【論文】

超スマート社会に向けたプログラミング教育の現状と課題
～大学生を対象としたプログラミング教育の実践を通して～

The Current Situation and Issues of the Programming Education towards Super Smart Society（Society 5.0）
－ Through a Practices of Programming Education for University Students －

東　　るみ子
Azuma Rumiko

（要旨）
　日本政府は，第四次産業革命や「超スマート社会」（Society5.0 ）の実現に向け，プログラミングなどの IT リテラシーを育成する教育を進めており，政府の成長戦略の中でも「義務教育段階からのプログラミング教育等の IT 教育を推進する」といった内容が盛り

込まれている。しかしながら現状では，日本は Society5.0 の実現の鍵となる人工知能（AI）や IoT 技術の基礎となる情報科学に関する教育が先進国の中でも遅れており，IT 人材の不足が課題となっている。さらに，今後日本の労働人口は減少が見込まれており，高度情報技術の需要が拡大する中，IT 人材の育成のための教育改革は必要不可欠となっている。

そこで本研究では，日本の情報教育およびプログラミング教育の現状を調査する。さらに，若者の IT スキルの向上を阻んでいる要因を分析するために，プログラミングに関するアンケート調査を実施し，課題を明らかにする。そして，情報科学を専門としない学生にも効果的なプログラミング教育を提案，実践することで，これからの時代に求められるスキルの変化と教育・人材育成の在り方について考察を行う。

1．はじめに

今後わが国では，第四次産業革命や「超スマート社会」（Society5.0）の実現に向け，イノベーションを創出するには多面的アプローチによる人材の育成・確保が必要となってくると言われている。必要となる人材の一つに，先端 IT 人材，いわゆる人工知能（AI：Artificial Intelligence）や IoT（Internet of Things），ビッグデータ等に携わる人材が挙げられる。

政府はそのための施策として，人材の創出において，「IT や外部人材を積極的に活用しながら，個々の理解度に応じた教育（アダプティブ・ラーニング），課題解決能力の育成に向け主体的・協働的に取組む教育（アクティブ・ラーニング），プログラミングなどの IT リテラシー（情報活用能力）を育成する教育を初等中等教育段階から実施する」と今後の対応の方向性を打ち出している。

しかし一方で，わが国の IT 人材の不足が懸念されており，経済産業省による「IT 人材の最新動向と将来推計に関する調査結果」の 2016 年の報告では，2020 年には 36.9 万人，2030 年には 78.9 万人の IT 人材が不足することが予測されている。IT 市場規模は今後拡大し続けていくにも関わらず，高齢化の進む日本では人材不足が深刻な問題となっている。

このような IT 人材問題を解決するために，文部科学省は 2020 年度からの初等教育におけるプログラミング教育の必修化に向けた方針を 2016 年 4 月に発表した。必修化の目的には，従来のプログラミング教育にみられるプログラムコードを書くコーディングの習得が目的ではなく，時代を超えて普遍的に求められる「プログラミング的思考」などを育み，「世界で通用する ICT 人材」を育成するという 2 つのねらいがある。情報教育の中でもプログラミング教育は，子供たちの論理的思考力や課題解決力を育てるとともに，ICT（Information and Communication Technology）に関する基礎的な知識・スキルを身につけさせるものとして，極めて重要であると考えられている。ここで述べられている「プログラミング的思考」の育成とは，1）コンピュータに処理させたい動きを決める，2）その動きを実現するために問題を手順に分解して考える，3）実際にその手順に従いプログラムコードを記述する，4）コンピュータで動作させる，5）不具合があれば原因を考え修正する，といったプログラミングにおけるプロセスを学習することで，情報を整理し，目的を達成するための手順を考えることができ，その結果物事を論理的に考える能力を養うことに繋がるというものである。山本ら（2010）や大場ら（2015）の研究においても，プロ

グラミング能力と論理的文章作成能力に相関があることは示されている。すなわちこれからの「超スマート社会」に向けて，IT 人材の育成とともに，どのような職業にも普遍的に求められるであろう「プログラミング的思考」を育むプログラミング教育の実施が求められている。

そこで本稿では，日本の ICT 教育，プログラミング教育の現状を調査するとともに，「超スマート社会」時代に向けたプログラミング教育の課題について，大学生へのアンケート調査およびプログラミング授業の実践をとおして考察を行う。

2．情報教育の現状と課題

2-1．日本の教育現場におけるプログラミング教育の現状

わが国では，学校教育における情報化への対応の必要性が 1985 年に臨時教育審議答申ではじめて示され，それ以降，情報教育および教育におけるコンピュータの活用についてさまざまな議論と提言がなされてきた（小泉，2009）。そして 1993 年に中学校の技術家庭科で「情報基礎」領域が設けられ，「コンピュータの基本操作と簡単なプログラム作成」の項目で初めてプログラミング教育が実施された（表 1）。当時は，コンピュータを扱うにはプログラミングの知識が必須であったため，情報教育にはプログラミング教育が必要不可欠であった。しかし 1990 年代前半は，世帯のパソコン普及率が 10% 台と現在よりも低く（図 1），一部の研究者や専門家，マニアに限られた普及であった。そのためパソコンを活用できる教員も少なく，実際に中学校でプログラミングに関する授業が行われるケースは極めて少なかった。

その後 Windows95 の発売をきっかけに，1990 年代後半から 2000 年代前半にかけて，パソコンの普及率とインターネットの利用率が急速に増加したことに伴い，学校教育においても情報教育への対応も必要となり，2003 年に高等学校で教科「情報」が必修教科として新設された。パソコンの普及とともに GUI ベースの OS が主流となり，ワープロソフトや表計算ソフトなどのアプリケーションソフトが普及し始めると，プログラミング教育よりもアプリケーションソフトの習得を目的とする授業が展開されるようになってきた。新設された当初，普通科の「情報」教科は「情報 A」「情報 B」「情報 C」の 3 科目で構成されており，学習指導要領（文部科学省，2000）をみると「アルゴリズム」や「プログラミング」について触れているのは「情報 B」のみであった。また 3 科目の中から 1 科目の選択必修であったため，「情報 B」の実施率は 2003 年で 7.6%，2012 年になっても 10.4% と低く，アプリケーションソフトの基本操作を教える「情報 A」の実施率が 83.9%（2003 年），72.4%（2012 年）と最も高くなっていた（佐藤，2012）。この背景には，情報科教員の配置状況や教員のスキルの問題があると考えられている。小泉・佐藤（2009）の調査によると，1938 校のうち 78.3% の高等学校が兼担で情報科を実施していることが報告されている。すなわち，情報科の専門知識を持たない教員が兼担で教えていることが多いため，結果として簡単な「情報 A」のみを実施する学校が多くなっていた。

2012 年には，中学校の技術家庭科で従来選択科目であった「プログラムによる計測・制御」が必修になった。しかし，未だ授業環境や授業時数の不足などから「プログラミング」の部分をほとんど行えていない学校も多く存在している。

そして 2010 年代にはいり，ビッグデータや IoT，人工知能，ロボット技術を活用する

第 4 次産業革命を見据えた人材育成が必要不可欠であると考えられるようになってきたため，小・中・高校でプログラミング教育の必修化など，IT 人材の育成を徹底するために学習指導要領の見直しを行うことが日本再興戦略 2016 において発表された。小学校では初のプログラミング教育となり，国語，算数，理科，総合的な学習などの各教科等の中で，その特質に応じて行っていく位置づけとなっている。

