

Vol. 110

## CONTENTS

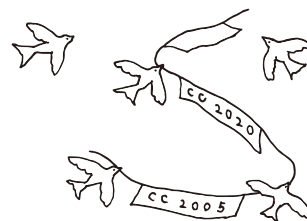
【コラム】 CC2020 プロジェクトと情報系カリキュラムについて…高田 眞吾

【解説】 脳科学からプログラミング教育を考える—プログラミング的思考は汎化するのか?—…細田 千尋

【解説】 遠隔環境による高大接続プログラミング演習の実践報告…赤澤 紀子・久野 靖

## COLUMN

### CC2020 プロジェクトと 情報系カリキュラムについて



CC2020 (Computing Curricula 2020) は、情報分野の大学のカリキュラムにかかわる国際的なプロジェクトであり、ACM および IEEE-CS を中心に 2016 年に開始しました。そのプロジェクトに日本からは、(当時 IPA, 現在早稲田大学および VJP (株) 取締役の) 林口英治さんと私が参加しています。本来は、本年 (2020 年) 3 月 7 日 (土)、全国大会のイベント企画において紹介する予定でしたが、新型コロナウイルスによる現地開催中止を受け、その企画自体が中止となりました。そこで、本コラムで CC2020 を簡単に紹介しようと思います。

15 年前、ACM および IEEE-CS を中心に CC2005 プロジェクトのレポートが公開されました。本会が策定した J07 や J17 はこの CC2005 を基にしています。CC2020 は CC2005 のレポートを更新することを主目的としており、2020 年末または 2021 年初めにレポートが公開される予定です。その中で、現在の情報分野を俯瞰し、今後について議論しています。なお、CC2020 はあくまでも情報分野を俯瞰することを目的としており、コンピュータサイエンスなど、特定の分野の詳細を決めるプロジェクトではありません。

さて、CC2020 の大きな柱に、コンピテンシがあります。従来は、知識に基づいた教育だったのに対し、今後はコンピテンシに基づいた教育への移行を提言しており、コンピテンシの枠組みを提案しています。コンピテンシに基づいた教育では、単に知識を学生に詰め込むのではなく、カリキュラム自体が、スキルやディスポジション (disposition) を意識するべきという考え方です。知識があるだけでは不十分であり、それをどの程度活用できるか (=スキル)、また、積極性・情熱・柔軟性など気質や態度 (=ディスポジション) にかかわる事項も重要視するべきということです。

実際にコンピテンシに基づいた教育に移行するためには、情報の各分野がカリキュラムガイドラインを今後更新する必要があります。IT (Information Technology) 分野<sup>☆1</sup>では、2017 年にすでにコンピテンシを中心としたカリキュラムガイドラインを公開しています。IS (Information Systems) 分野では、次期カリキュラムガイドラインをコンピテンシに基づくように、現在策定作業が行われている最中です。

CC2020 が提言しているコンピテンシは、一夜で根付くとは思えません。すでに書いた通り情報の各分野の新しいカリキュラムガイドラインが必要ですし、そもそも考え方を考える必要があります。日本に CC2020 で定義したコンピテンシをそのまま取り込むのではなく、日本流にアレンジしないといけないことも考えられます。少なくとも数年はかかるでしょうが、変化の波が近づいています。

高田眞吾 (慶應義塾大学)

☆1 ここでいう IT は、日本国内でよく言葉として聞く IT と対象が異なります。

# 脳科学からプログラミング教育を考える —プログラミング的思考は汎化するのか?—

細田千尋

帝京大学戦略的イノベーション研究センター／東京大学総合文化研究科

## プログラミング教育が目指すものとは？

2020年から全国で施行され始めた小学校学習指導要領総則において「自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」が「プログラミング的思考」と定義され、それを育成するための一手法としてプログラミング教育が採用された。ここでは、プログラミング言語を覚えたり、プログラミングの技能を習得したりといったことではなく、「論理的思考力を育むとともに、プログラムの働きや良さ、情報社会がコンピュータをはじめとする情報技術によって支えられていることなどに気づき、身近な問題の解決に主体的に取り組む態度やコンピュータ等を上手に利用してよりよい社会を築いていこうとする態度などを育むこと、さらに、教科等で学ぶ知識及び技能等をより確実に身に付けさせることにある」と示されている。つまり、プログラミング能力(コーディングなど)自体を伸ばすことが目的ではなく、学習の場で直面するさまざまな事象において、自分の意図する行動を実現するための手続きを考えるといった論理的思考(プログラミング的思考)を養う方法としてプログラミング教育を採用し、それを通して主体的・対話的で深い学びを実現することを目的としている、というのである。

実際のところ、プログラミングを通じて学べるこ

とは何か？ 実際にプログラミングを学ばせたことで、論理的思考力や、創造性が高まったか？ さらに、その他の教科(数学など)での成績が上がったか？ といった因果関係をダイレクトに実験的に調査し、結論を出しているものはない。ただし、プログラミングを学ぶことで、新しいアイデアを試す創造性、複雑な問題を紐解くための推論、いかに最小手数で目的にたどり着けるかという合理的で論理的な考え方、粘り強く前に進む方法、物事を深く理解する方法、ほかの人と協力して物事を進める方法を学ぶことができる、と述べているものは多い。さらに、プログラムを作成していく中で身に付いていくものとして、数学的概念の獲得、アルゴリズム的思考、自己の知識状態を把握する力、明確な表現力や説得力、デバッグ体験による柔軟な思考、などが挙げられている。

