

プログラミングに苦手意識を持つ学生に対する プログラミング教育の取り組み

佐々木 整^{*1}, 岡本 俊一^{*1}

^{*1} 拓殖大学工学部

An Approach to Programming Education for Students who Feel that It Is Their Weak Point

Hitoshi Sasaki^{*1}, Shunichi Okamoto^{*1}

^{*1} Faculty of Engineering, Takushoku University

情報工学を専攻する学生の中にも、プログラミングに苦手意識を持つ学生が少なからず存在している。プログラミングを含む、情報工学に関する知識や技術を習得するために情報工学科に入学したのにもかかわらず、苦手意識を持つ学生は学年を追うごとに増加し、苦手意識自体も強くなっている傾向が見受けられる。本発表では、このようなプログラミングに対して苦手意識を持つ情報工学科の3年生に対するプログラミング教育の取り組みについて報告する。

キーワード: 苦手意識, プログラミング教育, プロトタイピングツール, ハイブリッドアプリケーション開発

1. はじめに

文部科学省が平成26年に実施した「諸外国におけるプログラミング教育に関する調査研究」によると、諸外国ではプログラミング教育を初等教育から導入しようとする動きが見られ、イングランドではコンピュータサイエンス、情報技術、コンピュータリテラシーの三分野を、小学校からの系統的学習に取り組んでいること等が報告されている^(1,2)。また、経済産業省が平成28年に発表した「国内IT人材の最新動向と将来推計に関する調査結果⁽³⁾」では、IT人材は現時点で17万人超が不足しており、今後人口減少に伴い深刻化すると予測している。さらに、各国と比較して管理職クラスの割合や理系出身者が少ない傾向があり、2030年にはIT人材数が78.9万人不足すると予測しており、IT人材の育成や確保に向けて、女性やシニア、外国人が活躍できる環境作りや個々のスキルアップ支援の強化、処遇やキャリアなどの改善による魅力アップ、情報セキュリティ、先端分野、起業家などの重点的な育成強化が必要と報告している。

このような背景もあり、初等中等教育におけるプログラミング教育の取り組みやその報告等が、本学会の大会や研究会を初めとして、数多くなされている。これらの取り組みは情報教育の枠組みの中で行われることが多かったが、最近では「国語」と関係付けられた実践報告⁽⁴⁾もなされるなど、更なる広がりを見せている。高等教育においても、専門教育に限らず一般情報教育の中でプログラミングが取り上げられており⁽⁵⁻⁸⁾、初等教育から高等教育までプログラミング教育に関する取り組みが現在も行われている。

その一方で、専門教育としてプログラミングを学ぶ者の中には、プログラミングに対する苦手意識を持つ者が少なからず存在していることが参考文献⁽⁹⁾等で報告されており、エデュテイメントの導入⁽¹⁰⁾等の工夫でこの苦手意識を改善させていこうとする取り組みがなされている。

本稿では、本学情報工学科3年生を対象にした、プログラミングに対する苦手意識の軽減に向けた取り組みについて報告する。

2. プログラミング教育の概要

2.1 プログラミング関連科目とコース

本学科では、学生が2年時以降に「システムエンジニア(SE)」、「コンピュータソフトウェア(CS)」、「インターネットサービス(IS)」、「組み込みシステム(ES)」、「ゲーム&シミュレーション(GS)」、「ITコーディネーション(IC)」、「テクニカルライティング(CW)」の7つのコースのいずれかを選択することになっており、プログラミング教育に関する科目は、このコースによって必修科目となるか選択科目となるかが異なり、選択科目でも指定された科目群から指定単位以上履修する必要のあるもの(コースコア科目と呼ぶ)と、そうでないものがある。3年前期までのプログラミング関連科目と必修、選択の違いを表1に示す。なお、表1中の「必修」は必修科目、「コア」はコースコア科目、空白は選択科目を表しており、「プログラミング基礎」、「データ構造とアルゴリズム I」、「データ構造とアルゴリズム II」は講義科目(座学)、それ以外は演習科目で2コマ連続の実施である。また、「プログラミング I」は「プログラミング II」の前提科目、「プログラミング II」は「応用プログラミング」と「オブジェクト指向プログラミング」の前提科目となっており、それらを合格していなければ履修することができない。なお、「プログラミング I」、「プログラミング II」、「応用プログラミング」は、それぞれ前後期の2回開講している。一方の学期で不合格となった学生が次の学期に同じ科目を履修する事ができるようにすることで、なるべくプログラミングを行わない期間が生じないように配慮している。

