

情報処理入門教育のためのプログラミング言語の選定*

堀内泰輔*¹・横山靖樹*²・佐藤優介*³

Selection of the Programming Language for Teaching Introductory Information Processing

HORIUCHI Taisuke, YOKOYAMA Yasuki and SATO Yusuke

キーワード：情報処理教育，プログラミング教育，プログラミング言語，Processing 言語

1. はじめに

本校では、全学科 1 年生に共通の情報処理入門教育を行っており、筆者*¹は「情報処理基礎」(以下、本科目と略す)を通年で 2 単位開講している。

本科目においては、PC やインターネットのリテラシー教育のほかに、プログラミングの基礎教育も行っており、さまざまなプログラミング言語を利用してきた。

古くは BASIC や LOGO などの伝統的な教育用言語を、その後、Pascal や C などの実用言語を用いてきた。その後で、Basic を復活させたこともあったが、最近では Web の発達に同期して JavaScript や Java, 初学者に易しい Squeak, さらに平成 22 年度までは日本語で記述できる「ドリトル」と、半ば試行錯誤的に利用してきている。

本稿では、上級学年での言語教育に連携でき、現在の IT に関する社会情勢と学生気質にマッチした、高専入学生にふさわしい、理想的な入門用プログラミング言語について検討する。

2. これまでの言語の使用実績

本科目でのプログラミングはあくまで入門としてのものであり、2 学年以降(学科によっては 1 学年後期以降)の本格的なプログラミングに備えるためのもの

* 2011 年 8 月 25 日 第 31 回高等専門学校情報処理教育研究発表会にて発表

*1 一般科教授

*2 技術支援部技術専門職員

*3 技術支援部技術職員

原稿受付 2012 年 5 月 18 日

のである。もちろん、2 学年以降は各学科がそれぞれの方針でプログラミング教育を行うため、使われている言語も様々である。本科目に各学科のニーズをすべて取り入れるのは物理的に不可能であるため、プログラミング言語の選定とその教育内容は、担当教員に任されている。

表 1 には、筆者がこれまでに本科目で用いてきたプログラミング言語について、過去 15 年分の実績を示す。平成 18 年度までは、HTML を除いて、後期のみプログラミングを行うこととしていたが、19 年度より、前期の後半まで前倒しにして、プログラミングの比重を増してきている。

3. 高専入学生のための言語の条件

以上のような背景において、本科目にふさわしいプログラミング言語の条件、いわば高専入学生のたの条件を考察してみたい。

表 1 これまでに利用した言語一覧

年度	前 期	後 期
H9	(なし)	JavaScript
10		C
11		
12		
13		
14		
15		
16		
17		
18		Squeak
19	ドリトル	Java
20		JavaScript
21		Processing
22		
23	Processing	

最も重要と思われるのは、学生にプログラミングの魅力を知らしめることができる言語である。初期の言語では計算しかできなかったため、目標がソーティングなどの数値処理や数値計算に限定された。これでは、学生に興味を惹かせることは不可能であった。故に、グラフィックスやアニメーションに代表される図形処理ができる言語が必要となる。さらに、PC に接続されたもの（センサーやロボットなど）を制御できることが望ましい。これは、高専という学校特有の条件でもある。

次に、プログラミングが行いやすい環境を持っていることである。単なるエディタではなく、その言語の文法に対応して、構文を色分けしてくれたり適切な選択肢を自動表示するものが望まれる。また、実行環境においても、エラーを適切にわかりやすく指摘してくれるものが望ましい。

また、ある動作を実行させるための設定や準備ができるだけ少ないことも大切な条件の一つであろう。このため、多くの宣言を要する言語はふさわしいものではない。機能が多用実用言語というよりは、機能が少なくても、また実行速度が遅くとも、初学者に十分なプログラミングが可能であれば、本科目のための言語としては十分である。

加えて、2 学年以降の本格的なプログラミングとの連携が効率的に取れる言語であることが望まれる。表 2 は、本校で現在実施されている授業の中で、プログラミング教育に用いられる言語を学科別にまとめたものである（シラバス上で、言語名が明言されているもののみ）。これによると、C や Java など、文法が C 言語に似た、いわゆる C 系の言語が多いことがわかる。