確かにここに挙げられたような要素は、プログラミングをしていく上で必須のものであり、プログラミングのスキル獲得・向上の際には、これらの能力も獲得できている、あるいは、そもそもこれらの要素についての素養の高い人たちはプログラミングの獲得が早いという相関関係が存在すると考えられる。しかし、このプログラミングスキルと深く関連のある上述のような能力が、プログラミング以外の場にも汎用し得るものなのだろうか(プログラミング教育を通してプログラミング的思考が構築できれば、それ以外の場でもプログラミング的思考に沿った論理的思考で物事を進めていけるのか)？ そしてそもそも、プログラミング教育によって、大半の人が

プログラミング的思考やプログラミングスキルを獲得することができるのだろうか。プログラミング教育、プログラミング的思考が、すべての教科を学んでいく上で手段、手法として初等教育から用いられていく場合、仮に、プログラミングに対する適正が著しく低い特性を持つタイプの人が出たとしたらどうなるのだろうか。

こうした背景および疑問を踏まえて本稿では、プログラミング的思考（自分が意図する一連の活動を実現するための論理的に考える力）の獲得を目的としたプログラミング教育の実施について筆者の考えを、これまでの認知科学、社会科学、脳科学の研究成果の観点から述べる。

### 領域固有の知識が他の領域の問題解決に有用なのか？

認知科学の分野では、資質・能力・思考には、教科や学習領域固有のものもあれば、一般的、汎用的なものも存在する、とされている。特に近年、日本だけでなく諸外国でも教科固有ではなく、教科に依存しない汎用的な能力の育成が強調されていることは間違いない。プログラミング教育の導入は、この流れの中で決定したことであろう。実は、認知科学の分野では、古くは、学習が習熟する過程における、「領域固有の知識」の重要性が説かれてきた。ある領域の問題解決を図るとき「他の領域に固有の知識」を習得していても、問題解決には活かされない(転移しない)と考えられていたためである。加えて、複数の領域に汎用される「一般的な知識」の蓄積は、習熟過程における問題解決には、効果的ではないことが研究から示されてきた歴史も持つ。これは、ちょっと想像してみると納得がしやすい。たとえば、今現在プログラミング教育と同程度に、幼児一児童期の子どもを持つ親の関心の1つに将棋がある。我が子を将棋士にしたいということではないにもかかわらず、将棋教室に入れたい、藤井聡太二冠が幼少自體に学んでいた方法と同じことを我が子にさせたい、という教育熱心な親御さんが多

い。では、将棋を学んで得られる戦術的な知識や直感力、想像力が、学習科目における問題解決に役立つのか？ もっと端的にいうなら、将棋を習っていた子どもは、学校での学習成績が良くなるのであろうか？ これについては、残念ながら、その関係性を示すような科学的根拠はない。似たような話では、バイリンガルの方が記憶力が良く、学校成績が良くなるという研究についても、いまだにその関係性は明確になっていない。また、論理的思考の育成を目的とした教育では、プログラミング以前には、数学が効果的であるとされている流れが存在していたが、これもいまだに、数学の獲得によって、論理的思考力が向上するかについて、決定的な結果は出ていない。

一方で、汎用性が高いと考えられているものの中に、メタ認知がある。メタ認知とは、認知や知識に関する認知である。自分の思考や行動そのものを対象化して認識することにより、自分自身の認知行動を把握する思考である。これにより、認知、知識、プロセス、認知状態・感情状態を意識的、熟慮的にモニタし、自分自身の状態を判断することで、自分の考えの矛盾に気づき、課題の特性を把握した上で解決方略を修正していくといった活動を行うことができる。しかし、子どもは最初から一般的なメタ認知能力を発揮できるわけではないようだ。研究によれば、14歳頃まで、子どものメタ認知スキルは領域固有あるいは課題固有のものである<sup>1)</sup>。つまり、14歳頃までは、たとえば、プログラミング学習におけるメタ認知は、あくまでもプログラミングという教科に限って利用できるものだけである。それが、15歳頃からは、領域一般的なメタ認知スキルにまとまる、という。つまり、そこで得たメタ認知スキルを、その他の学習や生活行動の中でも利用できるようになるという結果が示されている。しかし別の研究では、大人でもメタ認知の正確さが課題の種類や、教科間で異なることも示されている<sup>2)</sup>。つまり、概して汎用性があると言われるメタ認知でも、必ずしも汎用、一般的とはいきれない。特に、この知見をもとにプログラミング的思考を考えたとき、年齢