2.2 プログラミングI・IIの講義内容

本稿では、インターネットサービスコースの必修科目での実践について報告するので、インターネットサービスコースで必修科目に指定されているプログラミングIとプログラミングIIについて、講義目的などについて以下に述べる。

2.2.1 プログラミング I

Javaの短く簡単なプログラムの作成実習を通して、プログラムの編集・コンパイル・実行・デバッグからなるソフトウェア開発の基本的な流れを理解することを授業の目的としている。変数への値の代入、値の表示と入力、条件分岐と単純な繰り返しを用いた定型の簡単なプログラムを作成する能力を身につけるために、エディタによるプログラムの編集から、for文による単純な繰り返しまでを学ぶ。

2.2.2 プログラミング II

オブジェクト指向プログラミングの概念を学ぶとともに、データ構造、アルゴリズムの設計に関する考え方を習得することを授業の目的とし、具体的には、次の各項に該当する総合的なプログラミング能力の習得を到達目標としている。

1. 問題に適したデータ構造を設計する能力
2. 正しい結果が得られる効率的なアルゴリズムを設計する能力
3. 設計したデータ構造とアルゴリズムをプログラミング言語で記述する能力
4. 作成したプログラムの正しさをテストし、誤りを発見し、修正する能力

表 1 コースとプログラミング関連科目の関係

学年	科目名	SE	CS	IS	ES	GS	IC	CW
1	プログラミング I	必修	必修	必修	必修	必修	必修	コア
1	プログラミング II	必修	必修	必修	必修	必修	必修	コア
2	プログラミング基礎							
2	オブジェクト指向プログラミング	必修	必修					
2	応用プログラミング		コア	コア	コア			
2	データ構造とアルゴリズム I	必修	必修	コア	コア	コア		コア
2	データ構造とアルゴリズム II	コア	必修					
3	プログラミング言語	コア	コア					

しかし、実際には少なくとも演習全体の 1/3 (5 回) をプログラミング I の復習として、データ型と変数や演算子と文字列操作についての復習を行い、残りの時間をメソッドやクラス、GUI プログラミングの演習に充てている。

このように、プログラミング I とプログラミング II は順序関係のある別の科目であるものの、重なりのある部分を多くすることによって、プログラミングに関する基本的な事項の定着を図る工夫を行っている。

2.2.3 プログラミングに対する意識調査

これまでに述べたとおり、プログラミング教育に様々な工夫を行い多くの時間をかけて実施している。その効果が十分認められる学生も少なくないが、学生の一部には、プログラミング教育の結果として苦手意識が芽生えてしまい、学習意欲が低下したり、学習そのものを諦めてしまったりしている (9,10)。

平成 26 年度と 27 年度にプログラミング II を履修した学生 159 名を対象に、著者らがプログラミング

II の初回の授業で行ったアンケート調査では、「プログラミングを面白いと思うか」という質問に対して、面白くないとの回答が 5.0%、あまり面白くないとの回答が 20.1%と、1/4 の学生が面白いとは感じていないことが分かった。また、「プログラミングは難しいと感じるか」という質問に対しては、49.7%が難しい、45.3%がやや難しいと回答しており、ほとんどの学生が簡単ではないと感じている。さらに、「自分のプログラムの実力はどの程度あるか」という質問では、22.6%が「ほとんどプログラムを書くことができない」と回答している。

それぞれの質問項目に対する回答理由（自由記述）では、「自分で書いたプログラムが正常に動作した時が楽しい」というような、達成感を得ている事が分かる回答も一定数あるものの、上記の質問項目に共通して「何をすればよいか分からない」、「考えるのが面倒くさい」、「プログラミング自体が面白くない」という主旨の回答が複数の学生によってなされている。また、「どんなプログラムが書けるようになりたいか」という質問に対しては、「ゲーム」や「人の役に立つもの」という回答が多い中で、「動く理由をしっかりと理解したプログラムが書きたい」や「参考資料を見ないで書きたい」、「プログラムを書けるようになる将来が想像

