

実態調査に基づく一般情報教育としての プログラミング教育の検討

堀越真理子*

A Study of Programming Education as General Education Based on Survey Data

Mariko HORIKOSHI*

抄 録

将来的に深刻な IT 人材不足を背景に、近年、初等中等教育においてプログラミング教育が重視され必修化が進んでいる。こうした中、高等教育におけるプログラミング教育の重要性も増しており、果たす役割は大きいといえる。筑波学院大学では、2010年度から1年次全学生を対象にプログラミング科目を必修として開講している。また、筆者は、新入生のプログラミング経験や意識について実態を把握するため、2015年度から継続的にアンケート調査を実施してきた。その結果、プログラミングに高い興味・関心を持っているものの、大学入学時点でのプログラミング経験者は毎年僅か1割強にとどまり、さらに、経験のある学生でも一部の学生を除いては「体験」した程度であることが明らかになった。本稿では、これらの調査結果を報告するとともに、授業の内容や取り組みについての有効性を検証し、高等教育における一般情報教育としてのプログラミング教育のあり方について考察した。

キーワード：プログラミング教育、一般情報教育、高等教育、実態調査、マクロ (VBA)

1. はじめに

近年、プログラミング教育のニーズが高まっている。とりわけ、初等中等教育においてプログラミング教育が重視され必修化が進んでいる。2017年の学習指導要領の改訂¹⁾で、2020年度から小学校でプログラミング教育が必修化されることになり注目を集めている。中学校では既に2012年度から技術・家庭科の授業で「プログラムによる計測・制御」

という内容が必修になり、生徒全員がプログラムを学ぶようになってきているが、2021年度からは「小学校におけるプログラミング教育の成果を生かし、発展させるという視点から、従前からの計測・制御に加えて、双方向性のあるコンテンツに関するプログラミングや、ネットワークやデータを活用して処理するプログラミングも題材として扱うこと」とし、拡充される²⁾。また、高等学校では科目の再編を行い、2022年度からは全ての生徒が履修

* 筑波学院大学経営情報学部、Tsukuba Gakuin University

する「情報Ⅰ」が新設され、「プログラミング、ネットワークやデータベースの基礎等の内容を必修化する」としている³⁾。さらに、政府は第16回未来投資会議⁴⁾において、大学入試でプログラミングが含まれる情報の試験を必須とするなどの入試改革を提言し、「情報Ⅰ」を大学入学共通テストへ導入する方針を示した。早ければ、「情報Ⅰ」を学習した高校生が初めて受験する2024年度から導入される見込みである。

こうした中、全国的に大学、企業、自治体、NPO、民間教育事業者などの主催でプログラミング教室が開催されているほか、プログラミング教育用のソフトや工作キットなどの教材開発、コンテストなどが盛んに行われている。筑波学院大学（以下、本学）においても、2017年から茨城県県南生涯学習センターの要請により、小学生を対象にしたプログラミング講座を実施している。1日で150名規模の参加者があり、学生アシスタントを付けて4クラス編成で行っている。今後、さらなるプログラミング教育の需要増が見込まれ、プログラミング教育市場は拡大することが予想される。

背景の1つに、将来的に深刻なIT人材不足がある。経済産業省が2016年に発表したIT人材の最新動向と将来推計に関する調査結果⁵⁾では、「産業界での大型のIT関連投資」、「情報セキュリティ等に対するニーズの増大」、「IT利活用の高度化・多様化」、「我が国の労働人口（特に若年人口）の減少」などの要因により、国内のIT人材の不足が今後より一層深刻化する可能性があるとし、IT人材を確保することは、我が国にとってきわめて重要な課題であるとしている。

このような状況を考えると、高等教育におけるプログラミング教育の重要性は増しており、果たす役割は大きいといえるが、河村ら（2016）が2013年から2014年にかけて全国の大学に対して行った一般教育としての情報

教育（以下、一般情報教育）に関する調査⁶⁾では、3割弱の大学で授業内にプログラミング言語に関するトピックが取りあげられているものの、プログラミング教育を全く行っていない大学は67.1%で、科目全体でプログラミング教育を行っている大学は10.7%と低い結果になっている。さらに、低年齢層から始まるプログラミング教育は、大学入学時点での学生のプログラミングの知識やスキルに大きな格差をもたらすことが懸念される。

本学は、経営情報学部ビジネスデザイン学科という1学部1学科の文科系の大学であるが、履修コースは、ビジネスマネジメント、グローバルコミュニケーション、メディアデザイン、情報デザインの4つのコースがある。一般情報教育として1年次全学生を対象に、文書処理、表計算、インターネット、情報倫理などICTの技術を理解し活用するために必要な情報基礎科目を必修としているほか、プログラミングの入門科目「コンピュータ言語入門A（PHP）・B（マクロ）・C（JavaScript）」を1言語以上選択して履修する選択必修科目として開講している。これらの科目は、2010年度に現在の経営情報学部に変更した際に開設し、今年度で9年目になる。

筆者は、2015年度から「コンピュータ言語入門B（マクロ）」を担当しており、1年次全学生を対象にした一般情報教育としてのプログラミング教育のあり方について独自の実態調査を基に検討を行ってきた。新入生が入学時点でどのくらいのプログラミング経験があるのか、また、プログラミングに対してどのような意識を持っているのか等について、現状を把握するため、2015年から継続的に日本人新入生全員を対象にアンケート調査を実施している。さらに、筆者が担当している「コンピュータ言語入門B（マクロ）」の履修者を対象に、授業履修後の意識の変化や理解度などについてもアンケート調査を行っている。

本稿では、これらの調査結果を報告するとともに、授業の内容や取り組みについての有効性を検証し、高等教育（大学）における一般情報教育としてのプログラミング教育のあり方について考察する。