によって領域固有なのか汎用的なのか(転移できる能力なのか?)が変わってくるという所見はきわめて重要であろう。また、プログラミング的思考という名称で定義された中身が、本当にすべて汎化し得るのかに加え、そもそもこの思考が、プログラミング教育によって得られるものなのか、を見極めることは重要であろう。特に、小学校から、プログラミング的思考が汎用性のあるものであり、かつ、プログラミング教育によって獲得できる思考である、として教育することの最適性にかかわる問題にもなり得る。単純にプログラミング的思考という、それ自体の定義に曖昧性が残るものに対し、この思考が教科横断的スキルと見なし、すべてを決定していくことは早計かもしれない。プログラミング教育によるプログラミング的思考の育成、を教育目標として、カリキュラムを構成するのならば、その特徴や構造、プロセスについての実証的な吟味が必要不可欠だと考える。

これらを踏まえ、次に、脳科学の観点から、プログラミングについて研究されているものについて紹介する。

## ■ プログラミングスキルと脳

### □ プログラミングスキル獲得を予測する要因と

#### 脳の特徴

プログラミング自体の学習能力には個人差が大きく、優れた生徒と苦手な生徒に大きく分かれることが経験的な観点からも知られている。この傾向は実際のデータを用いた研究でも実証されており、情報系大学生であっても約4割はプログラミングスキルを十分に獲得できないという報告すらある<sup>3)</sup>。こういったプログラミングの学習能力の個人差はどのような因子に求めることができるのだろうか。

プログラミング能力にかかわる因子の研究は古い歴史を持ち、過去何十年間にも渡って活発に行われてきた。主な因子として、性別、抽象化能力、因果関係の理解能力、自閉症的性格傾向の有無、短期的に記憶を貯蔵するといったワーキングメモリ容量、

一般知能、理解の深達度、認知的要素が、どのくらい周りの環境による影響を受けやすいかという、場依存性一場独立性、などがある。さらに、問題を定式化して解決する能力であるメンタルモデルの構築・運用能力、自分はプログラミングができる、というプログラミング学習自体に関する自己効力感、空間情報処理能力、数学的能力がプログラミング能力獲得に密接にかかわると考えられている。実際、プログラミングに関する自己効力感、高校卒業試験における数学的能力、一週間のコンピュータゲーム時間の3因子によって8割以上の確率で履修結果の予測に成功したことを報告した研究もある。数学的能力と、プログラミング能力、空間情報処理能力が相関することは双生児を用いた研究からも報告されており、数学的能力の優れたものは、空間情報処理課題においても、その正答率が高く、課題実施時には、左右の頭頂葉の活動が高まることが分かっている。さらに、頭頂葉の中にある頭頂間溝は、数的情報の処理と空間情報の処理の両方で重要な役割を果たす場所でもあり、この脳領域はプログラミング学習処理の個人差を担っていると考えられる。事実、プログラミング未経験者に約12週間にわたってJAVAプログラミング学習介入を実施した筆者の実験から、頭頂葉の一部と前頭前野の発達度合いが、プログラミングスキル獲得可否の予測要因である可能性が示唆されている<sup>4)</sup>。

### □ プログラミング実施時の脳活動

次に、プログラミング実施時に脳が活動する部位について考えてみる。脳活動は、神経活動に伴う脳血流の変化を主に計算することで、ダイナミックな脳活動を可視化することのできる非侵襲的脳機能イメージング(functional Magnetic Resonance Imaging fMRI や functional Near-infrared spectroscopy : fNIRS など)により計測ができる。プログラミングでは、データ構造の操作能力が必要と考えられ

る。このデータ構造操作課題時の神経活動についてfMRIやfNIRSを用いて調査した結果、運動前野(BA6/図-1の6番の領域)や補足運動野(BA6)、上頭頂小葉(BA5)、ウェルニッケ野(BA22)、ブローカ野(BA44, 45)、前部島皮質(BA13)、背外側前頭前野(BA9)などが活動することが示されている。また、被験者にソースコードの理解課題を行わせた際には、運動前野(BA6)、下頭頂小葉(BA40)、下前頭回(BA44)、中側頭回(BA21)の活動が増加し、帯状回前部と後部の活動が低下することが示された。下頭頂小葉(BA40)および運動前野(BA6)は、一般に注意力、作業記憶、または問題解決を必要とする認知作業においてしばしば活性化される領域である。特に下頭頂小葉はソースコードに見られるような数字と文字の組合せによる関係性把握課題において活動し、中でも数学的アルゴリズムの運用課題においてその活動が顕著であることが報告されている。また運動前野は構文理解におけるワーキングメモリ機能に加え、注意ネットワークの一部として注意機能にもかかわっており、ソースコードに注意を払いながら理解するような状況で、これらの領域の活動が高まっていたと考えられる。下前頭回(BA44)、中側頭回(BA21)はいずれも言語処理の異なる側面に対応した領域である。中側頭回後部に位置するブロードマン21野(BA21)は単語レベルでの意味処理に関連しており、下前頭回に位置するブロードマン44野(BA44)は統語レベルでの意味処理にかかわる。人工言語を用いた課題でもこれら

