

プログラミング教育における 学習者のイメージ形成

神奈川県立産業技術短期大学校 吉田 玉緒

1. はじめに

物を作るに先立って図面を引く。図面を描いているときには、作ろうとしている物体は存在していないことが多い。われわれが存在していない物体を図面に描こうとする場合、よりどころとなるのはわれわれの頭の中にある物体のイメージである。このイメージはマイスター（熟練者）とビギナー（初心者）では異なるのだろうか。マイスターのイメージをビギナーに授受することができるのだろうか。

一般に物を作るときに出来上りをイメージするのは大切なことである。形のある物体を作る場合に比べて、ソフトウェアのような形のないものを作る場合は、この具体的なイメージを思い浮かべるのが難しい。

ソフトウェアを作るに先立って引く図面とは、フローチャートに代表されるチャート類である。われわれがチャートを描くときには、どのようなことをイメージしているだろうか。実際には、設計手法により、または個人的な設計のノウハウなどにより、プログラミングの際に持つイメージは人により異なるし、非常に複雑である。しかし、いずれにしてもより具体的なイメージを思い浮かべることができるになれば、より良いものを作ることができると考えられる。

どのようにしたら、より具体的なイメージを思い浮かべることができるようになるのだろうか。このような考え方は、スポーツにおけるイメージトレー

ニングに近いものがあると思われる。もちろん、スポーツで行われているイメージトレーニングをそのままプログラミングで行うわけではない。しかし、形のないソフトウェアをより正確に具体的にイメージできるようになれば、うまくイメージできない場合に比べて、プログラミングが容易になるのではないか。より良い「プログラムのイメージ」を学生に持たせるためには、どうしたらよいのだろうか。

このような観点から、端的な言い方をすれば「プログラミングのイメージトレーニング」が本研究の最終的な目的である。本稿では、その端緒として制御構造に関するイメージ調査について述べる。

Wirth,N. 曰く、「アルゴリズム + データ構造 = プログラム」である。プログラムをイメージするにあたって、アルゴリズムをイメージしなければならない。複雑なアルゴリズムをイメージするのは大変であるが、しかし、どのような複雑なアルゴリズムも図1に示す順次構造、分岐構造、繰り返し構造の3

