

Vol. 99

CONTENTS

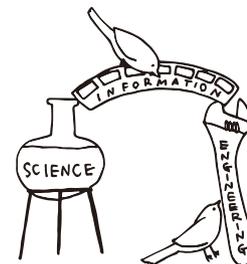
【コラム】科学と工学のはざままで情報を考える… 田中 淳裕

【解説】初学者向けプログラミングの授業におけるソーシャルな知のデザイン（第1回）… 斎藤 俊則

【解説】大学と高校教科「情報」のかかわり… 永松礼夫

COLUMN

科学と工学のはざままで情報を考える



今から 30 年ほど前に私が情報工学を専攻したのは、科学と工学の境目にある領域を学ぶことで、何か新しいことができるのではないだろうか？と漠然と考えていたからです。その鍵が情報にある！と考えたのは今思うと若干短絡的な思考だったかもしれませんが。

そんな古いことを思い出した理由は、本会の前回全国大会（第 81 回）で開催された中高生ポスターセッションの内容を知ったからでしょうか。発表の一覧は全国大会の Web ページ¹⁾に、そして受賞テーマに関しては本誌 8 月号²⁾に解説が述べられています。発表テーマの中の、人がかかわる社会問題を対象とするもの、実践を繰り返しながら解決手段をブラッシュアップするもの、何らかの自律的な動作を実現するシステムなどが、私が個人的に興味をひかれるテーマになります。

科学は真理を追究するもの、一方で工学は現実の課題解決をスマートに行うものと私は認識しています。ところで、高校までの教育課程で、科学（多くは数学・物理・化学など）と工学との関係はきちんと教えられているのでしょうか？工業高校、高等専門学校、工学科を持つ一部の高校を除き、多くの進学校では工学を改めて学ぶ機会はないように思えます。このような進学校でも、情報科目が大学入試に追加されたことをきっかけに、情報という視点で科学と工学との関係を学ぶ機会が必然的に増えてくるのではないかと考えています。

プログラムを書いて何かを認識したり、何かを制御したりするのも情報の 1 つの在り方かもしれません。しかしながら、情報が最も効果を発揮するのは、科学的な知識を背景に、現実の課題解決を情報という手段で提供する場合ではないでしょうか。一例としては、人々の健康問題を解決したり、高齢者の暮らし方を改善したりする手段の提供が挙げられます。新しく情報を学ぶ中高生には、現実社会の中に新しい情報の流れを作り出すことで、個人の行動や集団の振舞いをより良くすることを目指し、情報の成果を具体化させてほしいと思っています。

実は、前回のポスターセッションは個人的な都合で残念ながら参加できませんでした。次回は社会課題の解決につながる「情報」という視点で、ぜひ多くのジュニア会員の発表を聞き、直接対話してみたいと思っています。

参考文献

- 1) 情報処理学会 第 81 回全国大会 中高生ポスターセッション, https://www.ipsj.or.jp/event/taikai/81/ipsj_web2019/html/event/B-6.html
- 2) 大山 裕：中高生ポスターセッションの報告—受賞テーマ研究—, 情報処理, Vol.60, No.8, pp.766-769 (Aug. 2019).

田中 淳裕 (NEC)

初学者向けプログラミングの授業における ソーシャルな知のデザイン (第1回)

齋藤俊則

星槎大学

初学者向けプログラミング授業の課題

初学者向けプログラミング授業^{☆1}では、近い将来に学習者がプログラミングにかかわることが自明である場合を除き、学習者自身における、プログラミングを学ぶ理由の理解やプログラミングにかかわることへの納得感の形成がしばしば教授者にとっての課題となる。本稿では、そのような課題のある大学情報系科目を想定して、学習者が「プログラミングを学ぶ理由」や「プログラミングにかかわることへの納得感」を自ら見出させることに重きを置いた、初学者向けプログラミング授業の作り方を2回に分けて解説する。今回は、この課題の解決に力点を置かなければ、プログラミング授業は「デジタル・コンピテンシー」の獲得を学習目標とすべきであること、そしてプログラミング授業においてデジタル・コンピテンシーの獲得を学習目標とするならば、教授者がプログラミングを「〈ソーシャルな知〉のデザインへの参画」の主題のもとで解釈し授業を編成することが重要であることを説明する。