の領域がかかわることが報告されている。この下前頭回(BA44)と中側頭回後部(BA21)は相互性の神経接続がなされており、この接続により、適切な語彙が選択され、全体的な文脈に統合することが考えられおり、プログラム理解課題においても同様の活動が示されたと考えられる。

では、帯状回前部と後部の活動の低下は何を意味しているのだろうか。帯状回前部と後部はともにデフォルトモードネットワークと呼ばれる自己意識にかかわるシステムを構成しており、何らかの認知課題を行うときには活動が低下することが知られている。この実験ではソースコードが複雑になるほど、これらの領域の活動が低下しており、プログラミングの理解を妨げないように抑制的に活動していたことが考えられる。以上のように総じてプログラミング理解には言語処理機能や注意機能にかかわる領域の活動増大と自己意識にかかわる領域の活動低下が見られたものの、自然言語処理と人工言語処理にかかわる脳活動は異なるという議論もあり、この面でのさらなる研究が望まれている。

このようにfMRIを用いてプログラミング理解過程を調査した研究がなされているものの、fMRIは測定が時間的にも金銭的にもコストが高く、近年では他の生理学的指標を用いる試みもなされている。その中で、fMRIと瞳孔の変化について同時測定を行ったものがある。そこでは、瞳孔の散大がプログラミング課題の難易度および脳活動を反映する傾向があることを報告している。また脳波や皮膚電気抵抗、心拍信号を用いて自然言語とプログラミング言語の理解の違いを検出できるかについて研究を行い、心拍信号がプログラミング理解の最も高い予測精度を示した研究もある。

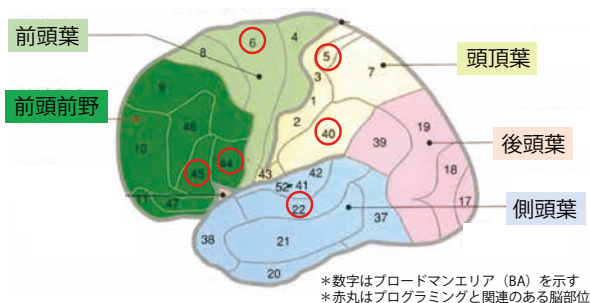


図-1 プログラミングに関連する脳領域

## □プログラミングスキル獲得によって脳が変化する?!

可塑性はヒトの脳が本来的に持ち合わせている性質である。内的・外的環境に合わせて、脳の解剖学



的構造や機能が変化していくことを示す。近年、アスリートのように、何年にもわたるトレーニングの末、高度な能力を持ち合わせたスポーツ選手の脳で、どのような脳可塑性を表象しているのか研究が進んでいる。実際、スポーツ選手だけでなく、音楽家など高度な技術を持ったプロの脳機能構造特性が明らかにされていると同時に、それらの技術を獲得することによって、どのように脳が可塑的に変化するかが示されてきている。スポーツや楽器演奏などの運動技術に限らず、筆者の研究から認知能力（英語力など）および非認知能力（やり抜く力など従来定量的に測る指標のなかったもの）の獲得による脳の可塑性<sup>5)</sup>も明らかにされ、学習効果を示す客観的指標としても、脳機能構造が有用である可能性が提唱されてきた(図-2)。

それでは、プログラミング能力、あるいは、プログラミング的思考に関連する要素における、脳の可塑性はあきらかにされているのだろうか？ プログラミング的思考を構成する要素のキーワードである、創造性、論理的思考については、いくつもの研究が示されている。たとえば論理的思考(推論)については、前頭葉下部の賦活が示され、右前頭葉下部が強く賦活する被験者ほど推論成績が良いという正の相関が認められている。また、創造性についても、創造性が高い人では、デフォルトモードネットワークや背外側前頭前野などを含む実行機能ネットワークの活動が高いことが分かっている。

また、プログラミングエキスパート（職業プログラマー）では、初心者に比べて、プログラミングコー

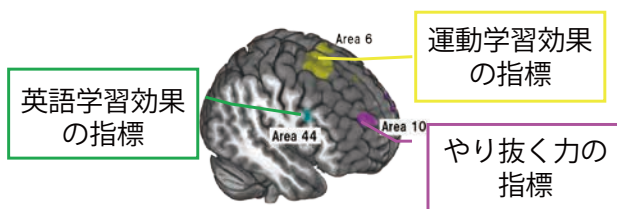


図-2 脳構造情報が能力の個人差を示す定量的指標になり得る

ド理解時に、左前頭葉下部、右中前頭回、右下前頭回三角部に活動が見られていたことが報告されている。この脳活動の差はエキスパートと初心者の認知戦略の違いを示していると考えられる。また、初心者がプログラミング学習をし、能力を獲得することで、前頭葉下部などの領域に可塑的变化が見られた。さらに、プログラミングエキスパート（日常的にプログラミングを行う大学院生）では、初心者に比べてこの領域が発達していたことも明らかになっている(図-3)<sup>4)</sup>。