また中学校の技術家庭科では，2021 年から新たに「情報の技術」が組み込まれ，

・情報通信ネットワークの構成と，情報を利用するための基本的な仕組みを理解し，安全・適切なプログラムの制作，動作の確認及びデバッグ等が出来ること。

・生活や社会における問題を，計測・制御のプログラミングによって解決する活動を通して身につける。

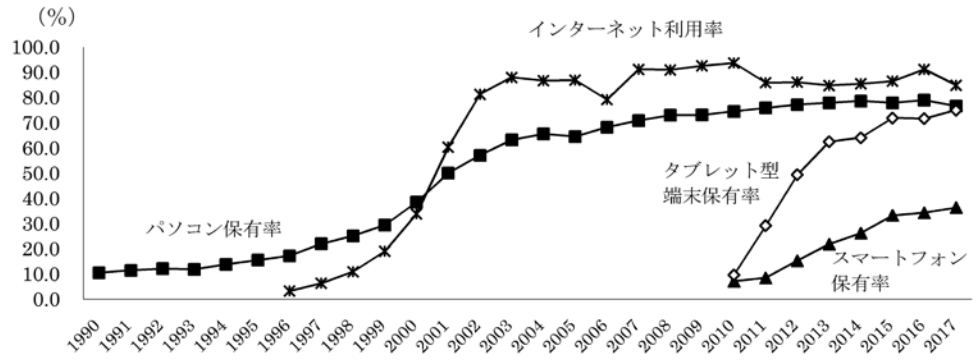
ことなどが学習目標として掲げられている。

さらに高等学校では，2022 年度から共通必履修科目「情報Ⅰ」が新設され，文系・理系問わず全ての生徒がプログラミングやデータベースの基礎，ネットワークについて学習する（堀田，2018）。初等教育ではプログラミングの体験を重視していたのに対し，中等教育ではより実践的な学習を目指した改訂になっている。

表 1　情報科目に関する必修化の流れ

1993 年	中学校の技術家庭科で情報基礎領域が新設
2003 年	高等学校で教科「情報」が必修化
2012 年	中学校の技術家庭科で「プログラムによる計測・制御」が必修化
2020 年	小学校でプログラミング教育が必修化
2021 年	中学校の技術家庭科で「情報の技術」が組み込まれ，プログラミングの内容が倍増
2022 年	高等学校で共通必履修科目「情報Ⅰ」が新設
2024 年	「情報Ⅰ」が大学の入試科目に追加される予定

出所）シリフグリ（2005），堀田（2018）



注）インターネット利用率は単身世帯を含む全世帯に占めるインターネットを利用した世帯員がいる世帯の比率であり，パソコンや携帯端末などインターネットの利用機種や利用場所は問わない。

出所）内閣府“消費者動向調査”https://www.esri.cao.go.jp/jp/stat/shouhi/shouhi.html
総務省“情報通信統計データベース”http://www.soumu.go.jp/johotsusintokei/field/tsuushin01.html
総務省“通信利用動向調査”http://www.soumu.go.jp/johotsusintokei/statistics/statistics05.html

図 1　情報機器の世帯保有率とインターネット世帯利用率の推移

2-2. IT 人材育成の課題

前節で述べたように，わが国では 2020 年から小学校でのプログラミング教育が導入される中，海外では表 2 に示されるように早い時期から検討，実施されている。また近年ではアジア諸国においても積極的に情報教育が実施されており，韓国やインドでは 2000 年代から初等教育における ICT リテラシーの必修化やプログラミング教育の導入が始まっている。

わが国の情報教育の遅れの要因には，教員側のスキル不足の他に学校における ICT 環境の整備の遅れも挙げられる。PISA（2015）によると，日本の学校での ICT 機器の活用の遅れは「底辺国」レベルであることが示されている。文部科学省の調査によると，2017 年の時点で教育用コンピュータは 1 学級 5.9 人に 1 台の割合である。

このような情報教育の遅れのため，わが国の若年層の情報通信技術の利用水準は，世界的に見ても明らかに低い部類に属すると言われている（斎藤，2008）。OECD（2015）の調査では，15 歳生徒のうち「表計算ソフトでグラフを作れる」，「パワーポイント等でプレゼン資料を作れる」割合が日本はどちらも約 30% と調査 45 か国のうちで最も低いレベルであることが報告されている。この結果には過大評価をしていると疑問を呈す声もあるが，いずれにせよ日本の若者の ICT スキルが高いレベルにないことに変わりはない。さらにアメリカやイギリスなど 7 か国の若者が情報機器をどのくらい所持しているかという 2013 年の内閣府（2013）の調査によれば，日本の若者は携帯ゲーム機器（53.9%）のみ高い所持率を示し，タブレット型端末（13.0%）やノートパソコン（43.3%），デスクトップパソコン（18.5%）といった情報機器の所持率に関しては，7 か国で最低の水準であった。

以上のように，日本の若者の IT スキルを他国と比較すると，パソコンの所持率は低く，基本的なスキルが身に着いていないことがわかる。その原因の一つに，教育の ICT 化の遅れが起因していることが挙げられ，これからの IT 人材の育成にはこれらの課題を解決していくことが必要不可欠である。そのためにも，早い段階からのプログラミング教育は重要であり，総務省でもその実現に向け，プログラミング教育の低コストかつ効果的な実施手法や指導者の育成方法等を，クラウドを活用しつつ実証し，2016 年度より「若年層に対するプログラミング教育の普及推進」事業（総務省，2017）として，全国的に普及させる取り組みを行っている。

表 2　海外におけるプログラミング教育の学校カリキュラムへの導入例

国名	取組概要
イギリス	2014 年 9 月のカリキュラム改訂で 5 歳～ 16 歳でのプログラミング教育を必修化
イスラエル	2000 年に高校におけるプログラミング教育を必修化、現在中学への導入も計画中
エストニア	2012 年に小学校から高校まで計 20 校のパイロット校でプログラミング教育を開始
オーストラリア	連邦政府の新たなカリキュラム案は 8 歳～ 13 歳のプログラミング教育を必修化する内容（現在最終承認待ち、2016 年頃から各州で実施の見込み）
韓国	2015 年から全中学校に正課外のプログラミング教育を実施 2018 年にはプログラミング教育を含む「ソフトウェア」学習を正式科目に採用予定
ニュージーランド	2011 年に高校生がプログラミング等のコンピュータサイエンスを学ぶ新カリキュラムを導入
フィンランド	2016 年のカリキュラム改訂で 7 歳～ 16 歳でのプログラミング教育を必修化

出所）総務省“教育・学習分野の情報化に係る国内外の動向と先進事例”第 3 回 ICT ドリームスクール懇談会

2-3. プログラミング教育の在り方

人工知能を活用するための技術力にはプログラミングスキルも必要なスキルの一つであるが，情報通信白書（2016）によると，日本の就労者はアメリカよりも各種人工知能活用スキルの習得意欲が低く，「対応・準備については，特に何も行わない」とする人が過半数を超えていることが報告されており，今後人工知能が普及浸透していく中で，人工知能を活用する流れから取り残される人が出てくることが懸念されると指摘されている。

このような懸念を払拭するために，ここ数年プログラミング教育に関する実践事例やワークショップ，民間の教室も増えている。実践研究においては，論文検索サイトのCiNii でキーワード「プログラミング教育」の検索を行ったところ，2014 年から急激に掲載論件数が増えており，2017 年は 2014 年の 4 倍にも増えていた。

大学教育においても，文系，理系を問わずプログラミング関連の講義は多く開講されており，東京大学では，リベラル・アーツ教育の一環として 2006 年から 1,2 年生向けにプログラミング教育を導入している（森畑，2016）。

しかし従来のプログラミング教育では，英語をベースとしたプログラミング言語に従い，テキスト（コード）を記述し（コーディング），プログラムを作成することを目的としているため，スペルミスでプログラムが動かなかったり，英語自体に苦手意識をもつ学生にとって習得が難しかったりと，英語を母国語としない日本の学生はプログラミングの仕組みやロジックを理解する前に，プログラミング学習に挫折してしまうケースが多くみられている。大岩(2014)も，現行のプログラミング教育はほとんどが言語教育で終わっていて，情報化に対応できる教育になっていないことを指摘している。そのため，文系学部の学生に対しては，プログラミング教育よりも就職後に使用するアプリケーションソフト（ワープロ，表計算，プレゼンテーション）のスキルだけを身に着けさせれば十分との主張もみられる。