平成 22 年度までは、日本人初学者がプログラミングに苦手意識を持たないように、日本語で記述できる言語「ドリトル」を採用してきたが、我々の意に反して学生には不評であった。この理由は、この言語が小学生にもプログラミング可能なものである反面、日本語としては不自然な文法を強いられる点と、英語で書かれていても上級学年や世の中で通用するポピュラーな言語を勉強したい、という高専生としての本音によるものであった。現在広く社会で用いられている言語は、C 言語の文法の影響を受けているものが多い。このため、C 系の言語が望ましいと考えられる。ただ、その一方で、Basic 系の言語もよく用いられている現状もある。

一方で、最近では中学校でプログラミングの初歩を教えることも多くなり、基礎的なプログラミング能力を持つ入学生も増えてきた。表 3 は、本科目で毎

年実施している入学直後のアンケートの中で、これまでに勉強した経験のある言語（独学・学校を問わない）とその人数（延べ人数）を、この3年間に付いてまとめたものである。

これによると、経験者人数は次第に増加しており、平成 23 年度では4人にひとり経験があると答えている。また言語別では、HTML を除くと C 言語の人数が年々増えている。その反面 Java は減少、Basic は増加傾向にある。前述のドリトルが不人気な原因が、このような経験者の増加傾向にもあると

表 2 本校の授業で使われている言語一覧

学科名	学年	科目名	言語名
機械	3	プログラミング演習	C
	5	数値計算法	C
電気電子	3	プログラミング言語 I	C
	4	プログラミング言語 II	C
電子制御	3	情報処理	C
電子情報	1	情報工学入門	C
	2	情報処理	C
	2	工学実験実習 II	C
	3	アルゴリズム論	C
	3	オブジェクト指向	C++, Java
	4	マイコンシステム	C
	4	プログラミング演習	Java
	4	工学実験実習 IV	C
	5	コンピュータグラフィックス	Java
5	工学実験実習 V	Lisp	
環境都市	4	情報処理	FORTRAN
全学科 (選択)	4	情報処理応用 A	Python
	4	情報処理応用 B	PureData

表 3 中学時代までに経験した言語 (単位:人)

言語名	H21年度	H22年度	H23年度
HTML	18	15	21
C	2	12	13
Java	12	7	5
Basic	4	9	8
C++	2	5	1
ドリトル	1	4	2
HSP	1	1	1
Perl,JavaScript,Delphi, なでしこ,XCode	各 1		/
C#,Tonyu,Processing,ActionScript	各 1		/
「わからない」	1	3	5
「忘れた」	1	/	1
回答人数 (%)	33 (16.7%)	43 (22.0%)	48 (26.1%)
非回答人数	165	153	136

見られる。

その他の条件としては、オブジェクト指向プログラミングが実現できることが望ましい。入門教育の段階ではオブジェクト指向は必須とは限らないが、オブジェクト指向への移行がしやすい言語であれば、高学年での応用プログラミングで役立てることができよう。

4. 代表的な言語とその比較

以上までの考察により、本科目にふさわしいプログラミング言語の候補をいくつか挙げて上述の条件にどの程度あてはまるかを検討する。

まず、C 系言語の元祖にあたる C 言語であるが、オブジェクト指向ではない欠点がある。他の条件については一応満足はするものの、初心者用に工夫された点は、処理系にもよるが、ほとんどない。

次に、C++は C 言語にオブジェクト指向を加えたものだが、初心者の習得は非常に困難である。

C#は Microsoft 社が C や C++を改良したもので、開発環境が整っている。しかし、簡単なプログラムを作成するのにさまざまな宣言などの準備が必要なのは上述の C 系言語と同様である。

Java は C#と同様に開発環境が整っているが、初学者にはハードルがかなり高い。

JavaScript は HTML と組み合わせてプログラミングを行う特徴があるが、本来のプログラミングの環境を学ぶには違和感がある。エラーのデバッグもやりにくい。

Processing は、C 系言語の文法を踏襲しているが、余計な準備や設定をせずに、直接プログラミングの核部分を記述できることは大いに評価ができる。グラフィックスは 3D も備え、アニメーションも比較的簡単にプログラムできる。構造エディタを内蔵し使い勝手も良い。作成したプログラムは、Web 上で簡単に公開できるようになっている点は斬新である。ただし、日本語がコメント以外のところでは使用できない点が難点である。さらに、エラーメッセージが小さな英字で表示されるため、極めてわかりにくい。