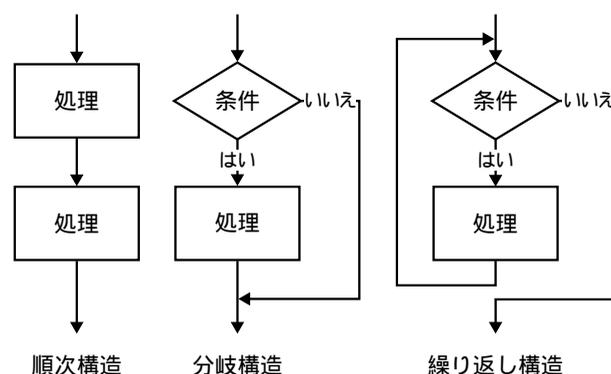


図1 プログラムの制御構造

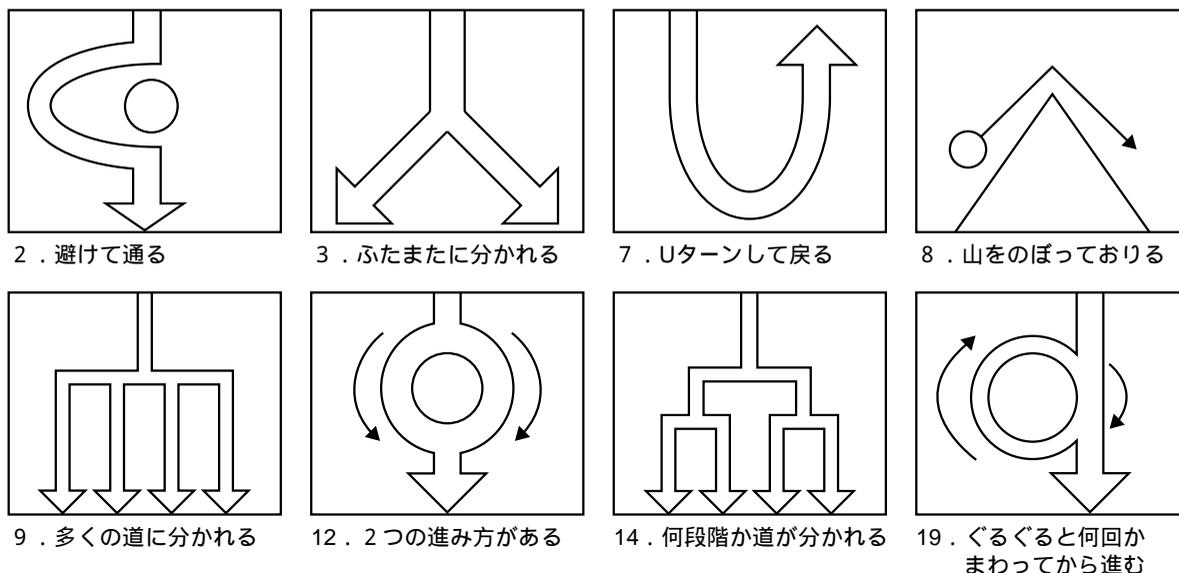


図2 イメージ図の例(一部)¹⁾

つの基本的な制御構造から構成される。

順次構造とは、いくつかの処理を順番に実行することである。分岐構造とは、何らかの条件に従って処理を分けることである。繰り返し構造とは、ある処理を複数回繰り返すことである。これは、繰り返す回数を指定することもあるが、回数は指定せずに何らかの条件を満たすまで繰り返すこともある。

プログラムをイメージするためには、これらの基本的な制御構造を正確にイメージすることが必要となる。そこで、まず、これらの制御構造を当校の学生がどのようにイメージしているか調べた。

2. 制御構造イメージ調査

学習者個人の制御構造に関するイメージは、実際には千差万別である。そこで、図2に示すような24通りのイメージ図を学生に提示し、C/C++言語で分岐構造を記述する if文、switch文、繰り返し構造を記述する while文、for文の4通りの制御文のイメージに最も近いと感じられるものを回答するアンケートを実施した。これをイメージテストと呼ぶ。なお、イメージテストおよびイメージ図は先行研究¹⁾である長谷川ら(1998)のオリジナルである。

また、個々の学生がプログラミングにどの程度習

プログラム実行後の a, b それぞれの値を答えなさい。

```
a=12;
b=0;
for ( a=0;a<10;a+=3 )
    b++;
```

a= b=

図3 理解度判定テストの例(一部)

熟しているかを測るために、図3のようにプログラムの一部を示して、そこで扱っている変数のプログラム実行後の値を問う簡単なテストを実施した。これを理解度判定テストと呼ぶ。

調査対象は情報技術科3期生34名である。2年次の9月にこれらのテストを実施した。イメージテストではイメージ図の番号を回答するだけなので、テスト実施後に数人の学生にインタビュー形式で聞き取り調査を行うことで、イメージ図を選択した理由を把握した。

また、上述の長谷川ら(1998)が先行研究を名古屋文理短期大学(現在の名古屋文理大学)で実施しているため、その結果との比較も併せて行った。

3. 結果と考察

3.1 結果概要

イメージテストの結果を表1に示す。

分岐のif文は条件を満たすものと満たさないものとの2つに分岐する命令なので、学生の回答もイメージ図番号3「ふたまたに分かれる」に集中している。同じくswitch文は多岐分岐なので、イメージ図番号9「多くの道に分かれる」と6「たくさんの中からえらぶ」(次頁図4参照)に集中している。

繰り返しのwhile文はイメージ図番号19「ぐるぐると何回かまわってから進む」に集中している。for文では、名古屋文理短期大学ではイメージ図番号19に集中したが、当校では分散する傾向を示した。

3.2 不適切なイメージ

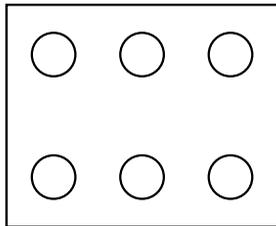
イメージテストと言った場合、テストという言葉がつくとあたかも正解が存在するような錯覚を招きがちである。現実には、制御構造に関してどのようなイメージが正しいとは言えない。しかし、どのイメージ図が不適切であるかは言えるであろう。例えば、繰り返し構造に関してイメージ図番号6「たくさんの中からえらぶ」や2「避けて通る」は不適切である。