これらの脳領域は、先に示した論理的思考力を司る領域でもある。ただし、筆者の研究では、プログラミング学習に伴って、一般的な論理的思考力の向上が見られるというところまでは明らかにできなかった。しかし、プログラミングスキルの獲得に伴って可塑的变化が見られた前頭葉下部が、論理的思考の神経基盤の重要な一部でもあることを考慮すると、少なくとも、プログラミングスキル獲得に必要な領域固有の論理的思考は向上している可能性が考えられる。

## プログラミング教育に対する提言

世界的にプログラマーの需要が増している現代において、早いうちからプログラミング教育を行うこと自体は、子どもの視野や将来的な職業の幅を広げるといふ点からも、早期教育によるハイスキルの獲得、という観点からも良いことである。ただし、プログラミングを教育することの意義を、プログラミ

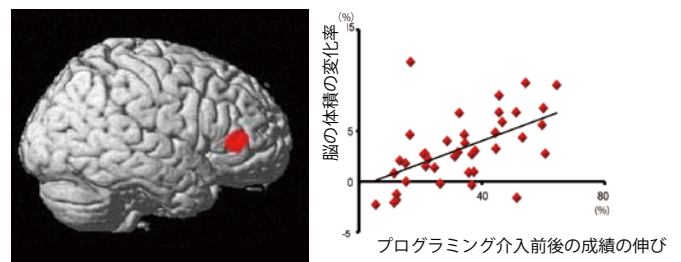


図-3 プログラミングスキル獲得によって変化する脳構造

-【解説】脳科学からプログラミング教育を考える—プログラミング的思考は汎化するのか?—

ングスキルの獲得以外の、そこから汎用される能力とし、さまざまな科目において利用する仕組みを作るには、もう少しその妥当性を検討していく必要があるのではないだろうか。数学や運動、英語などと同様、プログラミングも得意不得意が生じることは明らかである。さらに、筆者らの研究から、そこには、脳神経基盤差(脳機能や脳構造の違い)が存在している可能性が示唆されている。そのようなスキルをすべての科目に導入すれば、いくらそれが必要な能力だとはいえ、不得意な人が、すべての科目において自己効力感を失っていくことにもつながりかねない。そうなれば、元の木阿弥である。事実、学校生活においては、たった1つの失敗による自己効力感の低下が、生活全般の自己効力感の低下に結びつくことが知られている。

一方で、自分の作りたいもの(目標)に対し計画をたて実行することで、目標が達成されていくプロセスは、自己効力感を一気に高める作業であり、より一層の学習を促進する素晴らしい経験ができる教育にもなり得る。文部科学省の2007～2018年度の全国学力・学習調査によれば、年々、学生の自己効力感が上がってきている中で、プログラミング教育の導入が、この傾向に良い影響を及ぼすことが重

要であろう。そのためには、科学的にも実証し得る、効果的なプログラミング教育法、プログラミング的思考育成法を早急に明らかにしていくことも重要課題の1つである。今後、プログラミング的思考育成のために取り入れるプログラミング教育がどのような影響・効果をもたらすのか見極めながら詳細に検討していく必要があるだろう。

#### 参考文献

- 1) Veenman, M. V. J., Hesselink, R. D., Smeets, S., Liem, S. I. E. and Van Haaren, M. G. P.: Assessing Developmental Differences in Metacognitive Skills With Computer Logfiles: Gender by Age Interactions, *Psychological Topics*, 23(1), pp.99-113 (2014).
- 2) Veenman, M. V. J., Van Hout-Wolters, B. H. and Afflerbach, P.: *Metacognition and Learning: Conceptual and Methodological* (2006).
- 3) Hook, L. J. and Eckerdal, A.: On the Bimodality in an Introductory Programming Course: An Analysis of Student Performance Factors, 2015 International Conference on Learning and Teaching in Computing and Engineering, IEEE (2015).
- 4) Hosoda, C., Hamada, M., Maeshima, H., Nonaka, Y. and Okanoya, K.: Predictor of Programming Language Learning success: The Development of The Inferior Frontal Cortex and The Supramarginal Cortex Organization for Human Brain Mapping (2018).
- 5) Hosoda, C., Tsujimoto, S. and Tatekawa, M. et al.: Plastic Frontal Pole Cortex Structure Related to Individual Persistence for Goal Achievement, *Commun Biol* 3, 194 (2020).

(2020年8月6日受付)

細田千尋 chihirohosoda@gcc.u-tokyo.ac.jp

帝京大学戦略的イノベーション研究センター講師/兼 帝京大学大学院総合文化研究科特任研究員。脳機能・構造から、認知能力(英語やプログラミング等)や非認知能力(やり抜く力)の個人差を予測し効果的な学習法について研究。