次に Basic 系言語に言及する。Basic はグラフィックスを簡単に扱うには最高の言語であり、従来、

教育用に用いられてきた。現在では本来の Basic を現在の PC のために拡張した言語のほか、Microsoft 社からは VisualBasic や最近では SmallBasic が供給されている。前者はオブジェクト指向の Basic で開発環境も整っているが、Java などと同様に大規模な言語であるため習得は容易ではない。

これに対して後者は、VisualBasic や Java を学ぶ前の、教育用言語として開発された Basic 系言語であり、最適化されたエディタを内蔵しているため、初学者でも容易にプログラムの入力が可能である。また、Basic であるから、グラフィックスの描画も簡単である。ただ、C 系言語ではないため、特に制御構文の互換性が全くないことから、C 系言語へのステップにはなりにくいことが唯一の欠点である。

5. Processing を用いたシラバス作成

以上の議論により、平成 23 年度の本科目では Processing を通年で採用することとした。表 4 には、シラバスの中で、Processing に関する内容を示す²⁾。また、プログラミングの過程の画面例（内蔵エディタおよび、実行画面）を図 1～2 に示す。

本科目は 2 単位しかないので、全時間の 3 分の 1 程度しか Processing に割くことができない。そのた

表 4 Processing 教育に関するシラバス

授業項目	時間	内容
Processing 言語入門	2	Processing 言語の歴史や特徴が理解できる
プログラミングの意味とリテラシー	2	プログラミングの意味とこれに関する基本的な事項が理解できる
かたちと色のプログラミング	4	かたちと色のプログラミングに関するプログラミングができる
計算と変数	2	計算に関するプログラミングと変数が理解できる
繰り返しとランダム	3	繰り返し構造のプログラミングが理解できる
条件と分岐	3	条件と分岐のプログラミングが理解できる
画像や文字を使う	2	画像と文字をプログラミングに適用できる
作品を作る	2	オリジナルな作品を作成できる
Processing 総合演習	2	以上のプログラミング技術を用いて、基本的な作品を制作し公開できる

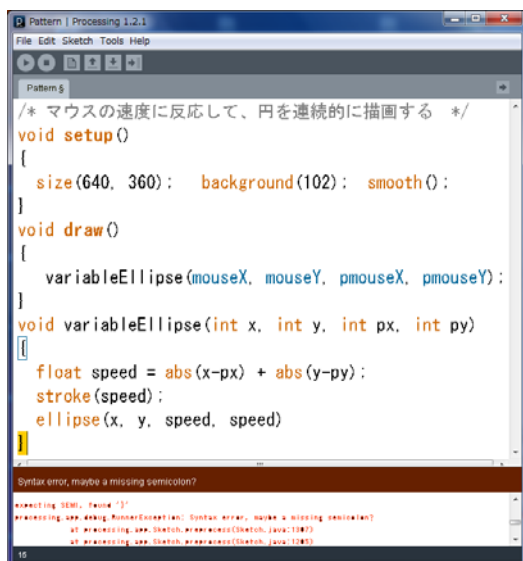


図1 Processingのエディタ画



図2 図1のプログラムの実行画面

めグラフィックスから入門させ、変数の利用、制御構造、画像と文字の処理程度までの内容になっている。次の段階である配列を犠牲にして、Processing

の特徴の一つである、簡単にできる Web 公開を目標とする、総合演習を最後に設けた。

教材は自作することも検討したが、準備期間の不足から、市販のもの³⁾を用いた。

6. おわりに

Processing という言語が、高専入学生にとってベストな選択であることを述べた。

前述のシラバスに沿って、平成 23 年度の前期授業を終える段階であるが、学生の受けは上々である。これは、命令語を入力時に色分けしてくれる構造エディタの存在、準備がほとんど不要なグラフィクスやアニメーション、などが学生のプログラミングへの意欲を掻き立てるためと思われる。今後の動向については次回に発表することとしたい。

参考文献

- 1) 堀内泰輔：「日本語プログラミング言語を用いた情報処理入門教育の実践」, 高等専門学校情報処理教育研究発表会論文集, Vol.30, pp.89-92(2010.8)
- 2) 国立長野高専シラバス「情報処理基礎」：
<http://www.nagano-nct.ac.jp/course/syllabus/H23/18000001.pdf>(2011.4)
- 3) 田中孝太郎・前川峻志：「Build with Processing デザイン／アートのためのプログラミング入門 (改訂版)」, BNN 新社(2008.3)