

# プログラミング導入学習の改善

大木 真\* 永田 和生\*

## The improvement of the introductory learning of the Programming

Makoto Ohki\*, Kazuo Nagata\*

**Abstract:** This paper tries to the improvement of introductory learning of the Programming which is indispensable capability of engineers. We divide Programming into "grammar" and "logical thinking", and aim at the lecture which can be learning while enjoying "logical thinking". As a result, students learned "logical thinking" steadily, with enjoying.

**キーワード:** 教育, 授業改善, プログラミング, C, Scratch

**Keywords:** Education, Improvement of the class, Programming, C, Scratch

### 1. はじめに

#### 1.1 言語とスポーツは同じ「練習」するもの

現状の本学の学生を見ると、プログラミングへの苦手意識の強い学生が多くみられる。そして、他の専門科目と違い、そういった苦手意識を持った学生はプログラミング能力が非常に低い。他の専門科目では、たとえ苦手だとしても基本的な内容は理解してきていると感じる。なぜ、プログラミングだけ特に格差がついてしまうのか？実はこれと同じ現象が一般科目の英語にもみられる。英語とプログラミングの共通点は「言語」であるということである。

言語は、他の科目とは違った特性を持っている。スポーツや芸術などと同じで「練習」をしないと上達できないという点である。野球のコーチング本を読んだだけではホームランは打てない。それと同じで英語の教科書を熟読しても、いきなり英語は話せない。英語やプログラミングが得意な人というのは、知らず知らずのうちに「練習」を繰り返している。英語なら「聞く」「読む」「話す」「書く」の反復練習を、プログラミングなら色々な目的のプログラムを沢山「書いて」いる。その過程で、間違いながら繰り返し練習することで自然に身に付けていくのが「言語」である。

英語やプログラミングの学習初期につまずいて苦手意識を持ってしまい、それを楽しくないと感じてしまうと、このような「練習」からなるべく避けるようになってしまう。そのため、他の科目に比べ、出来不出来に大きな差が表れてしまう。そこで、本論文での試みでは、まず「プログラミングが楽しい」と感じてもらうことを第一の目的とする。

#### 1.2 プログラミングの重要性

プログラミングは、もはや数学や英語と同じくらい、現代の技術者には必要とされる能力である。情報系の学科だけでなく、機械系、電気系、材料系、生物系、化学系、建築系など、どの分野の職業に就いたとしても、必ずかかわることになる。機械設計や回路設計、金属加工、素材の組合せ、薬剤の組合せ、構造物強度の計算、などなど、様々な計算やシミュレーション、機械制御などがコンピュータによって行われる。つまり、プログラミングの基本を理解していないと、仕事がスムーズに行えなくなる。

#### 1.3 プログラミング習得のための2つの学び

プログラミングを習得するためには、2つの学びを得る必要がある。それは「文法」と「論理的思考力」である。しかし一般的には、これら2つを分類して学習するということはない。

高専や大学において、プログラミングの入門科目としてC言語による講義が行われる。プログラミング言語の中でも有名なC言語は、世界中で広く使われており、上位言語習得への大きな足掛かりにもなるため、プログラミング入門科目として必ず習得すべき言語だからである。高専や大学での講義やC言語の入門書では、C言語の「文法」とプログラムに必要な「論理的思考力」を同時に習得させようとする。ここで、学習者がどちらかの習得につまずいてしまうと、それが苦手意識となりプログラミング習得の足かせとなってしまう。

本論文での試みは「文法」と「論理的思考力」を分け、それぞれを順番に学ぶことによって、より多くの学習者に、よりスムーズにプログラミングの基礎を身に付けてもらうことを第二の目的とする。そこで1年生の基礎科目である「創造基礎工学」の4月から前期中間試験までの期間(7週間)を対象として、まずは「論理的思考力」のみを誰でも

\* 熊本高等専門学校 情報通信エレクトロニクス工学科  
〒861-1102 熊本県合志市須屋2659-2  
Dept. of Information, Communication & Electronic Engineering,  
Kumamoto National College of Technology  
2659-2, Suya, Koshi, Kumamoto 861-1102