## 2020年度小中高教員【秋】のキャンペーン

<https://www.ipsj.or.jp/member/kyoinwaribiki-aki-2020.html>



**期間** 2020年10月1日～11月27日 (最終日は15時まで)

**対象** 小中高校(相当する教育機関を含む)に教職員として勤務されている方(現職)に限ります

### キャンペーン内容

☆新規の正会員: 入会金(2,000円)が免除となります

2020年度と2021年度の会費(10,800円)が半額(5,400円)に割引されます

☆既存の正会員: 2021年度の会費(10,800円)が半額(5,400円)に割引されます

※会員サービス内容は正会員と同じです

### 教員にとってのメリットとは

- 会誌「情報処理」(毎月)や「ジュニア会員通信」などの本会のお知らせが読める
- 教員免許更新講習を会員価格で受けられる
- 中高生情報学研究コンテスト/Exciting Coding! Junior/初等中等教員研究発表セッションなど生徒向けや教員向けイベントを情報教育に活用できる
- 「情報」に関する豊富な知識を得ることができる
- 情報処理学会の教育委員会が発信するトピックスやパブリックコメントをいち早くキャッチできる



# 遠隔環境による高大接続プログラミング演習の実践報告



赤澤紀子 久野 靖

電気通信大学

## プログラミング教育と高大連携

学習指導要領が改訂され、2020年度から小学校でプログラミングが必須の内容となった。中学校、高等学校でも、改訂が順次行われ、2022年度からすべての高校生が情報Iにて、プログラミングを学習することになる。プログラミングについて、強い意欲を持つ高校生も今以上に増えると考えられる。そこで、筆者らは、所属する大学の初年次情報基礎科目の内容を、そのままe-ラーニングにより提供する「大学授業科目先取学修」の環境を整備し、「高大連携・基礎プログラミング」を2018年度の試行を経て、2019年度から正式実施した。プログラミングに興味のある、進んだ内容を学びたい高校生に活用してもらうことを目的としている<sup>☆1</sup>。

## 遠隔環境でのプログラミング演習

### □ 実施概要

「高大連携・基礎プログラミング」は、本学の初年次の情報基礎科目「基礎プログラミングおよび演習」の内容を高校生に提供するものである。

### 初年次の情報基礎科目

「基礎プログラミングおよび演習」は、Ruby 言語と C 言語を用いてプログラミングの基礎概念を演習しつつ一定水準まで「自力でコードが書ける形で」身につけることを目標にする。この科目は、反転授業形式の

<sup>☆1</sup> 本稿は国立情報学研究所【第5回】4月からの大学等遠隔授業に関する取り組み状況共有サイバーシンポジウムの発表に基づく。  
<https://www.youtube.com/watch?v=ewlcCXb7Q6Q>

全15回で実施している。授業前に、LMS (Learning Management System) 上にて提供されている教材を用いて各回の内容を予習してくることを前提として、授業(対面)時、学生はプログラム作成演習を行い、教員は補足説明と質問に対する回答を行う。授業後に、課題演習に取り組み、LMSにて課題レポートを提出する。  
「高大連携・基礎プログラミング」

「高大連携・基礎プログラミング」は、表-1のスケジュールで実施し、本学での3回のスクーリング以外は、高校生の学校や自宅で学習できるように配慮している。教材と提出課題は、初年次の情報基礎科目と同じものをLMSに用意した。

- (1) テキスト：各回 A4 判 10 ページ前後
- (2) 演習ガイド：各回の演習で注意すべきことや進行案を記載したもの
- (3) 講義動画：板書、スライド、コンソール画面を用いて、プログラムおよび実行結果を示し、各回の内容を講義形式で解説する。
- (4) 確認問題：前回の演習内容を復習するための短冊形式の確認問題

表-1 スケジュール

| 日程              | 内容                               |
|-----------------|----------------------------------|
| 2018.12         | Web ページで広報開始                     |
| 2019.04.01 ~ 25 | 応募受付期間                           |
| 2019.05.07      | 選考結果の通知                          |
| 2019.06.08      | 第1回スクーリング                        |
| 2019.06 ~ 07    | 受講期間前半 (遠隔)<br>#1 ~ #6 の学習・課題提出  |
| 2019.07.27      | 第2回スクーリング                        |
| 2019.07 ~ 12    | 受講期間後半 (遠隔)<br>#1 ~ #15 の学習・課題提出 |
| 2019.12.21      | 第3回スクーリング                        |



- (5) 質問フォーラム：演習内容に関して不明なことがあるなどの場合には、フォーラムを使って質問をする。質問に対応したアドバイスを教員やティーチングアシスタントから回答する。
- (6) 提出課題：各回、複数の難易度の課題を出題し、高校生は、その中から1つ以上を選択してプログラムを作成し、レポートとして提出する。

高校生は遠隔で内容の理解、課題演習を行う。質問は、LMSのフォーラムにて随時受付、回答するようにした。これにより、本学から遠い高校であっても参加することができる。また、学習内容は、大学初年次のままであるが、取り組む演習には自由度があり、高校生にも学習できる内容となっている。

### □ 募集形態

対象は高等学校1～2年生（中等教育学校4～5年生）とし、特定学校への偏りを避ける観点から、各校定員は、3～8名とした。教材は前述のとおり高校や自宅等で利用できるよう配慮しているが、RubyやC言語のプログラムを作成し動かすための環境については、一定のサポートが必要であり、なおかつ学校や個人ごとに環境が異なる。そこで、参加募集を高校単位で行い、高等学校ごとに窓口担当の教員の配置と、遠隔環境で受講するため、次の3点をお願いすることとした。