一方，原田（1992）は，システムを自分の問題解決のために主体的に利用する姿勢とシステムを道具として使いこなす能力をもつ「健全なユーザ」育成の考えのもと，文系学部生に対してもプログラミング教育を行う意義を見直している。また東本（2017）は，プログラミング教育の意義として，「これからの教育は課題に沿って解かせて，答え合わせを行うのではなく，学習者自身が試行錯誤し，それに適切に答える知的学習環境の開発が今後の情報化に向けて急務である」ことを述べている。

さらにプログラミング学習の難しさを解決するために，小中学生向けのプログラミング教育では，ビジュアルプログラミングツールを利用し，工夫した実践報告も数多くみられる。ビジュアルプログラミングとは，視覚的なパーツを動かし，その組合せでプログラミングを実現することができるものである。視覚的であるため分かりやすく，従来のプログラミング言語にみられるような英語表現の文法を覚える必要もないため，低年齢からでも初歩的なプログラミングを行うことができる。そのため，大学のプログラミング教育でビジュアルプログラミングツールを利用した実践研究も報告されている（森，2010）。

また吉田ら（2015）の研究では，文系学生を対象に Raspberry Pi[1)] を用いながら学習することで，構築主義を背景としたフィジカル・コンピューティングを小，中，高校生に教えるための情報教育の知識や技術を身に着けることができ，プログラミング教育に対する意識の変容を促すのに効果的であったことが報告されている。フィジカル・コンピューティング（Physical Computing）とは，さまざまなセンサー技術を用いてコンピュータと

実世界との物理的なやりとりを実現するシステムや手法である。

これら先行研究にみられるように，プログラミング教育の導入には，学生のプログラミングへの苦手意識を克服する授業設計を検討していくことが課題である。

3．大学生のプログラミングに関する意識調査

3-1. アンケート調査概要

大学生がプログラミングに対してどのような意識をもっているのか，学生のプログラミング経験とプログラミングに関するイメージを調査するために，首都圏の文系学部に所属する大学生を対象にアンケート調査を行った。

⑴ アンケートの調査方法

情報関連科目を受講する大学生を対象に，初回講義時に，Web アンケートフォームを用いて，パソコンまたは携帯情報端末から答えてもらう形式で実施した。

⑵ アンケート実施期間

2017 年 9 月 18 日～ 10 月 3 日。

⑶ アンケート総数

213 名（有効アンケート数 213 件）うち男性 134 名，女性 79 名。

調査を実施した情報関連科目は，文系学部で開講されている必修または選択科目であり，主に 1 年生あるいは 2 年生が履修している講義であった。そのため，学生のプログラミングに対する興味の有無とは関係がない状況下でアンケートを実施した。アンケート項目は表 3 のとおりである。

表３　アンケート調査項目

1	高校の時の専攻
2	現在のプログラミングスキル
3	小学生の時のプログラミング経験
4	中学生の時のプログラミング経験
5	高校生の時のプログラミング経験
6	経験したことのあるプログラミング言語（自由記述）
7	プログラミングに関して抱いているイメージ（自由記述）
8	なぜ初等教育にプログラミング授業が導入されると思いますか（自由記述）

3-2. アンケートの集計結果

「高校の時の専攻」および「現在のプログラミングスキル」と「小・中・高校でのプログラミング経験」を尋ねた質問に対する回答の集計結果を図 2, 3 と表 4 に示す。アンケート対象者の 90% が普通科を専攻しており，そのうちの 54%（104 名）がプログラミングの経験がない学生であった。普通科でプログラミングの経験がある 46%（86 名）のうち，小学校の授業で経験した学生が 8%（7 名），中学校の授業で経験した学生が 24%（21 名），高校の授業で経験した学生が 41%（36 名）であった。また商業科，情報科出身の学生は全体の 5%（10 名）おり，そのうちの 60%（6 名）がプログラミングの経験があり，高校の授業で学んでいた。

「現在のプログラミングスキル」だけをみると，「かなりできる」と回答した学生は 0 であった。最も多かったのは，「経験ない（114 名）」の 53.3% で，次いで「経験はあるがほとんどわからない（79 名）」の 37.3%，「少しできる（19 名）」9.0% であった。この結果から，2003 年に教科「情報」が高校で必修化されてから 15 年が経過しているにもかかわらず，未だプログラミング教育を実施している学校が少ないことが推測できる。

次にプログラミング経験者に対して，経験したことのあるプログラミング言語（またはツール）を尋ねた結果を表 5 に示す。ビジュアルプログラミングを経験したことのある学生は 4 名（Scratch）と少なく，ほとんどの学生はテキストベースのプログラミング言語からはじめていることがうかがえる。

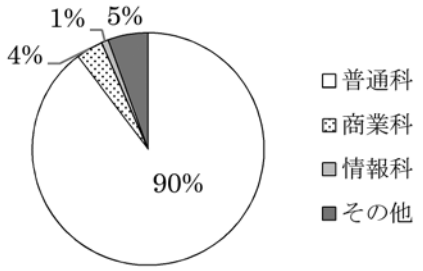


図２　高校の時の専攻の内訳（n=213）

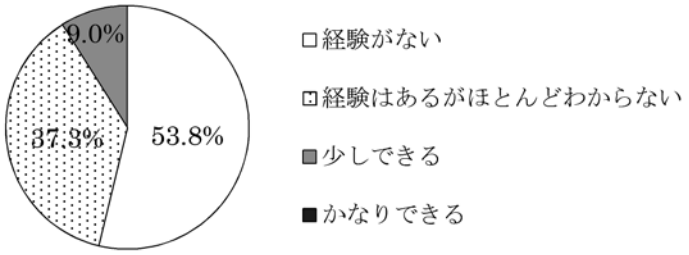


図３「現在のプログラミングスキル」の回答結果（n=213）

表４　小・中・高校生の時のプログラミング経験に関する回答数（人）

プログラミング経験	独学（インターネットや本）で勉強	学校で経験	ワークショップやイベントなどで経験	スクールに通った	なし	その他
小学生の時	2	8	0	0	201	2
中学生の時	3	27	0	1	181	1
高校生の時	5	53	1	0	154	0

表５　経験したことのあるプログラミング言語（複数回答）

言語（またはツール）	人数
Java	22
Visual Basic	11
Python	6
C	5
JavaScript	4
PHP	4
Scratch	4
COBOL	2
C#	2
その他	7

3-3. 自由記述の定量分析

3-3-1.「プログラミングに関して抱いているイメージ」の頻出語

次に，アンケート項目７の「プログラミングに関して抱いているイメージ」に対する自由記述の回答に対して，定量的に分析するためにテキストマイニング（樋口，2004）を施した。その際，頻度は多いがそれ自体意味を持たない一般的な語は分析からはずした（例：ある，いる，なる）。分析の対象となる文書数は 213，語の数は 883，出現回数の平均は 3.39，標準偏差は 10.29 であった。

表６に形態素解析し抽出した頻出単語上位 10 件の結果を示す。最も出現回数が多かった語は「難しい」であり，半数の学生がプログラミングのイメージとして挙げていた。その他に「複雑」「大変」などの否定的な語がみられた。この結果から，多くの学生がプログラミングを体験する前から難しいといった苦手意識を抱いていることがわかる。一方で，「楽しい」「面白い」「かっこいい」といった肯定的なイメージの語も１割強みられた。