2. アンケート調査の概要と結果

堀越（2018）は、本学において新入生のコンピュータリテラシーに関する実態調査を実施し、高等学校でコンピュータを使って学習してきた内容や入学時点でのタイピングスキルおよびワープロ操作の習熟度などについて明らかにした⁷⁾。

今回は、新入生の大学入学時点でのプログラミング経験やプログラミングに対する意識について、また、筆者が担当している「コンピュータ言語入門 B（マクロ）」を履修した学生の履修後のプログラミングに対する意識の変化や理解度を中心に、2015年から現在（2018年）までの4年間の調査結果を報告する。なお、本アンケート調査は日本人学生に限定して実施した。

2.1 調査方法

調査は、入学前と学期末の毎年2回ずつ実施している。入学前の対象者は、3月下旬に行っている入学前教育に参加した新生であり、学期末の対象者は、1年次後期選択必修科目「コンピュータ言語入門 B（マクロ）」の履修学生である。

回答者数および実施時期を表1に示す。2018年度入学生の授業は現時点ではまだ終了していないため実施日を「予定」と表記している。

調査方法は、筆者が独自に作成したマクロによるアンケートシステムを利用して行っている。学生は、指定したフォルダから自分の学籍番号と名前のファイルを開いて回答する。アンケート調査画面の例を図1に示す。

表1 調査の回答者数と実施時期

調査年度	調査時期			
	1回目 入学前教育		2回目 学期末	
	回答者数	実施日	回答者数	実施日
2018年	111名	3月21日	60名	1月予定
2017年	110名	3月22日	42名	1月25日
2016年	80名	3月23日	26名	1月26日
2015年	86名	3月25日	33名	1月21日

図1 マクロによるアンケート調査画面の例

2.2 調査結果

2.2.1 大学入学時点での学習経験

まず、本学の新入生の高等学校でのプログラミング学習経験について調査した結果を図2に示す。プログラミングの学習経験があると回答した学生は全体の約12~15%にとどまり、調査を開始した2015年からの4年間においてはほぼ横ばいで、同じ割合で推移していることがわかった。

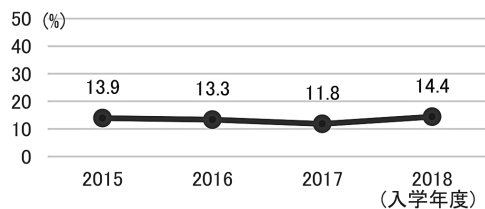


図2 高等学校でのプログラミング学習経験

最近2年間の新生が高等学校の情報の授業でパソコンを使って学習した内容については、ワープロ (Word)、表計算 (Excel) と回答した学生がそれぞれ80%前後で最も多く、続いて、プレゼンテーションソフトが60%弱、情報検索が25%弱、ホームページ作成が14~24%となっている (図3)。約8割の学生がワープロや表計算を学習しているのに対し、プログラミングの学習は1割強程度という低い結果である。

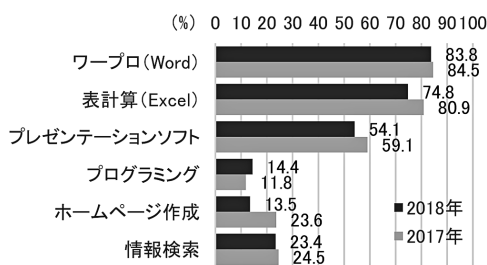


図3 高等学校でパソコンを使って学習した内容

2018年度の新生に対しては調査項目を増やしてさらに詳しく調べた。高等学校の授業以外でのプログラミング経験を含め、経験した場所や回数を調査した。プログラミングの経験があると回答した学生は36%で、内訳は70%が高校生のとき、25%が中学生のとき、5%が小学生のときであった (複数回答)。

高等学校在学時どこで経験したかの質問では、「授業で」と回答した学生が78%と最も多いが、それ以外の「部活」、「イベントや地域の講座」、「独学」、「その他」での経験者も20%程度みられた (図4)。その他 (自由記述) では「課題研究」、「ゲーム制作」、「専門学校のオープンキャンパスに参加して」などの回答もみられ、授業以外での学習経験や自らの興味・関心で経験する機会も増えている。

高等学校在学時に学習した回数については、3年間で1~3回が32%、4~10回が25%、11~20回が14%、21回以上が29%という結果であった (図5)。

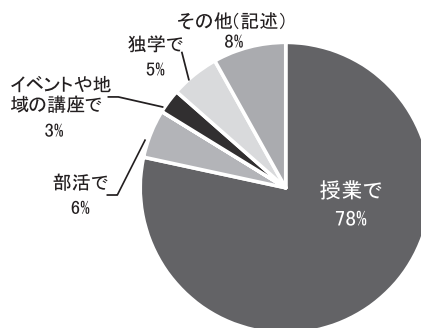


図4 高等学校在学時に学習した場所

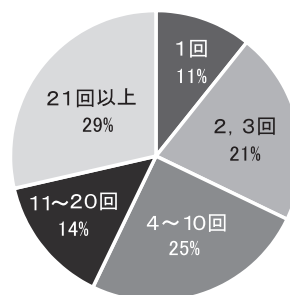


図5 高等学校在学時に学習した回数

学習した内容については、C言語、C#、C++、Java、VBAなどの一般的なプログラミング言語やロボット制御 (レゴなど) が多い。その他 (自由記述) では、比較的少ないコード量で簡単にプログラムが書けるといわれている「Python」やAndroid対応アプリケーションソフトウェアを開発するための「App Inventor」、専門的な「NC旋盤の切削プログラム」などの回答もみられた。また、学習した内容を「覚えていない」と回答した学生も2割程度いた (図6)。