できない」というような回答も、それぞれ複数あった。

3. 情報サービス演習の概要

3 年後期にインターネットサービスコース(IS)の必修科目として開講される「情報サービス演習」(2 コマ連続授業)で、2. で述べたような学生のプログラミングに対する学習意欲の低下を食い止め、苦手意識を払拭させることを目指した取り組みを行った。そのために、次の事を考えて演習を設計した。

1. 自身の成長を実感する
2. 知識や能力、経験が不足していても参加できる
3. 責任と適度な達成感を感じる

以下に、演習の詳細について報告する。

3.1 演習内容

情報サービス演習での演習内容を表 2 に示す。

情報サービス演習と同様に、インターネットサービスコース(IS)の必修科目である「Web アプリケーション&サービス」(2 年次開講)で、HTML や JavaScript と各種のライブラリ、CSS は既に学習済みであるので、1~3 回目では、JavaScript、CSS、jQuery の基礎的な項目について復習を行いながら、確認のための演習を行う。4,5 回目で、ハイブリッドアプリケーション開発環境である Monaca^(11,12)を使い、まずは Web アプリケーションとしての開発を行った後に、Monaca デバッガで学生が所有しているスマートフォンでそのアプリケーションを動作させる。さらに、6 回目では Monaca を利用して、モバイル向けの UI フレームワークである Onsen UI 2⁽¹³⁾とそれを利用した SPA(Single Page Application)の作成方法について学ぶ。7 回目で、これまでに学んだ内容を踏まえて、学生がそれぞれ自分のスマートフォンで動作するアプリケーションを 1 つ作成する。

8 回目から 11 回目はアプリケーション開発から一旦離れ、Prott⁽¹⁴⁾を用いてプロトタイピングの演習を行う。8 回目では Prott の操作方法を学び、9 回目で 3 人一組のチームを構成し、プロトタイプを作成するための仕様を検討し、その結果を他の学生に向けて発表する。10 回目で、他のチームに 9 回目で発表した仕様

に基づくスマートフォンアプリケーションの開発を依頼する。依頼されたチームはその仕様に基づいて Prott でプロトタイプを作成する。11 回目では、全体に対してプロトタイプの報告・議論を行い、依頼したチームとの最終調整を図る。12～14 回目では、Prott で作成したプロトタイプを、Monaca で Onsen UI 2 を使いながら実装する。14 回目では、開発したアプリケーションのプレゼンを行い、発注したチームへの納品を行う。15 回目では実装したスマートフォンアプリケーションの紹介の Web ページを各学生が作成する。1 つのスマートフォンアプリケーションに対して 3 つの紹介ページができることになるが、それぞれチーム内での自分の役割や、他のメンバーの取り組む様子などもレポートさせるので、記載内容の重複は多くない。

なお、6 回目と 8 回目は、それぞれの開発元のエンジニアに講義をお願いしている。

表 2 情報サービス演習の内容

回	内容	備考
1	JavaScript の基礎	
2	CSS の基礎	
3	jQuery の基礎	
4	Monaca 入門(基本操作)	
5	Monaca 入門(アプリケーション開発)	
6	Onsen UI 2 入門	外部講師
7	オリジナルアプリケーション開発	
8	Prott 入門	外部講師
9	仕様作成とプレゼン	チーム
10	Prott によるプロトタイプ作成とコメント作成	チーム
11	プロトタイプのプレゼン・再検討	チーム
12	Monaca による実装(1)	チーム
13	Monaca による実装(2)	チーム
14	Monaca による実装(3)とプレゼン	チーム
15	アプリケーション紹介 Web ページの作成	

3.2 演習の特徴

この演習の特徴について、以下の観点から説明する。

3.2.1 スマートフォンでの動作確認

プログラミング II では Swing や JavaFX を使い、GUI アプリケーション開発なども行っているが、表 1 で示したプログラミング関連科目の多くは PC のターミナルや Eclipse 等の IDE のターミナルで動作を確認するものであった。一方、現在の学生にとっての最も身近なプログラムは、スマートフォンで動作するものである。そのため、自分が現在学んでいる内容と自分が現在使っているものが関連づけることが難しい。