3-3-2. プログラミングスキルとイメージとの関連

次に，学生のプログラミングのスキルとプログラミングに対するイメージとの関連の強さを測定した結果を表７に示す。ここでは，学生のプログラミングのスキルとプログラミングに対するイメージの自由記述文章から抽出された全ての語の関連を，Jaccard 係数によって測定している。ここでは便宜上，プログラミングのスキルにおいて「経験がない」グループを A，「経験はあるがほとんどわからない」グループを B，「少しできる」グループを C とよぶことにする。

表７をみると，C グループでは，「楽しい」という語との関連が最も強かった。B グループにも，「興味」「面白い」といった肯定的な語との関連が見られた。すなわち，少しでもプログラミングの経験がある学生にとっては，プログラミングに対して「楽しい」や「興味」「面白い」といった肯定的なイメージが存在しているといえる。「興味」に関しては，B グループでは，「興味がある」という語と，「興味はあるが難しい」といった逆接のかたちで使われているケースがみられ，肯定的なイメージと否定的なイメージが混在していることがうかがえた。一方 C グループでは，「興味がある」という肯定的なイメージだけで使われており，逆接を伴った表現はなかった。

表６　プログラミングに関して抱いているイメージの頻出単語上位 10 件

抽出語	出現回数	割合（%）
難しい	108	50.7
興味	32	15.0
楽しい	26	12.2
複雑	26	12.2
面白い	12	5.6
論理	7	3.3
かっこいい	7	3.3
理系	4	1.9
大変	4	1.9
便利	4	1.9

表７　プログラミングのスキルとプログラミングに対するイメージとの関連

A: 経験がない		B: 経験はあるがほとんどわからない		C: 少しできる	
難しい	0.408	難しい	0.285	楽しい	0.250
イメージ	0.044	複雑	0.132	頭	0.100
理系	0.026	興味	0.124	達成	0.100
必要	0.018	論理	0.077	興味	0.063
少し	0.018	面白い	0.060	新しい	0.053
学ぶ	0.018	大変	0.039	作れる	0.053
堅い	0.018	数学	0.026	式	0.053
使う	0.018	地道	0.026	混乱	0.053
便利	0.018	ミス	0.026	綺麗	0.053
必須	0.009	難解	0.026	纏まる	0.053

注）数値は Jaccard の類似性測度

「難しい」との関連では，A グループで最も強い関連が見られ，B グループでは弱い関連，C グループにおいてはまったく関連が見られなかった。この結果から，学生はプログラミングから「数学」などを連想させることにより「難しい」「複雑」というイメージを抱き，プログラミングを体験し理解することで「興味」がうまれ，肯定的なイメージをもつようになっていくことが考えられる。

3-3-3. クラスター分析

アンケート項目 8 では，学生がプログラミング教育を学ぶ理由を理解しているのかを確かめるために，「なぜ初等教育にプログラミング授業が導入されると思いますか」の質問を行った。これらの回答についても自由記述のため，テキストマイニングにより定量的な分析を行った。分析の対象となった文書数は 210（3 名は無回答），抽出語の数は 3134，出現回数の平均は 3.89，標準偏差は 8.89 であった。この項目の回答に対しては，クラスター分析を行った（樋口，2014）。クラスター分析では，似通った文脈で使われている語のグループを見ることができるため，どのような意見が多かったのかを定量的に調べることができる。

表８「なぜ初等教育にプログラミング授業が導入されると思いますか」の
回答中に頻出していた語のクラスター分析

クラスター１（14）		クラスター２（17）		クラスター３（13）	
必要	0.197	将来	0.515	情報	0.458
今後	0.160	必要	0.173	進む	0.294
作り出す	0.071	スキル	0.059	コンピュータ	0.125
学び	0.071	文系	0.059	社会	0.115
必ず	0.071	問う	0.059	使う	0.095
日本人	0.067	理系	0.059	IT	0.087
先	0.067	予測	0.056	思う	0.079
創造	0.067	絶対	0.056	欠如	0.077
技術	0.063	役に立つ	0.056	挙げる	0.077
企業	0.063	役立つ	0.050	進展	0.077
クラスター４（30）		クラスター５（12）		クラスター６（22）	
社会	0.518	思考	0.800	プログラミング	0.420
出る	0.226	論理	0.647	知識	0.360
必要	0.202	養う	0.286	コンピュータ	0.125
ネット	0.129	身	0.167	現在	0.103
人	0.083	解決	0.167	必要	0.094
思う	0.074	能力	0.111	流れ	0.091
働く	0.067	育む	0.083	時代	0.088
スキル	0.065	設ける	0.083	利用	0.087
力	0.065	PC	0.083	多く	0.087
役立つ	0.063	触れ合う	0.083	社会	0.085
クラスター７（6）		クラスター８（25）		クラスター９（64）	
時代	0.400	技術	0.500	将来	0.155
最先端	0.286	発展	0.227	スキル	0.153
突入	0.167	AI	0.227	パソコン	0.125
利用	0.125	IT	0.214	思う	0.122
求める	0.091	先進	0.174	仕事	0.108
		導入	0.130	使う	0.105
		プログラミング	0.127	考える	0.099
		発達	0.120	今後	0.097
		日本	0.115	現在	0.088
		考える	0.091	当たり前	0.078

注）（ ）内は文書数，数値は Jaccard の類似性測度を表す

語の出現回数の平均から３回以上文章中に出現していた頻出語 82 種類をクラスター分析の対象とした。クラスター間の距離の計算方法には Ward 法を採用し，回答を９つのクラスターに分類した。その分析結果を表８に示す。どのクラスターにも属さない回答は７つであった。また各クラスターに分類された回答の一部を表９にまとめた。

まずクラスター１と２については，「必要」という語が共通して見られ，内容には必要である具体的な理由の記述はなく，単に「今後（将来）必要だから」と述べている回答の

表９　各クラスターに分類された回答例

I	クラスター１	「今後必要とされる技術だから」「企業で必要とされているから」
	クラスター２	「将来に役に立つから」「将来、文系理系問わず必要なスキルだから」
II	クラスター３	「情報化が進んでいるから」「IT 化が進んでいるから」
	クラスター８	「AI や IoT の発展に伴い企業が導入し始めたから」「AI 時代の到来のため」
III	クラスター４	「ネット社会だから」「社会において必要だから」
IV	クラスター５	「論理的思考力と問題解決能力を養うため」
V	クラスター６	「これからの時代、限定された人たちがプログラミングをするものでなく、だれでもある程度は知識がある状態になるべき」 「国際的にプログラミングが必要とされてきているから」
	クラスター７	「これからの時代に求められているから」「最先端の時代だから」
VI	クラスター９	「将来の進路の幅を広げるため」「パソコンを使った仕事が当たり前になってきていて～」「コンピュータを使う仕事が増えてきているため」「機械を操作することができないと遅れをとってしまうからプログラミングのスキルが必要不可欠になってきている」

注）ローマ数字は，大カテゴリーの分類を表している

集まりであった。

次にクラスター３と８に関しては，「AI」や「IT」「IoT」などの情報技術を表す用語が特徴的に見られ，回答では主に技術の進展に伴いプログラミング教育が必要であることが理由として述べられていた。

クラスター４では，ネット社会や社会の変化に対応するためといった理由が述べられていた。

次にクラスター５では，文部科学省もプログラミング教育の目的として掲げている「論理的思考力」と「問題解決能力」の育成を理由として挙げていたグループであった。

クラスター６，７では，「時代」という語がキーワードとなっていた。クラスター６では「“時代”＋具体的な理由」が記述されていたのに対し，クラスター７のグループでは「時代」自体を理由としている記述が集まっていた。