中学校在学時に経験があると回答した学生は11人で全体の1割程度であった。経験した場所については8割以上が授業のときで、回数については2, 3回と回答した学生が半数以上であった。内容については、ロボット制御 (レゴなど) と回答した学生が3名いたものの、半数以上の学生は「覚えていない」と

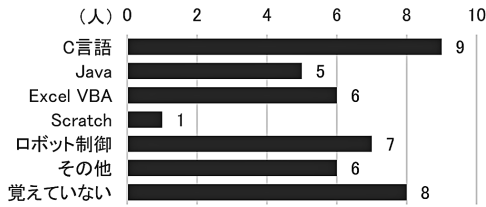


図6 高等学校在学時に学習したプログラミングの内容

回答している。

2012年度から中学校の技術・家庭科の授業で「プログラムによる計測・制御」という内容が必修になり、2018年度入学生は全員が経験しているはずであったが、このプログラミングの授業にあてられる時間は中学3年間で数時間程度であり、内容も各学校に任せられ、それぞれのICT環境や教育目標、指導体制などの実情に合わせて行われているのが現状のようである。経験があっても覚えていない、またはそれをプログラミングと理解していない学生も存在していることが推測できる結果となった。

小学校在学時に経験があると回答した学生は2名で、2名とも授業のときに2、3回学習したが内容は覚えていないと回答している。

2.2.2 大学入学時点での意識調査

次に、大学入学時点でのプログラミングに対する意識調査の結果を報告する。

プログラミングに興味があるかどうか調査したところ、「とてもある」や「少しある」と回答した学生は合わせて80%にのぼり、反対に「あまりない」や「まったくない」と回答した学生は僅か15%であった(図7)。また、将来、プログラミングをする仕事に就きたいと考えている学生は「とてもしたい」や「少ししたい」が合わせて58%、「あまりしたくない」や「まったくしたくない」が15%、「まだわからない」が27%という結果になった(図8)。

また、学習してみたいプログラミング言語を尋ねたところ、「ゲーム関連」が最も多く、続いて、「Web関連」となっている。その他の結果は図9のとおりである。

さらに、プログラミングという言葉聞いて何を連想するか、連想する言葉を5つずつ自由記述で回答してもらい、集計した結果、「難しい」が21人で最も多く、続いて、「ゲーム」19人、「C言語」18人、「ロボット」12人

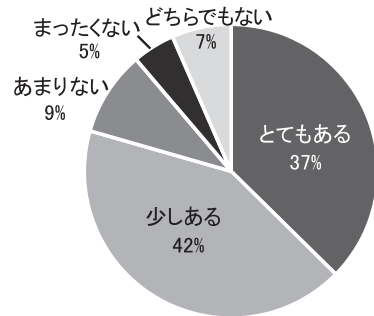


図7 プログラミングに興味があるか

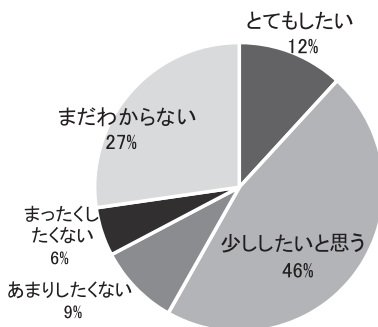


図8 将来、プログラミングの仕事がしたいか

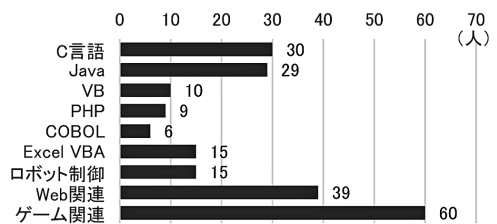


図9 学習したいプログラミング言語

などの言葉がみられた。また、「デバック」、「バグ」、「フローチャート」、「ループ」、「変数」、「for」、「if」、「while」、「return 0;」など、専門的な用語で学習経験者でなければ分からない言葉も複数あったほか、少数ではあるが、「憧れ」、「かっこいい」、「学びたい」、「できると楽しい」、「面白い」といった言葉もみられた。

プログラミングに対して高い興味・関心を示し、将来の仕事にしたいと考えている学生が多くいる一方で、「難しい」と感じている学生も多いことがわかった。

2. 2. 3 授業履修後の理解度と意識の変化

授業の内容については、3章で詳細を説明するが、筆者が担当する1年次後期選択必修科目「コンピュータ言語入門B（マクロ）」を履修した学生の履修後の理解度と意識の変化について調査した結果を報告する。

まず、授業で学習した内容については、8割以上の学生が「ちょうどよい」と回答している。「難しい」と回答した学生は2割弱で、「簡単すぎる」と回答した学生はゼロであった（図10）。また、マクロとは何か理解できたかの質問では、「よく理解できた」と「なんとなく理解できた」が合わせて9割以上となっている（図11）。さらに、マクロや他のプログラミング言語をもっと勉強したいか尋ねたところ、「勉強したい」または「まだわからない（どちらともいえない）」と回答した学生の割合は、調査した年によって差がみられたものの、「勉強したくない」と否定的な回答をした学生は毎年1割未満と非常に少ない傾向がみられた。（図12）。同様に、将来SEやIT関連の仕事がしたいかの質問でも、「したくない」と否定的な回答をした学生は2割未満と少数であることがわかった（図13）。

さらに、アンケートの自由記述の中には、「コードを書くのは難しかったがパズルを解

くようで楽しかった」、「プログラミング言語に興味を持てた」、「実際に動いた時が楽しかった」など「楽しい」という感想が多くみられたほか、「理解できた」、「達成感を味わえた」、「苦手意識がなくなった」などの意見もあった。