- (1) 演習環境構築のサポート：受講に必要なプログラミング言語処理系および実習に使用するアプリケーションを生徒が使用でき、問題なく実習が行えるようにサポートを行う。



図-1 スクーリングの様子

- (2) 進捗確認：定期的に（少なくとも2週間に1回以上、ただし休暇期間中は除く）ミーティングを開催し、受講の様子を聞き取り、受講上の問題等について相談にのる。
- (3) 問題解決サポート：受講上の問題が生じた場合は、本学と連絡をとり、問題の解決にあたる。

## ■ プログラミング演習の実施の様子

### □ 参加生徒について

参加校9校44名(1年生14名, 2年生30名)であった。参加した時点での進路希望は、理系だけではなく文系の生徒もいた。また、部活動や委員会活動や習い事をしているか調査したところ、プログラミングに関係する部活動を行っている生徒もいたが、吹奏楽部、バスケット部、社会部など異なる部活動を行っている生徒が多く、「生徒会、文化祭実行委員会、自転車競技部」「マンドリン部、保健委員、ダンス」など放課後に複数の活動を行っている生徒が多数いた。

### □ 環境構築

演習環境の構築は、学校によってさまざまであった。生徒にノートパソコンを必携させている学校では、生徒のノートパソコンに環境を構築した。学校のコンピュータ教室などのコンピュータに演習環境を構築した学校もあった。一方で、生徒必携のノートパソコンもなく、さらに、コンピュータ教室のパソコンに、自由にソフトウェアをインストールできず、高校で生徒の演習環境構築のサポートができない学校もあった。この学校の生徒たちは、自宅のパソコンに環境を構築していた。

### □ 第1回スクーリング

大学のコンピュータ演習室にて実施した(図-1)。はじめに、本学の教員より、LMS上の教材の使い方および、学習の進め方の説明を受け、教材#1の内容を授業形式で学んだ。



## □ 学習の様子

スケジュールに沿って、高校や自宅など生徒の学習環境にて、学習内容 #1 ~ #15 の学習を行い、LMS を使い課題提出を行った (表-2)。また、大学のコンピュータ教室にて、第 2 回、第 3 回スクーリングを実施し、総合演習に関するプレゼンテーションを行った (図-2)。学習時に分からないことがあれば LMS 上のフォーラム (掲示板) で質問してもらおうこととした。毎週、1 回分の課題の提出を目安として提示したが、各回の課題は特に期限は設けず、でき次第提出してもらい本学側で採点しフィードバックを与えるようにした。

課題を毎週提出する高校生はほぼおらず、受講期間前半は、7 月の定期考査が終わったあたりから提出が増え、第 2 回スクーリング (プレゼンテーション) に向けて、ピッチを上げて課題に取り組んでいる様子であった。#6 の課題を提出した高校生は 21 名 (全参加生徒のおよそ半数) であった。

#6 の総合演習の課題は、“自分 (たち) が「美しい」と思う画像の生成”である。通常より詳しい課題レポートを作成した上で、プレゼンテーションを実施することとした (図-2)。7 月下旬に実施したが、スクーリング日までに学習が進まなかった高校生もいたため、追加のスクーリングと高校単位のプレゼン

表-2 学習内容と課題提出数

| 各回の課題                     | 言語   | 課題提出 |
|---------------------------|------|------|
| #1 プログラミング入門;さまざまな誤差      | Ruby | 30   |
| #2 分岐と反復;数値積分             | Ruby | 31   |
| #3 制御構造;配列とその利用           | Ruby | 29   |
| #4 手続きと抽象化;再帰呼び出し         | Ruby | 24   |
| #5 2次元配列;レコード;画像          | Ruby | 21   |
| #6 画像の生成 (総合演習)           | Ruby | 21   |
| プレゼンテーション (1)             | Ruby | 21   |
| #7 整列アルゴリズム;計算量           | Ruby | 11   |
| #8 計算量 (2);乱数とランダム性       | Ruby | 7    |
| #9 オブジェクト指向               | Ruby | 7    |
| #10 動的データ構造;情報隠蔽          | Ruby | 7    |
| #11 C言語入門; $f(x)=0$ の求解   | C    | 15   |
| #12 さまざまな型;動的計画法          | C    | 16   |
| #13 文字列の操作;2次元配列 (2)      | C    | 14   |
| #14 構造体;動的データ構造 (2)       | C    | 11   |
| #15 チームによるソフトウェア開発 (総合演習) | C    | 14   |
| プレゼンテーション (2)             | C    | 12   |

の機会を用意した。

高校生は、再帰呼び出しを使ってフラクタル図形を生成したものや、基本の図形を使い美しい絵を作成したものなど、各自のレベルにあった絵を作成した (図-3)。

第 2 回スクーリングには、プレゼンテーションを実施した高校生だけでなく、#6 の課題まで提出できなかった高校生も出席していた。そこで、これまでの学習内容 (#1 ~ #6) の補足説明や、#10 から始まる C 言語の学習方法についての説明を行った。#7 ~ #10 はやや高度であるので、難しい場合は、飛ばして C 言語に進む学習方法もあるとアドバイスを行ったため、そのように学習を進めた高校生もいた。#15 の総合演習の課題を提出した高校生は 14 名、プレゼンテーションを行った高校生は 12 名であった (全参加生徒の 1/3 程度)。