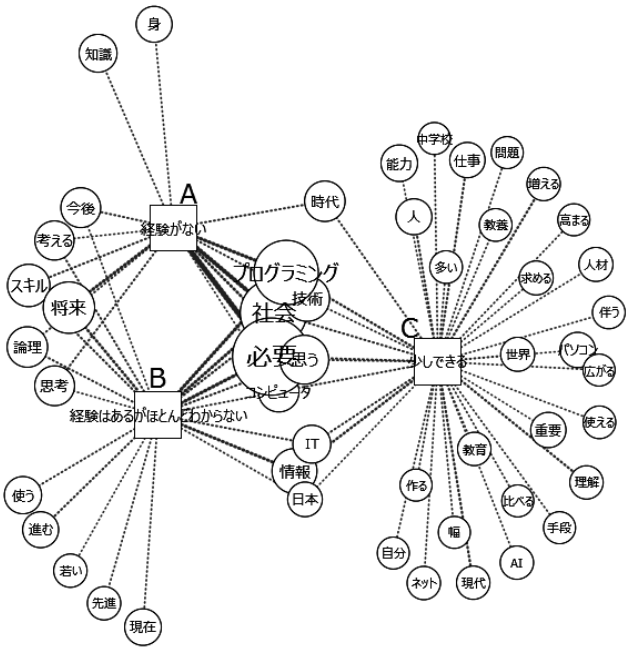
クラスター９には，仕事の業務内容が変わっていき、その上でパソコンや機械の操作が必要だからといった「将来の仕事のために必要」という理由を挙げる回答が集まった。またこのクラスターに分類された回答は最も多く 64 件であった。この結果から，多くの学生はコンピュータやプログラミングのスキルが，将来の仕事に直結するものだと感じていることがわかった。

この結果から，９つのクラスターを６つの大カテゴリー（I ～ VI）に分類した。カテゴリーI は「将来必要だと漠然と感じているグループ」，II は「技術の発展に伴い必要だと感じているグループ」，III は「ネット社会において必要だと感じているグループ」，IV は「論理的思考力や問題解決能力を育成するために必要だと感じているグループ」，V は「プログラミングを必要とする時代になるから必要だと感じているグループ」，VI は「仕事の業務においてプログラミングのスキルが必要不可欠だと感じているグループ」と解釈することができる。

3-3-4. 共起ネットワーク

さらに，学生のプログラミングスキルとこれらの回答に現れた語との関連を調べるために，共起ネットワークを作成した。共起ネットワーク図を作成することで，学生のプログラミングスキルとプログラミング授業が導入される理由に対する回答の頻出語とが互いにどのように結びついているのかを視覚的に確認することができる。なお，分析にあたっては，語の最小出現数を 3 に設定し，描画する共起関係は 70 に設定した。共起ネットワークを図 4 に示す。

図 4 をみると，「プログラミング」「コンピュータ」「技術」「社会」「必要」の語は，どのプログラミングスキルのグループにも共通して共起していることがわかる。特に A グループに関しては「将来」の語も含め，太い線で結ばれていることから強い共起関係があるといえる。このことからプログラミングの経験がない学生は，クラスター分析でカテゴリー I に分類した「将来必要だと漠然と感じている」や，カテゴリー III の「ネット社会において必要」を理由として挙げていることがわかる。B グループでは，「先進」「IT」「情報」「論理」「思考」の共起関係がみられた。これらは，カテゴリー II の「技術の発展に伴い必要」と IV の「論理的思考力や問題解決能力を育成するために必要」の回答に多く現れる語である。しかしながら，どの語も A グループあるいは C グループでも共通して共起しているものであり，B グループ独自で強く共起している語がみられなかったため，B グループはプログラミング経験なし（A）と少しできる（C）グループの両方の意見を持っていると考えることができる。C グループにおいては，どのグループよりも多くの語が共起していることから，プログラミング教育の必要性に関して様々な語を利用して意見を述べてい



注）図中の線の太さは共起関係の強さを，円の大きさは出現数の多さを表す。

図 4　学生のプログラミングスキルと「プログラミング授業が導入される理由」の回答における頻出語に対する共起ネットワーク

ることがわかる。すなわちプログラミングスキルが高くなるにつれて，プログラミングを学ぶ意義について具体的に理解できているのではないかと考える。

3-4. 考察

アンケート調査の結果から，高校に「情報」教科が必修で設置されているにも関わらず，多くの学生は大学入学前までにプログラミングの経験がないことがわかった。この原因の一つには，高校の「情報」を免許外教科担当者いわゆる専門性の低い教員が教えているケースが多いため，パソコンの基本操作や文書作成までしか実施されていないことが考えられる。また中西の実践事例の報告（中西，2016）でも，「教員自身の経験不足」や「生徒のエラーや不具合の個別対応に追われる」理由で情報科の教員の中にはプログラミング教育を敬遠している者がいることが述べられている。

プログラミングのスキルとプログラミングに対するイメージの関係では，スキルに応じて感じているイメージが異なり，経験したことがない学生ほど「難しい」というイメージが先行しているため，その印象を変えることがコンピュータやプログラミングへの興味を持たせるきっかけの一つになると考える。また，今回文系学部の大学生へのアンケート調査だったため，「プログラミング＝数学（理系）」という印象を抱いている学生もある一定数存在した。これは，以前の学習指導要領の「数学 B」でプログラミングが扱われていたことや，「論理的思考」という単語からそれらを関連付けて連想している可能性がある。そうなると，数学に苦手意識をもつ学生ほどプログラミングは「難しい」という固定観念を持ち，学習を避けることに繋がりかねない。そのため，次期学習指導要領で示されている「社会」や「理科」の授業の中で学習ツールの一つとしてプログラミングを導入することは，効果的なプログラミング教育だと考える。

また「初等教育でプログラミング授業を導入する」意義についても，プログラミング経験のある学生ほど理解している傾向にあった。このことからも，初等中等教育の時点でプログラミングを体験させることは，日本の若者の情報スキルを他の先進国並みに引き上げるためのきっかけとなる可能性がある。

4 . プログラミング授業の実践

4-1. 概要と目的

アンケート調査結果に表れていたように，学生はプログラミング対し「難しい」という印象を抱いているため，特に情報技術に対して消極的な学生には，教科書に沿って仕組みや用語を学ぶスタイルの講義ではなく，実際に手を動かし体感することで理解を深めることが必要だと考えられている。先行研究にもみられたように，プログラミング初学者にとってビジュアルプログラミングおよびフィジカル・コンピューティングの学習は視覚的に理解できることから効果的であると考え，本研究では文系学部の大学生に向けたプログラミング授業をデザインし，実践を通じてその学習効果を評価することを試みた。文系学部の学生に対して，従来のプログラミング言語の学習ではなく，ビジュアルプログラミングツールを使いプログラムの基本的な仕組みを視覚的に体感させ，センサーなどと組み合わせることで日常のあらゆるところでコンピュータプログラムが活用されていることを理解させることで，プログラミング学習でみられる「難しい」というイメージを払拭し，プログラ

ミングおよび情報技術への興味関心と理解を深めることができるのではないかと考えた。

本実践では，教職課程の高校（情報）の免許取得を目指す大学生にビジュアルプログラミングおよびフィジカル・コンピューティングを組み込んだ授業を体験してもらった。従来のプログラミングの講義との違いおよび理解度合いなどを考察することにより，学生に受講生の立場と教える立場の双方の観点から効果的なプログラミング学習について考えてもらうことを目的とした。

4-2. 方法

⑴ 対象

本学商学部の教職課程（高校「情報」）を履修している学生 8 名（男性 5 名，女性 3 名）を対象とした。8 名はいずれも，専門科目「コンピュータ・システム 2」で Visual Basic または Java といったプログラミング言語の学習を経験したことがある学生であった。

