の体操のような課題は論理的思考を鍛えると思う	4.45	0.64	14.36	**
5	曖昧さがないように手順を書く必要があることがわかった	4.40	0.74	11.90	**
6	プログラミングを学びたいと思うようになった	4.18	0.71	10.44	**
7	グループで考えることができた	4.18	0.71	10.44	**
8	グループ内で自分の意見を述べることができた	3.98	0.80	7.71	**
9	グループで考えることは, 楽しかった	4.03	0.83	7.79	**
10	グループで意見交換ができた	4.08	0.86	7.92	**
11	プログラミング言語で表現することは思ったより簡単であった	3.23	1.14	1.24	
12	プログラミング言語は, ルールに従えば書けると思った	3.93	1.02	5.72	**
13	プログラミングには論理的に手順を考えることが大切であると思った	4.45	0.71	12.84	**
14	他のグループの手順書を確認することは良い経験になった	4.23	0.62	12.50	**
15	他のグループの手順書を確認することは, 論理的思考を鍛えると思う	4.15	0.66	10.98	**
16	Cプログラム言語の文法をもっと知りたいと思う	4.30	0.79	10.39	**
17	プログラミングをもっと学びたいと思う	4.38	0.67	13.03	**
18	筋道をたてて考えることは大切だと思う	4.35	0.74	11.61	**
19	アルゴリズム(処理手順)を考えることは重要だと思う	4.40	0.78	11.38	**
20	プログラミングの授業で論理的考え方が身につくと思う	4.20	0.85	8.89	**
21	問題解決のための処理手順を考えることが楽しい	4.13	0.79	9.00	**
22	プログラムが思い通りに動いたときは達成感がある	4.58	0.64	15.66	**
23	互いに教えあうことで論理的に考える力が向上すると思う	4.38	0.67	13.03	**

** : p < .01