一方で、入学前と同様に授業後においてもプログラミングという言葉聞いて何を連想するか、連想する言葉を自由記述で回答してもらったが、こちらは、「楽しい」、「面白い」、「達成感がある」、「かっこいい」などの言葉もみられたものの、最も多い言葉は「難しい」で、入学前と変わらない結果となった。

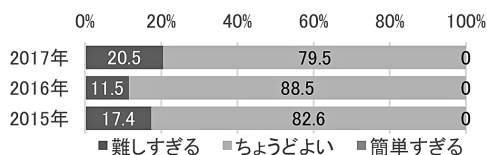


図10 授業で学習した内容について

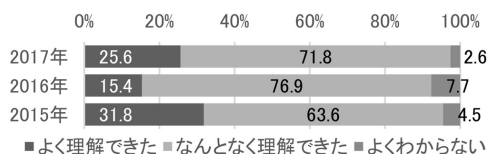


図11 マクロについての理解度

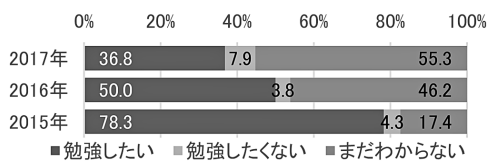


図12 プログラミング言語をもっと勉強したいか

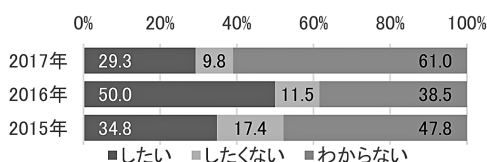


図13 将来SEやIT関連の仕事がしたいか

3. 本学のプログラミング教育

3.1 専門としてのプログラミング教育

本学は1学部1学科の文科系の大学であるが履修コースは4つあり、その中の1つである情報デザインコースでは、2年次以降の学年を対象にした専門基礎科目や専門発展科目として、情報分野の内容を幅広く学修できるカリキュラムを用意している。これらの科目は、他のコースを希望する学生でもコース間を跨いで自由に履修することが可能になっている。さらに、4年次で履修する卒業研究（必修、通年4単位）では、簡単なシステムやアプリケーションの開発、ゲームプログラムやWebシステムの作成などを通して、より専門的かつ実践的に修得できる環境が整っている。

情報デザインコースのカリキュラムの中から特にプログラミングに関連した科目を表2に示す。

表2 2年次以降対象のプログラミング科目

	プログラミング科目名
専門基礎	プログラミング (JAVA)、プログラミング (VB)、プログラミング (C #)、プログラミング (C 言語)、情報デザイン、組込コンピューティング、ビジュアルプログラミング、情報システム、フィジカルコンピューティング、アルゴリズム、ウェアラブルコンピューティング、Web デザイン A・B、アドバンストプログラミング、人間とロボットなど
専門発展	アプリケーションデザイン、システムデザイン、Web システム、IoT システムのデザイン、学習支援システム、情報デザインの研究、情報コンテンツの活用 など

3.2 一般情報教育としてのプログラミング教育

2年次からの専門教育に対して、1年次全学生を対象に行っている一般情報教育を表3に示す。

プログラミングの入門科目「コンピュータ言語入門A(PHP)・B(マクロ)・C(JavaScript)」を必修科目として開講している。この科目

表3 1年次全学生対象の一般情報教育科目

	授業科目名	開講期、形態	
入門科目群	必修	情報基礎 A (文書処理)	前期、演習
		情報基礎 B (表計算)	前期、演習
		情報基礎 C (インターネット)	前期、演習
		情報倫理	前期、講義
		経営と情報	後期、講義
	選択必修	コンピュータ言語入門 A (PHP)	後期、演習
		コンピュータ言語入門 B (マクロ)	後期、演習
		コンピュータ言語入門 C (JavaScript)	前・後期、演習
	選択	情報活用 A	後期、演習
		情報活用 B	後期、演習
その他	選択	情報科学入門	前期、講義
		情報技術と職業	前期、講義

は、3つの言語の中から好きな言語を1つ以上選択して履修する選択必修科目になっている。2年次以降、情報デザインコースに進む学生だけでなく、履修コースに関係なく全学生が修得する科目である。

この科目は、どれか1科目を履修することで履修要件を満たすが、実際には2科目または3科目すべての言語を履修する学生も多く、プログラミングに対する関心の高さを窺い知ることができる。図14は、在籍学生4学年分の履修科目数の割合を表したものである。2017年度入学生に至っては、半数以上の学生が2科目以上履修していることがわかる。さらに、2年次以降で履修する学生もいるため、2科目以上履修する学生の割合は実際にはもう少し高くなる可能性がある。なお、2018年度入学生の値は履修前の登録時点の数値であるため「見込み」と表記している。

また、毎年3言語の中では、Webアプリケーションの開発やゲーム開発のできるJavaScriptの履修者が一番多い傾向にある。これは、入学前の調査で、学習してみたいプログラミング言語として、「ゲーム関連」や「Web関連」を挙げた学生が多かったことと関連しているといえる（図9）。

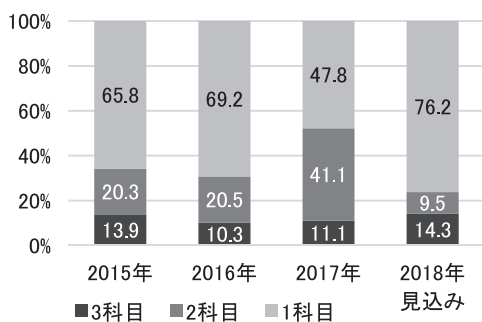


図 14 「コンピュータ言語入門」の履修科目数

3. 3 「コンピュータ言語入門 B (マクロ)」の概要

次に筆者が担当している「コンピュータ言語入門 B (マクロ)」の授業の内容や取り組みについて説明する。

3. 3. 1 授業の内容

プログラミングの入門科目で扱う教育用言語はいくつかあるが、マクロを作成するための言語である VBA (Visual Basic for Applications) の利点としては、①特別な動作環境を必要とせず、Microsoft Office があれば開発できる、②作成したプログラムが正しいか結果をすぐに確認できる、③構文が簡潔でプログラムの処理手順が理解しやすい、④ Office の延長として利用する可能性が高く実用的である、などが挙げられる。