そこで、スマートフォンでアプリケーションを動作させることで、今までに学習したことがスマートフォンアプリケーションの開発に役立つ、という事を実感させる事を考えた。しかし、そのためには、iOS であれば Swift や Xcode、Android であれば Java や Android Studio のような、スマートフォンアプリケーションを開発するための更なる知識や技術が必要となる。既にプログラミングに対する苦手意識を持っている学生に対して、新たにこれらの教育を行う事は現実的ではなく、現時点での知識や経験の範囲内か、それに少しだけ新しい知識を加えることだけでスマートフォンアプリケーションが開発できることが望ましい。また、学生所有のスマートフォンがどの OS を採用しているかで、開発言語や開発環境が異なる状況は望ましくなく、それらに依存せずにアプリケーションが開発できる必要がある。この条件を満たすものとして、著者らは Cordova を利用し Web 技術でスマートフォンなどプラットフォームに依存しないアプリケーションを、Web ブラウザで開発できる Monaca に着目した。これによって、学生は既学習の HTML5、JavaScript、CSS で、特別な環境などを用意することなく、スマートフォンアプリケーションを作成できるようになる。

3.2.2 プロトタイピングツールの活用

アイデアを形にするプロセスに重点を置き、プログラミングが得意ではないからといって、アプリケーション開発の全てを諦めてしまうようなことにならないように務めた。基本的なコンセプトから UI の設計と評価までを経験することで、アプリケーション開発の魅力を実感し、学習意欲が高まるものと考えている。しかし、そのためには 3.2.1 と同様に、実際に学生が

身近なもの・リアリティのあるものとして感じられるように配慮する必要がある。つまり、コンセプトに基づく UI 設計の結果を、学生自身のスマートフォンでプログラミングすることなく表示させ、それを利用して利用者の立場に立った UI の評価と改良の検討が行えることが重要である。

そこで、プロトタイピングツール **Prott** を使用することとした。**Prott** を使用したプロトタイプ作成は、次の手順で行う。

1. 作成したいスマートフォンアプリケーションの各画面をノートに手書きで描画
2. それらの画面を学生自身のスマートフォンのカメラで撮影
3. Web ブラウザでそれぞれの写真をつなぐ
4. 専用アプリケーションを使い、学生自身のスマートフォンでプロトタイプの動作チェック
5. 不都合があれば、1. または 3. に戻る

3.2.3 他のチームへの発注とコメント

アイデアを言語化して他人に伝えることは、プログラム開発においてもとても重要なことである。しかし、自分で考えたものを自分で開発する場合は、この言語化のプロセスが発生しないか、レポート作成時に発生することになる。しかも、レポート作成時の言語化は、完成したものに合わせがちになるため、元々どのようなものを作成しようとしたか、ではなく、締切日までにどこまでできたか、ということが言語化されてしまう。一方、他人によって言語化された内容（アイデアや仕様）を正しく理解するだけでなく、その内容の不足や矛盾などに気づくことや、理解に基づいた提案ができることも大変重要である。

これらを経験させるために、チームでスマートフォンアプリケーションとして実現したい1つのアイデアを考えさせ、そのアイデアを別のチームに伝えて開発を依頼するという取り組みを行った。依頼元のチームは依頼先のチームに、どのようなスマートフォンアプリケーションを開発したいか、各自の言葉で伝えなければならず、依頼先のチームはそれに基づいて3.2.2で述べたプロトタイプを作成する。そのプロトタイプに対して、依頼元のチームと教員・TAはコメント

を付ける。付けられたコメントに対して、依頼先のチームは再検討や、必要に応じて依頼元チームとの打合せを行う。

この作業を数回繰り返すことで、他人にアイデアを正確に伝えることの難しさや、依頼側の意図を把握することの難しさ、意見交換の重要性を認識するとともに、個人でなくチームとしての意志や見解の統一の必要性を体験することを期待している。