このような不適切、あるいは適切ではないと思われるイメージ図を選んだ学生に、そのイメージ図を選んだ理由をインタビュー形式で尋ねてみた。ページの都合上、ここでは、繰り返し構造に関して述べることで、分岐構造に関しては割愛する。

表1 イメージテスト結果比較

	if 文		if 文		switch 文		switch 文		for 文		for 文		while 文		while 文	
	産業短大	名古屋文理	産業短大	名古屋文理	産業短大	名古屋文理	産業短大	名古屋文理	産業短大	名古屋文理	産業短大	名古屋文理	産業短大	名古屋文理	産業短大	名古屋文理
1	3	8.8%									1	2.4%			1	2.4%
2			1	2.4%									1	2.9%		
3	13	38.2%	18	43.9%												
4	4	11.8%	4	9.8%												
5																
6					10	29.4%	13	31.7%	1	2.9%						
7							1	2.4%					4	11.8%		
8									1	2.9%						
9	1	2.9%	4	9.8%	21	61.8%	22	53.7%	1	2.9%	1	2.4%			1	2.4%
10																
11					2	5.9%	1	2.4%								
12	9	26.5%	3	7.3%												
13							2	4.9%	6	17.6%	3	7.3%			3	7.3%
14	3	8.8%	10	24.4%							1	2.4%			1	2.4%
15	1	2.9%														
16			1	2.4%					9	26.5%	4	9.8%			4	9.8%
17																
18																
19							1	2.4%	8	23.5%	24	58.5%	25	73.5%	24	58.5%
20											3	7.3%			3	7.3%
21					1	2.9%	1	2.4%	5	14.7%	3	7.3%			3	7.3%
22									3	8.8%	1	2.4%	2	5.9%	1	2.4%
23													2	5.9%		
24																
	34		41		34		41		34		41		34		41	

(1) イメージ図番号6「たくさんの中からえらぶ」

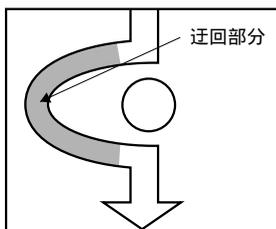


6. たくさんの中からえらぶ

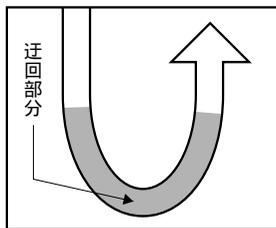
図4 イメージ図番号6

繰り返し構造のイメージとしてこの図を選んだ学生は1人で、for文のイメージとして選択している。この学生はif文のイメージとして12「2つの進み方がある」、switch文として9「多くの道に分かれる」、while文として19「ぐるぐると何回かまわってから進む」を選択しており、for文以外については不適切なイメージ図を回答したとは思われない。しかし、理解度判定テストで分岐や繰り返しがネスト(入れ子)になると正解できず、インタビューでもfor文が繰り返しであることが正しく理解できていないらしいことがわかった。

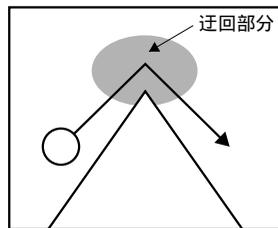
- (2) イメージ図番号2「避けて通る」
- (3) イメージ図番号7「Uターンして戻る」
- (4) イメージ図番号8「山をのぼっておりる」



2. 避けて通る



7. Uターンして戻る



8. 山をのぼっておりる

図5 イメージ図番号2, 7, 8

while文のイメージとして2「避けて通る」を選択した学生が1名、同じくwhile文のイメージとして7「Uターンして戻る」を選択した学生が4名、

for文のイメージとして8「山をのぼっておりる」を選択した学生が1名である。7「Uターンして戻る」の4名は少数ではあるが、調査対象の1割に当たり、いささか多いように思われる。また、同じ繰り返し構造でfor文とwhile文との回答の違いなど、この辺りをどう分析するかは今後の課題である。

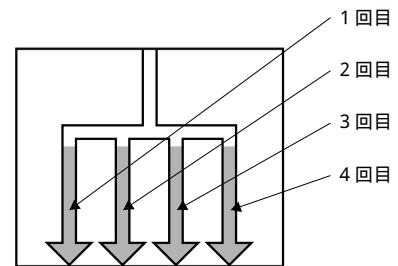
これらの学生は、それぞれ図5に網掛けで示すように、図の迂回している部分から繰り返しをイメージしていた。