⑵ 実施期間

2016 ～ 2017 年の前期に開講している教職課程科目「コンピュータサイエンス」の講義内の 3 週（週 1 コマ 90 分× 3 回）を利用して実施した。

⑶ 実習内容

全 3 回の実習内容は表 10 のとおりである。

表 10　全 3 回の実習内容

回	内　　容
第 1 週	ビジュアルプログラミング Scratch を用いてプログラミングの基礎（順次構造，分岐構造，反復構造）を学習
第 2 週	課題制作 Scratch を使った簡単なゲームの制作
第 3 週	フィジカル・コンピューティング Scratch とセンサーボードを組み合わせたプログラミングを学習

4-3. 実践内容

本実践では，ビジュアルプログラミングのツールとして，米国マサチューセッツ工科大学の MIT メディアラボによって開発された教育用のプログラミング環境である Scratch（Mitchel Resnick et al. , 2009）を使用した。Scratch は図 5 に示すようなブロックの組み合わせでプログラムを組むことができるため，従来のプログラミング言語のようにコーディングを必要とせず，タイピングミスなどを気にせず感覚的にプログラミングが学べる特徴がある。また Scratch は多言語に対応しており，日本語でも環境が用意されているため，従来の英語によるプログラミングを苦手としている学生にとってもとりかかりやすい環境ツールとなっている。

実習 1 週目には，Scratch の基本操作と，プログラムの基本処理である「順次構造」「分岐構造」「反復構造」の考え方を Scratch を使って実現する方法を学習した。実習は，理論の解説，例題の実践，例題の解説，演習課題といった流れで進めていった。図 6 は 1 週目の実習の様子である。

プログラムの基本処理である三大構造を学習する中で，「分岐構造」「反復構造」はプロ

グラミング初学者にとって難しいと感じるポイントであり，ここでつまずくことでプログラミングに対して苦手意識をもつ学生が多くなる。しかしScratchでは，「分岐構造」であれば「もし〜なら」ブロックを使って直観的にプログラムを作成することができるため，プログラミング初学者の学生は問題なく処理構造を理解し，演習に取り組んでいた。しかし一方で，Scratchには「もし〜なら，ずっと」のような複数の処理が含まれたブロックが複数あるため，従来の言語の学習経験がある学生はどのブロックを使えばよいのか迷う場面が見られた。その他には，プログラミングにおける座標の概念や，キーボード入力の判定処理，乱数の概念についても説明をし，演習問題に取り組んでもらった。演習問題の内容を表11に示す。

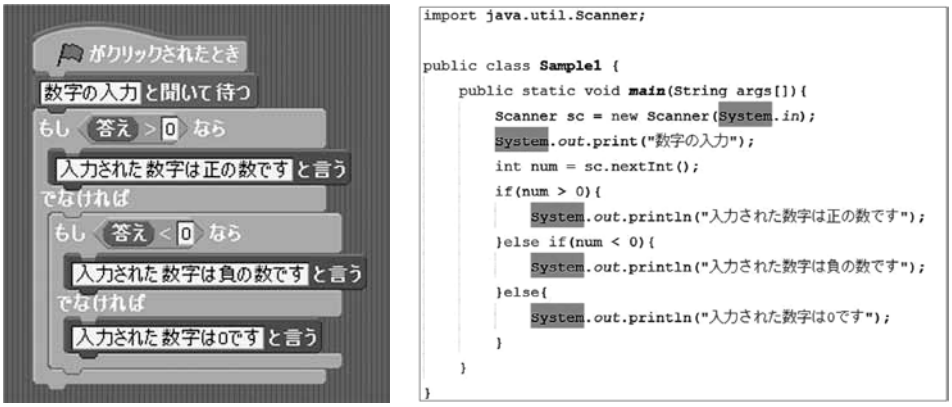


図５　ビジュアルプログラミング（左）とテキストベースのプログラミング（右）の例

図６　１週目の様子（Scratchでプログラミングを組んでいる最中のパソコン画面）

表11　１週目の演習問題の内容

	演習タイトル	学習項目
演習１	10歩動いてニャーと鳴くネコ	順次構造
演習２	左右に動くネコ	反復構造
演習３	マウスをクリックすると色を変えるネコ	分岐構造
演習４	名前を聞くネコ	キーボード入力判定
演習５	ネコをマウスでクリックした回数をカウント	変数
演習６	ネコとカニを同時に動かす	並列処理、乱数

2週目は，Scratchによるシューティングゲームをベースとした簡単なゲーム制作を行った。ゲームの基本的な仕様は教員側で提示し，それ以外のキャラクターの制作や動き，効果音，その他機能の追加などは学生個人に自由に考えてもらった。前週に学んだ分岐構造を使った処理で「弾」が「敵」にあたった時の判定や，反復構造や乱数を使った「敵」の動きの制御、「弾」があたった回数をカウントするための変数の利用などを学生自身に考えてもらうことで，学生が主体的に楽しみながら独自の作品を作り上げている様子が見られた。またScratchではマルチメディア素材を簡単に利用することができるため，「弾」があたった時に効果音をつけたり，背景画像を適用したりするなどの工夫を学生に促すことで，学生はプログラミング学習というよりモノづくりを体験しているような感覚で制作に取り組んでいた。

3週目は，プログラムの活用やプログラミングなどのソフトウェア面だけでなく，ハードウェアに関する知識も絡めてプログラミングの学習を行うために，フィジカル・コンピューティングの実習を行った。従来のコンピュータアプリケーションは，マウスやキーボードで指示を出し（入力），その結果をディスプレイに表示する（出力）ことが一般的であった。しかし，現在ではスマートフォンなどに代表されるように，さまざまなセンサーによって情報がコンピュータに入力され，画面表示だけでなく音や光といったあらゆる方法での出力が可能となっている。照明などを点灯させるための人体感知センサーや冷蔵庫などに利用されている温度センサーなどのように，センサーを使ったデータの入力方法が我々の生活の中に浸透してきている。そのためフィジカル・コンピューティングの技術を学ぶことは，学生が身の回りで使われているコンピュータプログラムの仕組みを理解することに繋がり，よりプログラミングを身近に感じてもらうことができると考えた。

実習ではフィジカル・コンピューティング用のプラットフォームとして，センサーボード（なのぼ〜どAG[2]）を使用した。フィジカル・コンピューティング用のプラットフォームとして有名なものの一つに，Arduino[3]がある。Arudinoは，回路構造やソフトウェアなど全ての情報が公開されているオープンソース・ハードウェアであるため，Arduinoをさらに使いやすくした互換製品も市場に多く流通している。なのぼ〜どAGもArduinoと互換性のあるセンサーボード1つであり，Scratchだけでなく Arduino言語（C言語に似た言語）でプログラミングをすることも可能である。

なのぼ〜どAGには，明るさ，音量，スライダー，スイッチ，傾きなどの各種センサーが搭載されており（図7），センサー値をScratchから読み込めるようになっている。また，光やモーターなどのアクチュエーターを使って周囲の環境に働きかける，いわゆる出力装置も搭載している。

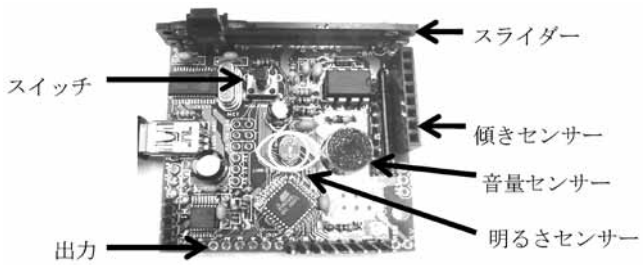


図7　センサーボード（なのぼ〜どAG）

実習では，音量センサーから受け取った値に応じてScratch上のネコのキャラクターの動きの速さを変えるプログラムや，明るさセンサーの値に応じて，ネコの色を変えるプログラムなどを作成し，センサーボードとプログラムの連携の学習を行った。その後演習課題として，ある基準値を超える大きな音をセンサーが受け取った時に，LEDが点滅するプログラムを考えてもらった。これは，聴覚障害の方向けに利用されている電話着信やインターホンの音を光で知らせる機器と同じ仕組みのプログラムである。当初学生は，キーボード以外の入力装置から情報を得て，その情報をコンピュータ上で処理をし，ディスプレイ以外の装置に結果を出力するといった流れを理解することに戸惑っていたが，例題の動きを確認しプログラムの流れを理解することで，その後の演習課題では試行錯誤しながらも自分の力でプログラムを組むことができるようになり，講義の後半では課題の内容以上の機能なども追加できるようになっていた。さらに演習課題の後，Scratchに用意されているセンサーボードを使ったさまざまなサンプルプログラムを実行･体験させることで，IoT技術の可能性についても考えてもらった。