表計算ソフト Excel は、今回の調査でも 8 割の学生が高等学校で既習しているという結果が得られているが (図 3)、多くの企業で使用され、学生が卒業後に実務の中で利用する可能性も高いソフトである。Excel のデータを効率的かつ効果的に処理するためのマクロを学習することは、将来仕事をする上でも有効と考え、実践的な内容を取り入れながら、マクロの活用法を身に付けさせている。

半期 15 回の大まかな授業構成を表 4 に示す。

まずは、マクロとは何か、どのようなこと

表 4 授業構成

回数	授業構成
1 回目	ガイダンス、マクロとは、活用例紹介
2～5 回目	マクロ記録の利用、変数とデータ型、繰り返し、条件分岐などプログラムの基本構造と VBA の基本的な知識とスキルを習得、アルゴリズムの理解
6～7 回目	実践問題
8～10 回目	Excel の関数やグラフ作成機能を利用したマクロの作成
11～13 回目	イベントやコントロールの活用
14～15 回目	応用問題 (実用的なシステムの作成)

ができるのかを活用例を紹介しながら理解させ、学習の動機付けを行う。次に、「マクロの記録」機能や例題を使ってマクロの基本操作に慣れさせる。最初からプログラムコードを記述させるのではなく、Excel のセルに入力した文字列を修飾するマクロや表のデータを並び替えるマクロを「マクロの記録」機能を利用して作成する。作成したプログラムコードを見せながら簡単に処理の流れやどのようにコードを記述するのかを説明し全体的な感じをつかませる。その後、マクロを作成するための言語である VBA の基本的な書き方や文法を学習する。また、順次、繰り返し、条件分岐などの基本的な制御構造や処理の流れ (アルゴリズム) を理解させる。イメージしやすいように扱うテーマはできるだけ身近で分かり易いものを使い、望む結果を得るために必要な処理の手順を 1 ステップずつ解説し理解させる。その「1 ステップずつ」をプログラミング言語で書けるように VBA の基本的な知識とスキルを習得させる。最初は例題を真似ることから始め、短いプログラムを組み合わせたり、一部を修正しながらプログラムを拡張させる。さらに、実践問題を使って繰り返し学習することで理解を深め、エラーの解決方法などデバッグの仕方にも慣れさせていく。また、それによって自分の理解度を確認させる。既に学習している Excel の関数やグラフ作成機能と組み合わせででき

るマクロや便利なイベントやコントロールの活用についても学習する。最終的には実用的なシステムの作成を通して、簡単なマクロを自らの力で作成できるようにする。1から考えるのが難しい場合は、穴埋め形式の未完成のプログラムを提示し、段階的にヒントを与えながら穴埋めの部分を考えさせ、できるだけ自分の力で完成できるように導いている。

毎回の授業の進め方は、まず、その日に学習する内容を説明する。次に、作成するマクロを実行して見せる。実行結果を得るために、手作業で行う場合の処理手順を考えさせる。その後、その処理手順をプログラムにするためにはどのようにコードを記述すればよいかを例文などを参考に考えさせ完成させていく。履修者全員が授業時間内に完成することを目標とし、早めに完成した学生には、遅れている学生やエラーを出している学生のサポートをさせる。

さらに、毎回授業内容に即した課題を出し、復習しながら繰り返し行うことで理解を促す。多くの演習を行い、デバッグ作業を通して、問題を解決する力や論理的に考える力を養い、実践力を付けさせていく。

1年次全学生が履修する入門科目としては、苦手意識を持たせず、プログラミングの楽しさや面白さを知ってもらいたい。また、自分にもできたという成功体験、達成感を味わわせることやプログラムは便利で役に立つものであることに気付かせることも重要であると考え、その点に留意して授業を行っている。将来、専門的に学びSEなどのプログラム開発者への道に進む学生だけでなく、開発者にならなくても一般実務の中で自分の仕事を効率よく行うためのプログラムを作成できるようにしてもらいたいと考える。

テキストは、実教出版の『30時間でマスター Excel 2007 VBA』を使用している。その他、適宜自作の配布資料を使う。VBAの書き方や文法およびアルゴリズムの解説資料

や課題の解説付き解答は、図解を使って直感的にわかりやすく作ることを心掛けている(図15)。ネットワーク上の共有ドライブにも保存しいつでも参照できるようにしている。

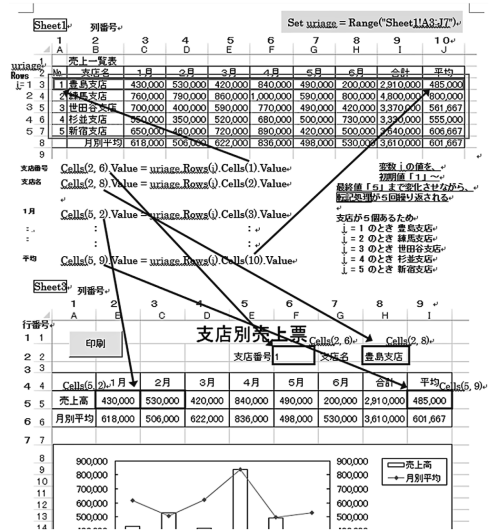


図 15 自作の配布資料の例

3.3.2 授業での取り組み

授業で特に工夫している点や留意している点を2つ挙げる。

(1) プログラミングの動機付け

プログラミングを学習する動機付けを行うためには、学ぶ意義を理解させる必要がある。プログラムは、自分を助けてくれる便利なツールであることに気付かせることが重要であり、役立つ例を数多く示すことで、プログラムを使うことで何ができるのかを考えさせ、自ら必要なプログラムを創造する力や自分の身の回りのことや実社会で役立てる実践力を養う糧にしてもらいたいと考え、以下の取り組みを行っている。