7「Uターンして戻る」については、1年次にアセンブラで学習したgoto文による繰り返しのイメージが混ざってしまった学生もいたようである。C/C++の授業ではgoto文による繰り返しを教えない。イメージは、非常に個人的で感覚的なものであるらしい。

(5) イメージ図番号9「多くの道に分かれる」

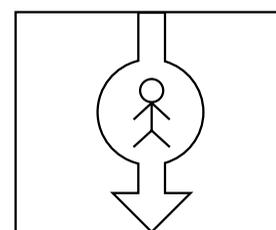
while文のイメージとしてこの図を選択した学生は1名である。このイメージ図は一般に分岐構造を表したものと思われるが、このイメージ図を回答した学生にとっては図6に示すように各々の道が繰り返しを表していると思えたようである。

- (6) イメージ図番号23「しばらく休んでから進む」
- このイメージ図を回答した学生は2名で、while



9. 多くの道に分かれる

図6 イメージ図番号9



23. しばらく休んでから進む

図7 イメージ図番号23

文のイメージとしてあげている。両名とも、for文に関しては19「ぐるぐると何回かまわってから進む」を選択している。理解度判定テストは1名が100点満点、もう1名が80点である。while文に関して、何回か処理を繰り返してから先へ進むという、処理を繰り返している部分を「休む」と連想したことからイメージしたようである。

3.3 適切なイメージ図

先行研究¹⁾では、特定のイメージ図を選択した学生の集団が適性検査および成績で優れていると報告されている。特定のイメージ図とは、if文ではイメージ図番号3および14、switch文では9、for文・while文では19である。前節で述べた不適切なイメージ図とは逆に、これらは適切なイメージ図と言える。

しかし、当校学生を対象に行った調査は必ずしもそのような結果になっていない。if文、switch文、for文、while文のイメージ回答群別に、そのグループの学生のプログラム理解度（理解度判定テスト）とプログラマ適性とを表2に示す。if文では14のイメージ図を回答した学生が理解度判定テスト・適性検査ともに点数が高いが、switch文では6のイメージ図を回答した学生の方が理解度判定テストの点数が高い。for文でも22のイメージ図を回答した学生の方が適性検査の点数が高いし、while文でも19以外のさまざまなイメージ図を回答した学生の方が理

解度判定テストの点数が高い。しかも、いずれの場合も標準偏差の値が大きい。したがって、あるイメージ図を回答した学生の集団が高得点に集中しているとは言えない。

先行研究とは異なる結果となったことから、イメージ図と適性検査およびC/C++言語プログラミング理解力との相関関係を検討するには、より多くの被験者による調査が必要と考えられる。

3.4 成績上位者のイメージ

前節で述べたように、今回の調査では特定のイメージ図を回答した学生の傾向は明らかにならなかった。そこで、逆にC/C++言語プログラミング能力が上位1割に該当する学生4名の回答を調べてみた。もちろん、4名という人数は標本としては少数である。したがって、これら4名の回答から汎用的な傾向を導き出すことはできない。しかし、今後行う調査の指針に役立つと考えている。4名の学生が回答したイメージ図番号、理解度判定テスト・適性

表3 成績上位者のイメージとプログラム理解度

学生	イメージ図番号				トレーステスト	適性検査
	if文	switch文	for文	while文		
A	14	9	19	7	100	46
B	3	9	16	19	80	65
C	12	6	22	19	100	86
D	14	11	16	19	51	77

表2 産業技術短大生の制御構造のイメージとプログラム理解度

if文					
イメージ図番号	人数	トレーステスト		適性検査	
		平均	標準偏差	平均	標準偏差
3	13	66.2	9.9	59.0	8.8
12	9	60.0	13.7	61.3	12.4
14	3	73.3	12.5	64.7	13.4
上記以外	9	51.1	10.7	55.1	10.7

switch文					
イメージ図番号	人数	トレーステスト		適性検査	
		平均	標準偏差	平均	標準偏差
9	21	61.9	11.1	62.9	9.8
6	10	64.0	13.0	56.9	12.6
上記以外	3	46.7	12.5	61.7	10.9

for文					
イメージ図番号	人数	トレーステスト		適性検査	
		平均	標準偏差	平均	標準偏差
16	9	62.2	13.7	59.6	13.6
19	8	67.5	11.1	56.3	7.2
13	6	60.0	10.0	56.3	11.1
21	5	56.0	8.0	61.0	7.9
22	3	66.7	12.5	73.7	8.7
上記以外	3	46.7	12.5	53.0	1.6

while文					
イメージ図番号	人数	トレーステスト		適性検査	
		平均	標準偏差	平均	標準偏差
19	25	60.0	12.2	60.6	11.8
上記以外	9	64.4	11.3	54.9	7.8