4-4. 実習後の学生の考察

3回の実習を終えた後，学生には下記の3つの設問に答えるかたちでレポートを書いてもらった。

設問1：あなたが教職に就き，中，高生にプログラミング教育を行わなければならない場合、あなたならどのような工夫をしますか。中，高生のプログラミングに対する苦手意識が高い理由を考え，それを克服する為にはどのような取組みが必要かを考えて下さい。

設問2：本実習で使用したプログラミング環境ツールScratchに関して，どのようなところがプログラミング教育に役立ち，逆にどのようなところが足りないと思いますか。

設問3：フィジカル・コンピューティングを教えることで、プログラミング教育にどのような効果があると思いますか。

まず設問1の「中，高生のプログラミングに対する苦手意識が高い理由」に対する回答では，8名中6名の学生が「英語で書かれているプログラミング言語を正しく入力することが難しい」「用語が覚えづらい」「英語を苦手としている学生にとって抵抗がある」といった英語で構成されているプログラミング言語そのものへの抵抗感を挙げていた。その他に，「現在スマートフォンでインターネットなども利用できるため，パソコンが出来なくても困らずパソコンに対する苦手意識を持っているため」や「なぜ中学生でプログラミングを学ばなければいけないのかの意識付けができていない」,「何に使うのか分からない」といった理由もみられた。また，高校の授業でプログラミングを学んだ経験のある学生からは，「先生と同じものを入力する流れ作業になってしまう傾向にあるため，プログラミングを学ぶ意義や重要性が理解できない」という意見もあった。これらの回答から，学生がプログラミングを学ぶことに関して，必要性を理解しておらず難しいという印象だけが残ってしまっていることがわかった。

そして設問1で挙げられた苦手意識を克服するための工夫としては，「図やイラストを使いプログラムの流れを分かりやすく説明する」「簡単に作れるゲーム等、生徒のレベルにあったプログラミングを体験させることで興味を引き付ける」「画像や音など五感でプログラムに触れることができるような授業をする」「例題に生徒が取り掛かりやすいような身近に感じるものにする」など，単にテキストを扱うプログラミングではなく，画像や

音といった様々な媒体を扱うとともにゲームのような生徒の興味をひく題材を使って体験させることで，プログラミングが楽しいという印象を持たせることが重要だと考える回答が多かった。また，「日本語でプログラミングができる環境を利用する」や「プログラミングは英文法ではなく，パズルのようなものだという意識をつける」といった英語を使わないプログラミング教育を提案している学生も 3 名おり，そういった意味で「今回実習で使用した Scratch がプログラミング初学者にとって有効であると思う」と述べている回答もあった。

次に設問 2 の Scratch を使ったプログラミング教育のメリットに対する回答では，「説明なしに感覚的に使い方がわかる」「プログラミングの流れを想像しやすい」「視覚的にプログラムを作成することができる」「命令させたいことを自ら思考しながら、取り組むことができる」「日本語による（ブロックによる）プログラミングなので比較的簡単に組める」「文法などを気にせずにプログラムを組むことができる」といった，簡単にプログラミングを体験することができる点を 8 名の学生が挙げていた。

一方デメリットには，「Scratch を利用した授業では容易にプログラムが作成できるので、より高度なプログラミング作成に移行する際に難しいという印象を抱く」や「本格的なプログラミング言語が身につかない」，「本格的なプログラミング言語を学ぶときにやはりつまずいてしまう」といった日本語で文法も気にせず楽にプログラムが作成できるからこそ，その後のプログラミング教育にとって逆に基本知識とならない可能性があることを言及する意見が多く見られた。他には，「汎用性に欠ける部分がある」や「できることが限られてくる」など，ブロックプログラムの限界についての意見が挙げられた。

設問 3 のフィジカル・コンピューティングの教育効果に関しては，全員の学生が好意的な効果を挙げ，さらに受講した学生自身もプログラミングに対する意識が変わった，とても興味深かったと答えていた。例えば「モノづくりなどへの興味や関心をより高めることができる」や「中高生が普段使っているスマートフォンをプログラミングの授業で取り入れることができればプログラミングに対する興味につながる」，「従来の文字を入力するだけのプログラミングのイメージの転換につながり，プログラミングに対して苦手意識をもつ生徒が減ることが期待できる」といったプログラミングに興味をもつきっかけになることを挙げていた。さらには「パソコンの中で何かが起きているのではなく，自分たちの日常生活にどう取り込めるのか，それによってどのような問題が生じ，どのように解決していくのか考えさせることにより，興味がわかない生徒にも深く考える機会を与えることができる」や「これからの VR[4] や AR[5] の発展により，従来のプログラミングより生徒の創造力を掻き立てる効果がある」などの生徒の「自ら考える力を養う」効果を期待する回答もみられた。

以上のように，受講生のレポートの回答から，学生がプログラミングに苦手意識を抱く要因として，英語で構成されているプログラミング言語に対する抵抗感や文法の難しさの他に，学ぶ意義や重要性を理解していないことや，今やスマートフォンなどの携帯端末の普及により日常でパソコンを使う機会が少ないため，パソコンを使ったプログラミングに必要性を感じていないことがわかった。現在ではプログラミングはパソコンに限らず，タブレット端末などでも気軽に行える。そのため，2020 年の次期学習指導要領に向け，タブレット端末を使ったプログラミング教育を取り入れている小, 中学校も増えてきている。そのようなことからも，プログラミングを単体で教えるのではなく，タブレット端末を使

い日本語で利用することのできるビジュアルプログラミングを算数や理科，社会などの教科と組み合わせた授業を展開することは，プログラミングの必要性を自然と理解できる機会にもなり，2020 年からのプログラミング必修化は，現在の学生が抱いているプログラミングの負のイメージを払拭できる可能性があると考える。現に，小学校の各教科にプログラミング教育を取り入れ，生徒の理解力や興味が高まったという研究事例も報告されている（上出ら，2017)。

今回の実践により，大学生に対しても従来のテキストベースのプログラミング授業より，ビジュアルプログラミングの方が理解しやすく，アルゴリズムに集中してプログラムを組むことが出来ることが明らかとなった。プログラミングの基礎的な位置づけとして大学生に対してもビジュアルプログラミングを利用することは効果があると思われる。

フィジカル・コンピューティングにおいては，受講生の回答にみられたように，身近なものとプログラミングを組み合わせることにより，よりプログラミングに関心を持ち，学生自らが考え，創作するきっかけを与える可能性を秘めている。文部科学省が 2018 年 11 月 6 日に発表した「小学校プログラミング教育の手引」の改訂版でも，指導例として「プログラミングの楽しさや面白さ，達成感などを味わえる題材などでプログラミングを体験する例」が追加されているように，楽しさ，興味を持たせる指導方法が重要であることがわかる。またこれから IoT 技術がさらに身近なものとなる社会において，従来の「コンピュータ×プログラミング」あるいは Web 技術のような基礎的な知識だけでなく，「もの×プログラミング」を早い段階から学ばせることで，社会に役立つ ICT の利用法や活用法を提案できる人材育成にも繋がることが期待できる。