身に付けられるような講義を実現する。

実現手段として、日本語によってプログラムを記述することができる Scratch<sup>(1)(2)</sup> を用いて、「論理的思考力」の習得を支援する。Scratch は、例えば C 言語での for 文を「～回、繰り返す」のように日本語で記述することができるため、文法を学ぶ必要がない。この Scratch を用いてプログラミングに必要な「繰り返し」「条件分岐」「変数」「配列」の概念を学ぶ。

また Scratch は、Internet Explorer や Google Chrome 等のウェブブラウザ上で起動することが可能である。インストールの必要もないため、自宅等の PC でも今すぐに始めることができ、プログラミングの「練習」をする環境の準備を必要としない。講義で使用する際も、何十台もあるパソコンにインストールする手間を取らず、すぐに実施することが可能である。

## 2. Scratch とは



図1 Scratch によるプログラミング記述画面

Scratch とは、最初に文法を学ばなくとも記述した結果（動作）を得られるプログラミング言語である。触覚や視覚的 GUI によって、感覚的にプログラムを記述することが出来る。マサチューセッツ工科大学の MIT Media Lab によって子供向けに開発された。

図1に示す中央枠内のブロックを組み合わせて右枠内に置く（記述する）ことでプログラムが完成され、左枠内のキャラクターが動作する。以下のような特徴を有す。

- (1) 文章を記述する必要がない  
「もし○なら」や「○歩動かす」などの多数の用意されたブロックを組み合わせることでプログラミングできる。
- (2) 文法の学習が必要ない  
例えば、繰り返し処理を「○回繰り返す」というブロックで表現できる。
- (3) 自分の描いた絵が、すぐ目の前で動く  
Scratch 内で描くことも、ペイント等で書いた画像の読み込みも可能。

つまり、言語を気にせずプログラミングが可能であるため、文法を学ばせる必要なく、論理的思考力のための練習が可能である。また、すぐ目の前で動作してくれることから、

難しさを感じさせずに導入学習を実施できる。

## 3. 実施内容

本章では、平成24年度と25年度の情報通信エレクトロニクス工学科1年生を対象に実施した講義の概要を記述する。実際に使用したテキストは、本校の WebClass 「創造基礎工学 (TE1)」コースに全文が公開されているため、詳細はそちらを参考されたい。

### 3.1 「繰り返し」の基礎

「繰り返し」の概念の学習として、図形描画を実施した。図2の左側の様な四角形を描画するには、「直線を描く」「90度方向を変える」という処理を4回実施すれば良い。これをそのままプログラミングすると図2中央の記述になるが、「繰り返し」を用いれば図2右のように簡素になる。この説明を受けた後、図3のような複数の図形をなるべく少ないブロック数で描画するプログラムを考える。



図2 「繰り返し」の基礎

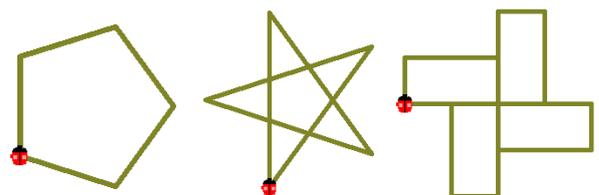


図3 課題の図形 一例

### 3.2 「条件分岐」の基礎

条件分岐ではライントレイサを作る。車の図形と、コースの背景を描き、「もしコースアウトしたら戻る」というプログラムの作成を行う。右に出たら左に、左に出たら右にハンドルを切ればいだけだが、それを各自で考える。



図4 「条件分岐」でライントレイサ

### 3.3 「変数」の基礎

乱数を用いたおみくじと、変数を用いた自動計算プログラムを作成する。「世界のなべあつ」のような動作（3の倍数と3の付く数字に反応する）をさせたり、基数や偶数や素数の数を数えるプログラムを作成することで、プログラミングの数学的な考え方と、変数の概念を身に付ける。

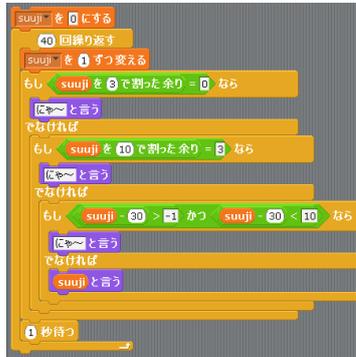


図5 3の倍数と3の付く数字に反応

### 3.4 「配列」の基礎

前節の「変数」と違い、沢山の値に順番を付けて格納できる「配列」の概念を学ぶために、画面に散らばせた色違いの玉を探索するプログラムを作成する。キャラクターをランダムに動かし、見つけた玉の色を配列に格納していく。見つけた順番によって、格納される順も変わってくる。