また、苦手意識を持つ学生に対して、新しいことに挑戦しようという気持ちを持たせることは難しい。自分で目標を設定させると、その時点で自分が確実にできる範囲での目標設定や、サンプルプログラムが容易に入手できるものになりがちである。その結果、できあがるものは、いわゆる「ありきたりなもの」や「どこかで見たことのあるもの」になってしまう。このような状況に陥ることをある程度回避する手段としても、このような仕掛けは有効であると考えている。

4. アンケート調査の結果

3.で述べた取り組みの評価を行うため、履修者全員(24名)に対して、アンケート調査を行った。その結果を以下に示す。

4.1 興味関心の向上

本演習によって「プログラミングに対する興味関心の変化」がどのように変化したかを「非常に薄く・弱くなった」から「非常に高く・強くなった」の5段階で調査した結果を、図1に示す。

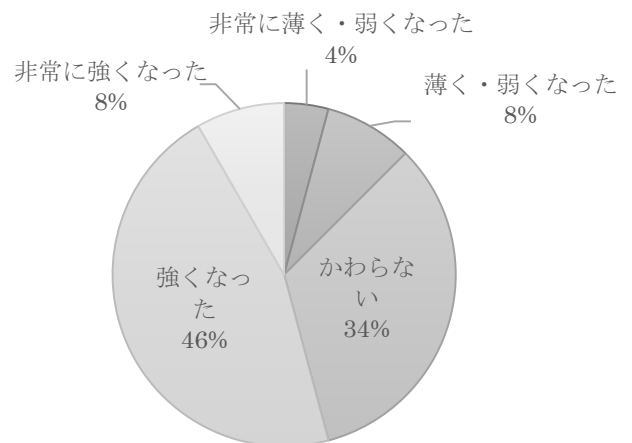


図 1 プログラミングに対する興味関心の変化

54%の学生が、本演習によってプログラミングに

対する興味・関心が強くなったと回答しており、約半数の学生の意識に良い変化が現れたことが確認できた。その一方で、関心が弱くなった学生が 12%存在している。その理由には、チームでの開発やチームのメンバー間のスキルの違いなどが考えられるが、今後調査が必要である。

さらに、本演習で使用した JavaScript などの Web 技術に関する興味関心がどのように変化したかを同様に調査した。その結果を図 2 に示す。

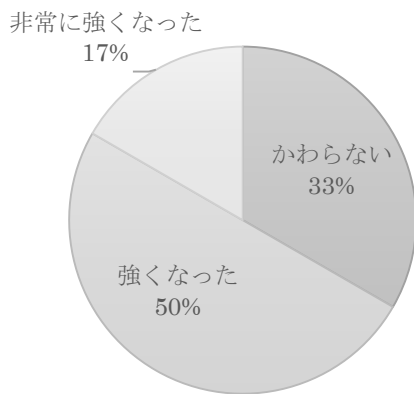


図 2 Web 技術に対する興味関心の変化

67%の学生に興味関心の向上が認められ、興味関心が低下した学生は存在していない事から、学習の動機付けとして本演習が有効に働いたものと考えられる。

4.2 成長の実感

本演習でプログラミング能力の向上を実感したか、という質問に対しては図 3 に示すとおり、41%の学生が実感したと回答している。一方で、全く向上したとは思わないという回答は 0%であったものの、向上を実感しない学生は 17%、どちらでもない学生は 42%となった。自由記述形式で調査した演習全体の感想には、「今回の授業で、計画性の大切さ技術力の不足に気づいたのでプログラミングについて精進していきたいと思いました。」や、「正直、プログラミングに対する苦手意識がかなり強いため、一人やグループで実際にプログラミングを打ってアプリを作成するのは不安でもあり自分の実力不足な点があるように思うことが多々ありました。」というような記述が複数見られたことから、成長を実感したと回答していない学生でも、自分に足りないところを自覚し、今後の成長に繋げる糸口

を見出すことが出来た者もいると考えている。

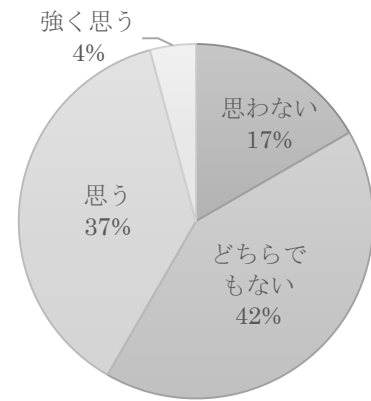


図 3 プログラミング上達に対する意欲の変化

4.3 達成感と自信

演習で達成感を感じたか、という質問には、あまり感じなかったという回答が 4%あったものの、全く感じないという回答は 0%であり、学生は程度の差はあるものの達成感を感じていることが分かった(図 4)。