また、全 15 回の学習のうち、最初の 3 回は入門部分でやさしい内容であったため、30 名程度が課



図-2 プレゼンテーションの様子

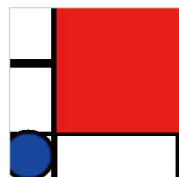
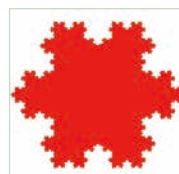


図-3 生徒の作成した美しい絵

題を提出した。しかし、全参加生徒の 2/3 であり、最初から自分で学習し課題を提出することが難しい生徒も一定数いたことが分かる。学習時に分からないことがあれば LMS 上のフォーラム（掲示板）で質問を受け付けていたが、ほぼ利用されていなかった。利用されなかった理由として、LMS のフォーラムはほかの人に内容を見られているので、投稿ができなかった、どんなことを質問してよいのか分からないなど、投稿したときの他校の高校生や、大学の教員やティーチングアシスタントの反応が気になり投稿を躊躇したことも挙げられる。

第 3 回スクーリングに出席した高校生にアンケートを実施した (21 名中 19 名回答) (表-3)。e-ラーニングでの受講はおおむね好意的に捉えられていた。また、初年次のテキストと課題をそのまま使用したが、テキストの内容は難しく感じるものの、知識を広げることができるなど興味を持って取り組むことができたことが分かる。複数の課題を用意し、提出する課題の選択、課題数を高校生に任せることにより、自分に適切な難易度の課題に興味を持って取り組んでいたことも分かった。

また、友だちや後輩にこの講座を勧めるかとの問いに、19 名中 13 名が勧めると回答した。勧める理由として次の回答があった。

- 基礎から順に行っていくので、初歩からしっかり固めていけると感じたため
- 中等教育のうちに、触れることのない経験できないことができるので、ぜひ受講したほうがよいと思う
- 情報系に興味のある方はとてもためになることが多く学べるから
- インターネットや本だけでは難しい学習を映像授

業という形で受けられるため

- プログラミングに興味のある人なら、難しくても楽しめるかなと思う

プログラミングに興味のある、進んだ内容を学びたい高校生に活用してもらうことを目標にしている「高大連携・基礎プログラミング」の運営方法もおおむね適切であったと考える。

勧めないと回答した理由としては、学習を進める上で、分からないことを尋ねる相手がおらず、一人で取り組むことが難しかったことを挙げていた。今後、フォーラムを活用しやすくするなど、分からないことがあるときのサポート体制の充実も必要であると考えている。

## 2020 年度の実施

2019 年度は「高大連携・基礎プログラミング」の正式実施初年度であったが、全参加生徒の 1/3 程度の高校生が最終 (#15) まで課題を提出することができ、単位取得水準まで到達するという、良い結果が得られた。2020 年度は、フォーラムを活用してもらえるよう高校生とティーチングアシスタントのアイスブレイクを行うなど細かい改良をしつつ、進めている。

(2020 年 8 月 3 日受付)

赤澤紀子 (正会員) akazawa@uec.ac.jp

電気通信大学共通教育部特任准教授。博士 (工学)。1997 年電気通信大学電気通信学研究所博士前期課程修了。同年日本電気 (株) 入社、ソフトウェア開発等に従事。2015 年電気通信大学情報理工学研究所博士後期課程修了。プログラミング教育、情報教育に興味を持つ。

久野 靖 (正会員) y-kuno@uec.ac.jp

1984 年東京工業大学理工学研究所情報科学専攻博士後期課程単位取得退学。同大学助手、筑波大学講師、助教授、教授を経て現在、電気通信大大学院情報理工学研究所、筑波大学名誉教授。理学博士。プログラミング言語、プログラミング教育、情報教育に興味を持つ。

表-3 講座のテキスト、課題等に関する調査

|                                | 当てはまる      | やや当てはまる   | 分からない     | やや当てはまらない | 当てはまらない |
|--------------------------------|------------|-----------|-----------|-----------|---------|
| この講座が e-ラーニングでの学習でよかった。        | 7 (36.8%)  | 9 (47.4%) | 3 (15.8%) | 0         | 0       |
| テキストの内容は難しかった。                 | 12 (63.2%) | 6 (31.6%) | 0         | 1 (5.3%)  | 0       |
| テキストの内容は知識を広げることができるなど、面白く感じた。 | 13 (68.4%) | 6 (31.6%) | 0         | 0         | 0       |
| 自分の取り組みたい難易度の課題を選ぶことができた。      | 11 (57.9%) | 6 (31.6%) | 1 (5.3%)  | 1 (5.3%)  | 0       |
| 取り組んだ課題の内容は面白かった。              | 15 (78.9%) | 4 (21.1%) | 0         | 0         | 0       |