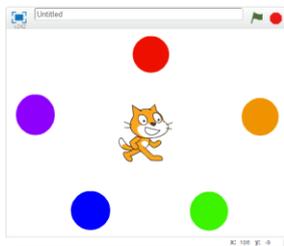


図6 ランダム探索プログラム

### 3.5 総合課題

最後の2週間（4コマ）は、2つの総合課題に自力で挑む。原文をそのまま記載する。

#### (1) 迷路探索プログラム

##### 【制約】

- 壁に触れるのはOKだが、すり抜けた場合は失格。
- キャラクターは70ピクセル×70ピクセル（スタート地点の水色マス）に収まるサイズとする。
- スタート地点は水色。赤丸に触れたらゴールとする。
- プログラム作成時のためのテスト迷路は公開するが、本番用迷路は非公開とする。
- 1回のループ内で進める歩数は3歩までとする。
- キャラクターは1つのみ。2つ以上作ってはいけない。

##### 【評価】

- ・2度チャレンジして、その平均タイムを評価する。
- ・60秒以内にゴールできたら100点。
- ・60秒を超えた場合、1秒につき-1点。

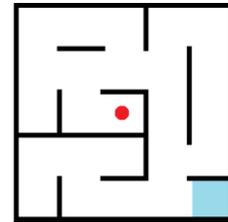


図7 迷路探索プログラム

#### (2) 「数値計算プログラム」

「1からnまでの自然数に素数はいくつ存在するか？」を数えるプログラムを作成する。

## 4. 演習状況と学生アンケート結果

### 4.1 演習状況と所感

平成24年度と25年度の情報通信エレクトロニクス工学科1年生を対象に、4月の一番最初の講義から6月の中間試験までの約7週×2コマの講義を実施した。

#### (1) 第1週目

一番最初の講義ではなるべく平易な説明を心がけ、PC操作とScratchに完全になれさせる。2週目からは基本的にはテキストを各自で読み、演習形式で例題を順次解いていくことで、その日の提出課題まで行き着かせる。

#### (2) 第2週と第3週

2週目から「繰り返し」の基礎に入る。中学から進学してきたばかりで、PC操作そのものに慣れていない学生とPCに詳しい学生とで差が激しい。そのため、各自の進みに大きな差が出る。しかし、もともとPC操作をしたことのない学生を想定しているため、不慣れな学生でもギリギリ課題達成できる。進みの早い学生は、完成後に独自の工夫をいれたり、他者に教えることが出来る。

#### (3) 第4週と第5週

ほぼ全員がPC操作、Scratch操作ともに慣れ、トラブルや質問が出てくるのが減る。課題も難しくなってくるのでちょうど良い。このあたりからは学生の工夫が多くなり、教員は見ていると楽しめる。

#### (4) 第6週と第7週

今までに習得した技術を総動員して総合課題に挑んでもらう。平成24年度、25年度ともに、迷路探索課題で最初はほとんどの学生が「絶対に来ない」と諦めがちである。しかし最終的には全学生が100点（60秒以内に迷路を攻略）のプログラムを完成させる。一方で数値計算プログラムは半数程の学生が完成に至らない。

### 4.2 学生アンケート結果

表1、表2に実施したアンケートの結果を記載する。全体

表1 学生アンケート結果

	プログラミング経験の有無	問い	「繰り返し」「条件分岐」「変数」「配列」について何項目理解できたか	プログラミングを得意と感じるか苦手と感じるか	Scratchによるプログラミングの授業は楽しいか	後期に行われるC言語の授業は楽しみか
	言語	回答	1: 全て 2: 3項目 3: 2項目 4: 1項目 5: 0	1: 得意 2: まあ得意 3: すこし苦手 4: 苦手	1: とても楽しい 2: まあ楽しい 3: あまり楽しくない 4: 楽しくない	1: とても楽しみ 2: まあ楽しみ 3: あまり楽しみではない 4: やりたくない
H24	経験者なし	1	13人	2人	29人	21人
		2	14人	16人	15人	20人
		3	17人	19人	1人	4人
		4	1人	8人	0人	0人
		5	0人			
H25	スクイーク	1	19人	0人	18人	13人
	KIROBO	2	8人	14人	22人	25人
	覚えていない×3	3	15人	21人	2人	1人
	計5名	4	0人	5人	0人	1人
		5	0人			