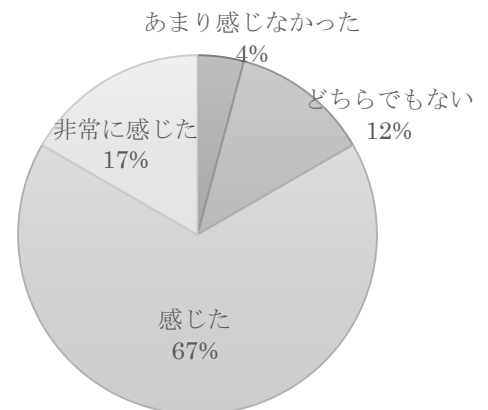


図 4 達成感

一方で、アプリケーション開発に自信がついたか、という質問に対しては、図 5 に示すとおり、非常に自信が無くなったと回答した学生は 0%であったものの、自信が無くなったと回答した学生が 21%存在した。これは、演習を通じて実際のアプリケーションの開発を具体的にイメージすることができるようになったことの現れであると考えている。その一方で、自信を持って回答した学生は 41%と、自信を無くした学生の 2 倍となっていることから、自信を無くした学生をどのようにケアしていくかの検討が必要であると感じている。また、自信を持つことができた学生に対しても、その自信をより確かなものにしていくための方策の検

討も重要と考えている。

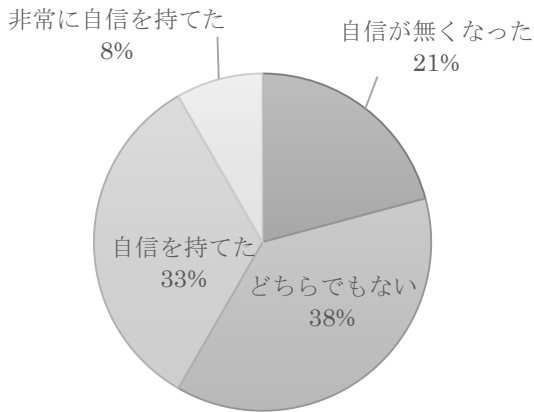


図 5 アプリケーション開発に対する自信

4.4 チーム開発

他のチームが考えたものをアプリケーションとして、チームで開発することに関しては、「今までのプログラムの授業は与えられた課題をただひたすら個人でこなすだけの授業だったがこの授業は自分で考えたものをつくったり、課題も他の生徒が考えたものをグループで開発するなど新鮮で楽しかった。」や、「自分はプログラミングが得意な方ではありませんでした。しかしこの半期でプログラムを作成することが必要となったので自分で勉強もしました。思ったよりも難しく、苦戦していましたがチームの二人が助言をくれたお陰で完成をさせることが出来たのだと思います。作成が終わった時には自然とメンバー全員でおつかれさまという言葉も飛び交い、思わず笑みがこぼれました。こんなにも達成感があったことはありませんでした。この授業を受けたことでチームの大切さが改めてわかりました。」というような意見が多く見られ、概ね良い方向に機能したと考えている。

その一方で、「グループでの作成についてはそれぞれの意見を出し合い、話し合っていくのはとてもいいと思ったが、自分たちのグループで決めたものが他のグループで作成するのは職につくためのいい経験になるが、自分たちの身の丈にあわない、自分たちで作れないので気分が乗らないことがあるので学生目線としてはあまりよくないと思った。」という意見もあった。ある程度の制約を設け、その制約の中でスマートフォンアプリケーションの仕様を考えるようにするなどの工夫が必要である。

5. おわりに

本稿では、専門科目でありながらプログラミングに苦手意識を持ってしまっている学生に対して、演習を通して自らの成長を実感させ、専門分野への興味関心の低下を防ぐ事を目的とした取り組みについて報告した。この取り組みによって、全員ではないものの学生はプログラミングや Web 技術への興味関心が向上したり、成長や達成感を感じたりするなどの効果が確認できた。