- ・学習する内容は理解し易いようにできるだけ身近なテーマを選ぶ
- ・実用的な例を紹介する
- ・学習した内容を応用する
- ・簡単な短いプログラムの組み合わせを利用

する

具体的には、学習した内容をどう活用するかについて、折に触れて説明しながら実習に取り入れている。実践例を2つ挙げる。

1つ目は、セルを色分けするプログラムを考えると、色番号を調べるための色見本を作成した。図16がプログラムと実行例である。繰り返し（For 文）を使って1から50番までの色でセルを塗りつぶし、対応する色番号を表示するプログラムである。実行すると一瞬で50色が出てくる。僅か6行の簡単なプログラムでも役立つものが作れることに、実行後、興味を示す学生も多い。

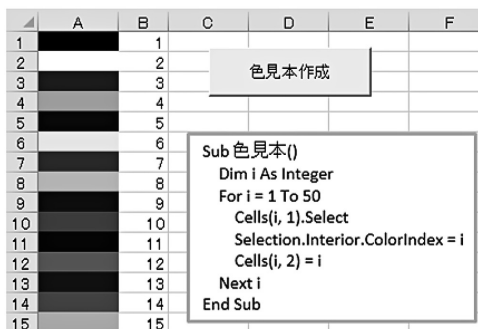


図 16 色見本作成マクロのプログラムと実行例

2つ目は、学生に対して実施したアンケート調査にマクロを利用した。授業後半で、Userform の利用およびイベントやコントロールについて学習するが、それらを応用した例である（図17）。Userform 上に、Label（質問文）、OptionButton（択一）、CheckBox（複数回答）、TextBox（自由記述）、CommandButton（保存）を配置するだけで、簡単にアンケートシステムが作成できる。実行結果は Excel に取り込み、全データを一つの表にまとめ、結果を集計し、グラフ作成するまでを短時間で効率よく作業できることや一度作ったマクロを毎年何度でも利用できることを説明する。画面構成を工夫するだけで、受付時の名簿作成など様々なことに利用

できる例も紹介している。

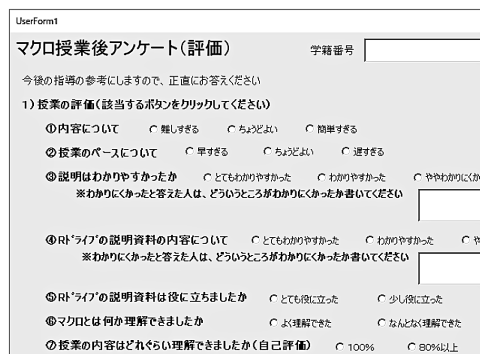


図 17 マクロによるアンケートの例

(2) 達成感を与える

成功体験や達成感を与えることは、学生のプログラミングに対するモチベーションを上げるためには必要不可欠であり、次のようなことに留意しながら授業設計を行っている。

- ・ 技術的な到達目標は欲張らず、基本的な内容だけ教える
- ・ 学生同士が助け合って演習を行う
- ・ 穴埋め形式の未完成プログラムを利用し、できるだけ自分の力で完成できるように導く

半期15回で学習できる内容には限りがあるため、技術的な到達目標は欲張らず、繰り返し、条件分岐など基本的なプログラムの制御構造を中心に、最初は例題を真似ることから始め、短いプログラムを組み合わせたり、一部を修正しながらプログラムを拡張し、実行結果がどのように変化するかを確認させながら、同じようなプログラムを繰り返し作成することで慣れさせていく。また、Excel の関数やグラフ作成機能と組み合わせることで応用力を高め、達成感が得られるよう工夫をしている。

また、新入生の8割以上がプログラミング未経験者であり（図2）、授業の履修者数は毎年2クラス編成で50名を超えている。こうした中、演習を行うと、早く完了する学生と

そうでない学生の差が大きいのが現実である。早く完了した学生に発展課題を用意するなどの対策も考えられるが、これは毎回の授業以外の課題の中で行い、授業中の演習については履修者全員が授業時間内に完成することを目標とし、早めに完成した学生には、遅れている学生やエラーを出している学生のサポートをさせる工夫をしている。

さらに、応用問題は最初から自力で考えるのが難しい学生も多いため、穴埋め形式の未完成のプログラムを提示し、段階的にヒントを与えながら穴埋めの部分を考えさせ、できるだけ自分の力で完成できるよう導く工夫をしている。穴埋め形式のプログラム例を図18に示す。

店名	請求金額	請求期間	支払日
1 大商店	w1,250,000	2009/08/29	
2 小商店	w2,250,000	2009/07/29	2009/07/29
3 C商店	w999,000	2009/09/29	
4 D商店	w3,456,000	2009/08/29	
5 E商店	w999,000	2009/07/29	
6 F商店	w2,190,000	2009/07/29	
7 G商店	w3,500,000	2009/08/29	2009/08/29
8 H商店	w3,900,000	2009/07/29	
9 I商店	w4,000,000	2009/09/29	2009/09/29
10 J商店	w1,345,000	2009/09/29	

```

Sub 請求書 (請求書引)
    Dim 請求書 As Range, i As Integer
    Set sitem = Range("Sheet1!A10:A14")
    For i = 1 To sitem.Rows.Count
        Dim 請求金額 As Integer, 請求期間 As Date, 支払日 As Date
        請求金額 = sitem.Cells(i, 2).Value
        請求期間 = sitem.Cells(i, 3).Value
        支払日 = sitem.Cells(i, 4).Value
        If 請求金額 > 1000000 And 請求期間 <= Date Then
            支払日 = Date + 30
        End If
    Next i
End Sub
  