表2 学生アンケート コメント抜粋

	特に楽しかったこと自由記述 (多かったコメント抜粋)	分かりにくかったこと自由記述 (多かったコメント抜粋)	感想、意見、要望、なんでも自由記述 (多かったコメント抜粋)
H24	自分で考えながらできた	変数、配列の使い方	(C言語などの)本格的なプログラミングをしなくなった
	最初がすごく簡単で入りやすい	条件文が複雑になると、どう動くのか	皆で学び合うことができた
	後半は結構難しいが出来た時の達成感。特に最後の迷路探索	乱数の動作	ヒントがあるのはいいが、解答例まであるのは良くない。解答例をすぐ見てしまった。
	思った通りに動かせたとき	for, ifなどの用語	
	答えが一通りでないのに、友人と比較できる 友人や先生と考えながら出来た		
H25	自分が、そうなって欲しいと思っていた通りにスプライトが動いてくれた時	難しい課題にヒントがないものもあり、ちょっときつかったものもあった	難しいけど、楽しい授業だと思います。友人と話し合いながら進めていくことができていると思いました。
	各種課題が面白かった	変数が、よく分からなかった。	本格的なプログラムも楽しみ
	周りの人と相談しながらすることも楽しかったです。	特になし	テキストがとても分かりやすかった
	課題を順に沿ってしているうちにだんだんプログラミングに上手になっていくのが分かったこと		もう少しゆっくりなペースの授業だと嬉しい
			もう少し変数についてのヒントがほしかった

的に「それなりに理解度が高い」「半数は苦手意識が残る」「プログラミングを楽しいと感じる」「C言語の講義が楽しみ」という結果となった。自由記述のコメントを見ても楽しんで身についたと感じてくれている学生が多い、一方で「変数」の概念理解に苦戦していることが分かる。

最大の目的である「論理的思考力の習得」と「プログラミングを楽しいと感じる」ことは大まかには達成できたと言えるが、まだまだ改善の余地がある。

## 5. まとめと今後の展望

### 5.1 まとめ

本論文では、数学や英語とほぼ同列に技術者の必須能力となっているプログラミングの導入学習の改善を試みた。プログラミングを習得するために必要な「文法」と「論理的思考力」の2つの学びのうち、「論理的思考力」のみを楽しみながら学べるような講義を目指した。結果として9割以上の学生が楽しみながらも、着実にプログラミングに必要な「論理的思考力」を身に付けられた。しかしながら、「変数」の概念など、まだまだ改良をすべき余地が残るため、来年度以降の改善が必要である。

### 5.2 今後の展望

#### (1) 演習資料改善

平成24年度から25年度の間でも改善を行っているが、いくつかの言葉や表現の改善が必要である。また「変数」と「配列」については、その概念がより伝わりやすいように

改善が必要であると考えられる。

#### (2) C言語へのつながり

本論文での試みはScratchを身に付けさせることが目的ではない。これを足掛かりとして、C言語の習得をスムーズにすることである。そのため、本講義からのつながりとしてC言語の講義を後期に実施予定である。平成25年度ではScratchの「繰り返し」がforになったり、「条件分岐」がifになる、というようにScratchで得た考え方を基に、C言語文法で学ぶようなやりかたを考えている。そして、少しずつScratchの話は減らしていき、1年生の終わりには完全にC言語の講義になるようにしたい。あくまでScratchは足掛かりであるので、最終的には完全に離れさせる。

## 6. 謝辞

本論文での試みは平成24年度および25年度の本学「校長裁量経費(B)教育改善充実経費」の助成による。

(平成25年9月25日受付)

(平成25年12月3日受理)

## 参考文献

- (1) Scratch ウェブサイト: <http://scratch.mit.edu/>
- (2) 石原正雄:「スクラッチアイデアブック」, 株式会社カットシステム, (2009).