その一方で、演習内容に十分ついていくことが出来ず、以前よりも興味関心が薄れてしまったり、苦手意識をより強く持ったりする学生も、少数ではあるが確認された。自由記述の感想や普段の演習の様子からは、チーム開発に馴染めない学生や、制御構造や変数の働きなどプログラミングの基礎的な部分でのケアが必要な学生が存在していることが読み取れるので、そのような学生にどのように対応していくかの検討が急務である。

また、本稿では専門教育におけるプログラミングへの苦手意識について議論したが、1.で述べたように初等教育から高等教育における一般情報教育に至るまで、幅広くプログラミング教育が行われていくなかでも、苦手意識を持つ者が現れると考えられるので、そのような学習者への応用を検討する必要もあると考えている。

本取り組みは今年度から始めたばかりなので、改善すべき点は非常に多いと考えている。今回報告した調査結果や他分野での取り組み⁽¹⁵⁾などを幅広く参考にして、来年度の演習をより良いものにしていきたい。

謝辞

本演習の実施にあたって、アシアル株式会社の岡本雄樹氏、株式会社グッドパッチの佐野明稀氏には多大な御協力を頂いた。この場をお借りして感謝申し上げます。

本研究は JSPS 科研費 JP26350286 の助成を受けたものです。

参 考 文 献

- (1) 文部科学省：“諸外国におけるプログラミング教育に関する調査研究”，
http://jouhouka.mext.go.jp/school/pdf/programming_syogaikoku_houkokusyo.pdf (2017年1月26日確認)
- (2) 堀田達也：“初等中等教育における情報教育”，日本教育工学会論文誌 40(3), pp.131-142 (2016)
- (3) 経済産業省商務情報政策局情報処理振興課：“IT人材の最新動向と将来推計に関する調査結果”，
www.meti.go.jp/policy/it_policy/jinzai/27FY/ITjinzai_report_summary.pdf (2017年1月26日確認)
- (4) 三井一希：“小学校国語科の「書く活動」へのプログラミング導入による学習効果”，教育システム情報学会誌 Vol. 34, No. 1, pp.60-65 (2017)
- (5) 河村一樹，稲垣知宏，稲葉利江子，ほか：“これからの大学の情報教育”，日経 BP マーケティング，東京 (2016)
- (6) 岡部成玄：“高等教育における情報教育”，日本教育工学会論文誌, 40(3), pp.143-152 (2016)
- (7) 岡部成玄：“一般情報教育の全国調査(1)”，情報処理，情報処理学会誌 Vol.55, No.12 pp.1400-1403 (2014)
- (8) 岡部成玄：“一般情報教育の全国調査(2)”，情報処理，情報処理学会誌 Vol.56, No.1 pp.94-97 (2015)
- (9) 佐々木整，水野一徳，青鳶健一，作左部剛：“情報工科学科学生に対するプログラミングの意識調査”，教育システム情報学会第 35 回全国大会講演論文集, p.241-242 (2010)
- (10) 佐々木整，水野一徳，青鳶健一，作左部剛視：“Educational Game を利用したプログラミング教育の効果に関する一考察”，教育システム情報学会研究報告, Vol.24, No.5 pp.78-81 (2010)
- (11) Monaca – HTML5 ハイブリッドアプリ開発プラットフォーム：<https://ja.monaca.io> (2017年2月4日確認)
- (12) アシアル株式会社，生形可奈子，岡本雄樹：“Monaca で学ぶはじめてのプログラミング ～モバイルアプリ入門編～”，アシアル株式会社，東京 (2016)
- (13) Onsen UI 2:HTML5 モバイルアプリを早く美しく：
<https://ja.onsen.io/> (2017年2月4日確認)
- (14) Prott – Prototyping tool for Web iOS Android apps：
<https://prottapp.com/ja/> (2017年2月4日確認)
- (15) 毎日新聞：“小学校 英語授業 3 分刻みで達成感”，
<http://mainichi.jp/articles/20170204/k00/00e/040/218000c> (2017年2月4日確認)