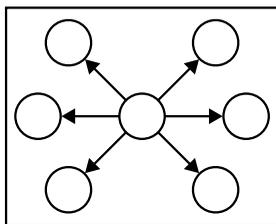
検査の得点を前頁表3に示す。

学生Aは適性検査の得点が低く、while文のイメージ図として7「Uターンして戻る」を回答している。理解度判定テストの得点は満点で、プログラム理解度は高い。プログラミングに関して適性というものは確かに存在する。しかし、プログラミングは芸術・文学などと違って、センスがなくても努力することによってこれをある程度は補うことができる。Aはそのような努力型の学生であると考えられる。

学生B、Cは適性検査の得点も高く、理解度判定テストの結果も良好である。学生Dは適性検査の得点は高いが、理解度判定テストの結果は芳しくない。この学生は実際の作業では理解力・応用力の高さを示すが、ムラが多く、理解度判定テストのようなペーパーテストではなかなか高得点を得られない。

これらの学生が回答したイメージ図のうち、表1で少数意見となっているものについて検討した。

分岐に関してはDがswitch文のイメージ図として11「ちらばっていく」を回答した。処理すべきデータが図の中央の丸で、この丸が条件に従って左右6つの丸のどれかに移り変わっていく、状態遷移図のようなイメージでDはこの図を選んだようである。

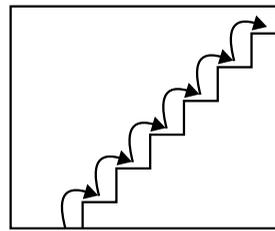


11. ちらばっていく

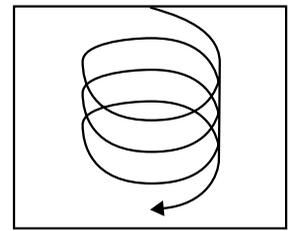
図8 イメージ図番号11

繰り返しに関してはBとDがfor文のイメージ図として16「階段をのぼる」、Cが同じくfor文のイメージ図として22「らせん状に進む」を回答している。いずれも、繰り返すことによって処理が進行している様子をイメージしたようである。

そのように考えると、分岐のイメージ図として3、9、12、14や繰り返しのイメージ図として19を選ぶのは、フローチャート記述法の影響を受けているように思われる。もちろん、フローチャート記述法に



16. 階段をのぼる



22. らせん状に進む

図9 イメージ図番号16, 22

基づいて制御構造のイメージを形成しても構わない。しかし、フローチャート以外の記述法の影響を受けたイメージや、その学生が本来持っているイメージにより近いものがこの場合のイメージ図番号11、16、22であるならば、これらは「不適切なイメージ図」とは言えない。同様に、「適切なイメージ図」も前節で述べたもの以外に複数あり得ると思われる。これらについては、現在検討中であり、今後の課題である。

4. おわりに

プログラミングの学習は、文法の理解と課題演習をもって行うのが一般的である。制御構造の学習に際しても、理解を助けるためにフローチャートなどの図式表現を用いることはあるが、学習者が持っているイメージに注目した学習法は知られていない。

今回、学習者の持つ制御構造のイメージを調べてみて、その多様性に驚かされた。本研究は端緒に終わったばかりである。本稿においても、for文とwhile文との回答の違いや適切なイメージ図・不適切なイメージ図の扱いなど今後の課題が山積している。

当然のことながら、教育手法のための実験を行うことによって授業に支障が出ては本末転倒であるため、正確な比較実験を行うのが難しいことが多い。条件が許せば、学生を複数のグループに分け、動画などマルチメディア教材²⁾を用いて適切と思われるイメージを持つように指導した場合にどのような相違が見られるかを比較してみたい。

参考文献

- 1) 長谷川聡・山住富也：「プログラミング教育と学習者のイメージ形成(その2)」,名古屋文理短期大学紀要,1998.
- 2) <http://www.asahi-net.or.jp/JF1T-YSD/report/2000-01/2000-01.html/>