```

図 18 穴埋め形式の未完成プログラムの例

4. 考察

今回の調査結果から、新入生がプログラミングに高い興味・関心を持っているものの(図7、図8、図14)、大学入学時点でのプログラミング経験者は毎年全体の僅か15%未満と低いことが明らかになった(図2)。布施・岡部(2016)の調査⁸⁾では北海道大学においても2割弱という同様の結果があり、本学だけが特別低い値でないことがわかる。さらにこの少ない経験者の中には学習した内容を覚えていない学生や学習した回数が1~3回と

回答した学生が約3割いることも明らかになり、これらの学生は、単に「体験」程度の学習内容であったことが推測できる(図5、図6)。この学習経験のあるなしの違いや、学習した回数に差がみられるのは、本学の学生の出身校が普通科以外に情報系、工業系、商業系、通信制など多様化していることによるものと考えられる。

こうした現状を考えると、高等教育においてもまだまだ基本的な内容から教える必要があるといえる。「コンピュータ言語入門B(マクロ)」を履修した学生の中には、授業で学習した内容について8割以上が「ちょうどよい」、2割弱が「難しかった」と回答しており、現時点では「簡単すぎる」と回答した学生は一人もいなかった(図10)。

しかし、今後、初等中等教育においてプログラミング教育が推進されることを考えると、大学入学時点でのプログラミングスキルの格差が大きくなることも予想され、入門教育とはいえ、経験者に対してはより高度な内容を教えるなど、新入生のプログラミングスキルのレベルに対応した授業設計をする必要が出てくる可能性もあり、実態調査を継続しながら学生の状況を把握していく必要がある。

小学校でプログラミング教育が必修化される目的は、プログラミングを体験することによってプログラミング的思考、つまり、論理的に考える力や創造力を養うことにあるとしている⁹⁾。パソコンを使う場面だけでなく、望む結果を得るためには何を、どのように、どういった順番で行えばよいのかと論理的に考える力や問題解決力、創造力は普遍的なものであり、低年齢層から体系的に行うことでより効果的に身に付けられる。将来、社会で活躍するためには不可欠な能力であり、プログラミング教育を通してこれらの能力を育成することは高等教育においても重要な役割の1つであるといえる。

さらに、高等教育における一般情報教育としてのプログラミング教育には、「プログラミングを学習する動機付け」や「基本的なプログラムの構造を理解させ、実践で役立つプログラミングスキルを修得させる」という役割があると考えられる。

1年次全学生が受ける入門教育では、まず、プログラミングを学習する意義や必要性を理解させ、動機付けを行うことが重要である。そのためには、苦手意識を植え付けないことや自分にもできるという自信を持たせること、達成感を味合わせプログラミングは楽しく面白いものと感じてもらうことが大切である。また、2年次以降に専門的な内容を学修することを希望し、将来プログラム開発者への道に進む学生はもちろんだが、それ以外の学生にも、プログラムは便利で役に立つツールであり、身近なことに利用できる応用性の高いものであることを理解させることが有効である。そして、基本的なプログラムの構造を理解させ、実践で役立つプログラミングスキルをしっかりと修得させることが重要であると考えられる。

授業後の調査では、プログラミングに対して「難しい」と感じている学生が2割程度いたが、8割以上の学生が学習内容はちょうどよい、9割以上の学生がマクロについて理解できたと回答している(図10、図11)。「プログラミング言語をもっと勉強したいか」、「将来SEやIT関連の仕事がしたいか」の質問では、「まだわからない」と回答した学生が多くいたものの、「したくない」と否定的な回答をした学生は少数であった(図12、図13)。さらに、アンケートの自由記述の中には、「役に立つ知識が得られた」、「プログラミング言語に興味を持った」、「苦手意識がなくなった」、「理解できた」、「達成感を味わえた」などの意見があったほか、「コードを書くのは難しかったがパズルを解くようで楽しかった」、「実際に動いた時が楽しかった」な

ど「楽しい」という感想も多くみられた。ある程度の学生に対しては、動機付けが上手くできていると推察する。

これらは、3章で述べた授業での取り組みの成果でもあると考える。毎回の授業で、履修者全員が授業時間内に課題を完成させることを目標にし、早めに完成した学生には、遅れている学生やエラーを出している学生のサポートをさせる工夫を行っているが、これは、格差が広がることにより、遅れている学生に苦手意識を持たせないための配慮でもあるが、この取り組みはサポートする側の学生にとっても理解を深めるよい機会になっている。中には得意になってサポートしている学生の姿も見られ、こうした行為が面白さや自信にも繋がっていると考える。さらに、図15に例を示した自作の配布資料についても「わかりやすかった」や「役に立った」という意見が多くみられ、学習効果の向上に役立っているといえる。

一方で、期末試験の結果を見ると、授業中の演習で正しくプログラムを実行することができていても、必ずしも理解できているとはいえない学生もいることがわかる。期末試験は、テキストの持ち込みを許可し、変数の扱いやセルやSheetの参照、プログラムの基本構造を理解しているかを確認するため、学習した基本的なコードを記述させる問題や比較的短いプログラムの穴埋めおよびFor文などの実行後に得られる結果を書かせる問題などを机上で筆記させる形式で行っている。満点を取る学生は1割強で、概ね理解できていると判断できる学生も半数以上はいるものの、自分の力でプログラムを作成できるだけの力を身に付けていないと判断できる学生も存在する。小テストなどを早い時期から段階的に行い、理解度を確認しながら授業を進めるなどの改善が必要であり、今後の課題とした。