5．おわりに

本研究では，超スマート社会における人材育成で必要となるプログラミング教育について，現状を調査し，実践研究を通して効果的な授業設計について考察を行った。

わが国の情報教育は海外の先進国の中でも遅れをとっており，特にプログラミング教育に関しては教員のスキル不足や環境の未整備などにより，小・中・高では積極的に実施されていない状況が分かった。また，多くの大学生がプログラミングに対して「難しい」などの否定的なイメージを持っており，それはプログラミング未経験者やテキストベースのプログラミングを体験したことのある学生に顕著にあらわれていた。プログラミング教育の主な目的は特定の言語の習得ではなく「アルゴリズム構築能力」であるため，言語の文法などにより生じてしまう「難しさ」や「数学的要素」は，学習初期の段階では取り除く必要がある。その解決策の一つとして，本研究では，ビジュアルプログラミングツールとフィジカル・コンピューティングを組み合わせたプログラミング授業を設計し，教職課程の学生に向けて実習を行うことで，その効果を考察した。実習後の学生のレポートでは，「視覚的に分かりやすかった」「プログラミングを学ぶ意義が理解できた」などの肯定的な意見が得られた。学習において「できた」「わかった」という達成感とそれを学ぶ必要感は重要であり，そういった意味でも少しの作業で何かしら動くものが作れるビジュアルプログラミングは短いプロセスで達成感を得ることができるため，プログラミング初学者にとって効果的なツールであるといえる。また，身の回りに存在するセンサー技術を体感できるフィジカル・コンピューティングと組み合わせて学習することで，必要感も得られや

すいと考える。

しかし，本格的にシステムを構築するには，C や Java のようなテキスト型のプログラミング言語を習得する必要がある。そのためには，ビジュアルプログラミング言語から従来のプログラミング言語への移行を考慮した授業設計も必要であり，学習の発展性が今後の課題である。

〔注〕

1）イギリスの Raspberry Pi Foundation が学校教育用に開発した内蔵ハードディスクを搭載しない小型のコンピュータ。
公式サイト：http://www.raspberrypi.org （2018/10/28 アクセス）

2）ちっちゃいものくらぶが開発した小型センサーボード。
公式サイト：https://tiisai.dip.jp/,（2018/10/28 アクセス）

3）統合開発環境のひとつであり, ATMEL 社がリリースしているマイクロコントローラ。オープンソース・ハードウェアという特徴がある。
公式サイト，http://arduino.cc/,（2018/11/3 アクセス）

4）Virtual Reality の略で，一般的に「仮想現実」と訳される。

5）Augmented Reality の略で，一般的に「拡張現実」と訳される。

〔参考文献〕

［1］上出吉則，辰巳丈夫，村上祐子（2017）「プログラミングの算数数学教育での効果と検証－生徒の創作した Scratch プログラム教材を授業で活かす－」『情報教育シンポジウム情報処理学会』pp.239-246.

［2］大岩元（2014）「識字教育としてのプログラミング」『情報教育シンポジウム 2014 論文集』2014（2），pp.127-132.

［3］大場みち子，伊藤恵，下郡啓夫（2015）「プログラミング力と論理的思考力との相関に関する分析」『情報処理学会研究報告』2015-IFAT-118（2）, pp.1-4.

［4］経済産業省（2016）「IT 人材の最新動向と将来推計に関する調査結果」『News Release』.

［5］小泉力一（2009）「学習指導要領改訂における情報教育」『日本科学教育学会年会論文集』33（0），pp.31-34.

［6］小泉力一, 佐藤義弘（2009）「全国アンケート調査で見る情報科教育の現状」『会誌「情報処理」』Vol.50，No.10，pp.1005-1008.

［7］国立教育政策研究所“OECD 生徒の学習到達度調査（PISA）”
http://www.nier.go.jp/kokusai/pisa/index.html,（2018/11/25 アクセス）.

［8］斎藤俊則（2008）「日本教育大学院大学における情報教育」『日本教育大学院大学紀要』1，pp.81-90.

［9］佐藤万寿美（2012）「高等学校全体の教科「情報」の状況について」『大学教育と情報』No.1，pp.2-6.

［10］シリフグリ・キラム，菊地章（2005）「情報教育の国際比較」『鳴門教育大学情報教育ジャーナル』2，pp.31-39.

［11］総務省（2016）「平成 28 年版情報通信白書（PDF 版）」
http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/pdf/,（2018/11/30 アクセス）.

［12］総務省（2017）「若年層に対するプログラミング教育の普及推進事業，平成 29 年度行政事業レビュー「公開プロセス」補足説明資料」『情報流通行政局』.

［13］中西渉（2016）「高校におけるプログラミング教育－愛知県の状況と実践事例の報告－」

『会誌「情報処理」』Vol. 57, No.4, pp.358-361.
［14］日本経済再生本部（2016）「成長戦略プロジェクトに係る検討課題」『第 26 回産業競争力会議』参考資料 1.
［15］内閣府（2013）「平成 25 年度我が国と諸外国の若者の意識に関する調査（PDF 版）」.
［16］原田悦子（1992）「文系学部におけるプログラミング教育の意義：健全なユーザ育成のための情報教育の視点から」『社会労働研究』3（3/4），pp.119-134.
［17］東本崇仁（2017）「プログラミング教育必修化の意義と現場での教育」『教育システム情報学会誌』Vol. 34, No. 2, pp. 79–81.
［18］樋口耕一（2004）「テキスト型データの計量的分析 :2 つのアプローチの峻別と統合」『理論と方法』19（1），pp.101-115.
［19］樋口耕一（2014）「社会調査のための計量テキスト分析」『ナカニシヤ出版』.
［20］堀田龍也（2018）「新学習指導要領における情報教育の動向」『会誌「情報処理」』Vol.59, No.1, pp.72-79.
［21］森秀樹（2010）「Scratch を用いた文系大学生向けプログラミング教育『日本教育工学会論文誌』34, pp.141-144.
［21］森畑明昌（2016）「東京大学における全学プログラミング教育」『会誌「情報処理」』Vol. 57, No.4, pp.362-365.
［22］文部科学省（2000）「高等学校学習指導要領解説 情報編」『開隆堂出版』.
［23］山本樹，國宗永佳，香山瑞恵（2010）「アルゴリズム的思考と論理的な文章作成力との相関についての考察」『日本教育工学会研究報告集』2010（5）, pp.171-176.
［24］吉田葵・来住伸子・阿部和宏（2015）「大学における授業科目「小中高におけるコンピュータ教育実践報告」『情報処理学会研究報告』Vol.2015-CE-129, No.26.
［25］OECD（2015）“OECD Skills Outlook 2015 : Youth, Skills and Employability”, *OECD publishing*.
［26］Mitchel Resnick et al.（2009）“Scratch: Programming for All, Programming for All”, *Communications of the ACM*, Vol. 52（11）, pp. 60-67.

本研究は，平成 28 年度および平成 29 年度の日本大学商学部情報科学研究所による共同研究（研究代表者：所伸之）の研究成果の一部である。

Abstract

This study aims to investigate the current state of information and communications technology (ICT) education in Japan. According to the data reported by the MIAC, in Japan, the current state of ICT education lags in comparison with other developed nations and the Japanese youth clearly lack basic IT skills in comparison with their American and British counterparts.

To analyze the factors that hinder the growth of IT skills in youngsters, this study involved taking a questionnaire on computer programming, which was attempted by university students. Results show that many students lacked any programming experience until they entered university and were often not comfortable with this field. A course was then conducted on visual and physical programming and the students were encouraged to consider programming education from the viewpoint of students and instructor. Furthermore, the study examined effective programming education for students specializing in information sciences.