プログラミングの入門教育において扱うプ

プログラミング言語についてはいくつか考えられるが、河村ら（2016）が2013年から2014年にかけて全国の大学に対して行った一般情報教育に関する調査⁶⁾では、プログラミング演習で使用されているプログラミング言語の中で最も多いのがVBAの18.4%、次いでC言語17.3%、Java 9.2%、VB 8.2%、Ruby 6.1%、JavaScript 5.1%という結果がある。VBAが多い理由としては「Microsoft Office アプリケーションの操作方法に関する教育との関連・連続性があるものと想像できる」としている。

筆者が行った調査でも、表計算ソフトExcelは、8割の学生が高等学校で既習しているという結果が得られている（図3）。本学では、1年次前期必修科目「情報基礎B」で全員が学修する。また、多くの企業で利用され、学生が卒業後に実務の中で活用する可能性の高いソフトである。Excelのデータを効率的に処理するためのVBAを学習することは将来仕事をする上でも有効と考える。実際に複数の卒業生からも仕事でVBAを利用しているという話を聞くことがある。実行環境においても、特別な環境を必要としないため、表計算ソフトExcelがあれば開発できる手軽さもある。学生にとってもOfficeの延長として利用しやすい言語であると考えられる。さらに、情報処理学会が行った情報教育の在り方に関する調査研究¹⁰⁾では、教育用言語において学生の達成度レベルが高いプログラミング言語は1位がC言語で2位がVisual Basic/VBAとなっている。授業後の調査でも、マクロについて「理解できた」と回答した学生は9割を超えている（図11）。構文が簡潔でプログラムの処理手順が理解しやすいことや作成したプログラムが正しいか結果をすぐに確認できるなど、1年次全学生を対象にしたプログラミングの入門教育としては扱いやすく適した言語であるといえる。

5. おわりに

2015年から継続的に実施している新入生のプログラミング経験や意識についてのアンケート調査結果を報告し、それを基に授業の内容や取り組みについての有効性を検証し、高等教育における一般情報教育としてのプログラミング教育のあり方について考察した。

大学入学時点でのプログラミング経験者は毎年僅か1割強にとどまり、さらに、経験のある学生でも情報科や商業科出身の一部の学生を除いては「体験」した程度であり、プログラムを作成できるだけのスキルを身に付けている学生は少ないことが明らかになった。また、プログラミングに対しては、興味・関心は高いものの、「難しい」と感じている学生が多いこともわかった。

現時点では、高等教育におけるプログラミング教育でも、基本的なことから教える必要がある。そして、プログラミングの経験を通して、普遍的な論理的思考や問題解決力、創造力を養うことも大切であるが、プログラミングを学ぶ必要性を理解させ、実社会で活用できるだけのプログラミングの知識とスキルをしっかりと修得させることが重要であると考える。

今回の調査では、授業での取り組みの成果も確認できたが、課題も見えてきた。

今後も、初等中等教育で推進されるプログラミング教育の動向および情報化社会の進展やニーズを注視し、継続的に実態調査を行い学生の状況を把握しながら、現状に合ったプログラミング教育を検討していきたい。

参考文献

- 1) 文部科学省：新学習指導要領（本文、解説、資料等），2017年3月，http://www.mext.go.jp/a_menu/shotou/new-cs/1383986.htm，(accessed 2018-05-21)
- 2) 文部科学省：中学校学習指導要領（平成29

- 年告示) 解説 技術・家庭編, 2017年7月, http://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afieldfile/2018/05/07/1387018_9_1.pdf, (accessed 2018-05-21)
- 3) 文部科学省: 高等学校学習指導要領の改訂のポイント, 2018年4月, http://www.mext.go.jp/a_menu/shotou/new-cs/__icsFiles/afieldfile/2018/04/18/1384662_3.pdf, (accessed 2018-05-21)
 - 4) 首相官邸: 第16回未来投資会議, 2018年5月17日, https://www.kantei.go.jp/jp/98_abe/actions/201805/17mirai_toshi.html, 第16回未来投資会議 議事要旨, <http://www.kantei.go.jp/jp/singi/keizaisaisei/miraitoshikaigi/dai16/gijiyoushi.pdf>, (accessed 2018-08-21)
 - 5) 経済産業省: IT人材の最新動向と将来推計に関する調査結果, 2016年6月, <http://www.meti.go.jp/press/2016/06/20160610002/20160610002.html>, 報告書概要版, <http://www.meti.go.jp/press/2016/06/20160610002/20160610002-7.pdf>, (accessed 2018-08-21)
 - 6) 河村一樹ほか: これからの大学の情報教育, 日経 BP マーケティング, 2016年3月
 - 7) 堀越真理子: 実態調査に基づく一般情報教育の検討 -1年次前期必修科目「情報基礎A(文書処理)」の取り組みを通して-, 筑波学院大学紀要第13集, pp75-88 (2018)
 - 8) 布施 泉, 岡部成玄: 高等教育の一般情報教育におけるプログラミング教育—北海道大学の実践を通して—, 高等教育ジャーナル—高等教育と生涯学習—23, pp53-63 (2016)
 - 9) 文部科学省: 小学校プログラミング教育の手引(第一版), 2018年3月, http://www.mext.go.jp/b_menu/houdou/30/03/1403192.htm, (accessed 2018-08-24)
 - 10) 情報処理学会: 超スマート社会における情報教育の在り方に関する調査研究, [文部科学省先導的・大学改革推進委託事業] 平成28年度報告書, 2017年3月, http://www.mext.go.jp/a_menu/koutou/itaku/__icsFiles/afieldfile/2017/06/26/1386892_1.pdf, (accessed 2018-08-24)