デジタル・コンピテンシー

情報系や理工系の専門科目への接続が想定されない情報系科目における、初学者向けプログラミング授業で優先すべき学習目標は、コンピューティングの専門家ないしは高度な利用者となるための基礎能

☆1 本稿では科目全体でプログラミングを扱う場合だけでなく、科目の学習内容の一部にプログラミングを組み込む場合を念頭に置き、双方を合わせて「プログラミング授業」と呼ぶ。

力の獲得よりも、まずデジタル技術一般の利用者としての汎用的な素養の獲得であらうと考えられる。そのような素養をここで大掴みにデジタル・コンピテンシーと呼びその詳細を検討する。

コンピテンシーとは、たとえば自らが属するグループの問題解決など、社会的な文脈の中で有能に振る舞うことのできる人の力量を包括的に表す概念である¹⁾。コンピテンシーには特定分野の知識やスキルのような顕在化されやすい力量のほかに、力量を発揮することの意欲や動機などの潜在的かつ心理的要因^{☆2}や、いわゆるソーシャルスキルのような社会的場面での力量などが含まれる²⁾。コンピテンシー概念は従来の客観化、専門分化されたスキル観や能力観ではカバーしきれない、社会的文脈を前提とする包括的な能力モデルとして、複雑さを増す21世紀の社会の教育課題を展望する議論の中で主流となっている(たとえば文献3)、4)。

これを受けてデジタル・コンピテンシーとは、デジタル技術の社会的な応用場面で有能な人が持つであろう顕在的、潜在的、社会的な力量と定義することができる。そのような応用場面としては、現在の具体的な社会課題の解決に結びついた多様な場面が想定される(表-1に社会課題のデジタル技術による解決可能性について授業中に議論を行うことを想定したディスカッショントピックの例を示した)。ただし課題解決への参画や貢献には、その人自身がデジタル技術を実装することに限らず、課

☆2 前者の顕在化されやすい力量を認知的スキルと呼ぶのに対して、力量の発揮にかかわる潜在的・心理的要因を非認知的スキルと呼ぶことがある。

題の当事者としてデジタル技術の専門家と効果的に協働することなどを含む。この意味において、デジタル・コンピテンシーの獲得は、近い将来、デジタル技術の専門家に限らずさまざまな立場から社会課題に参画し貢献することを求められる一般の学習者にとって(こそ)、優先されるべき学習課題であると考えられる。

〈ソーシャルな知〉のデザイン

デジタル・コンピテンシーの獲得を学習目標とするプログラミング授業を作るためには、教授者がまずデジタル・コンピテンシーとしてのプログラミングの在り方を深く理解する必要がある。専門家としてコンピューティングに深くかかわる教授者の多くにとっては、プログラミングはそれ自体が知的かつ創造的な探求の対象であり、学習すべき価値を見出す上でほかに理由は必要ないと感じられるかもしれない。しかし、プログラミングをデジタル・コンピテンシーとして位置づけるためには、社会的文脈におけるプログラミングの意味や価値について今一步踏み込んだ理解が必要である。この理解を深める上で「デジタル・コンピテンシーとしてのプログラミングを学ぶことは、すなわち〈ソーシャルな知〉のデザインを学ぶことである」という考え方が示唆を与える。

〈ソーシャルな知〉とは、ひとことで言えばソーシャルな、すなわち社会的な関係の中にある状況のもとで、その意味や価値が問われる知である。さら

表-1 社会課題のデジタル技術による解決可能性を議論するためのディスカッショントピックの例

社会課題の分類	ディスカッショントピックの例	デジタル技術による解決可能性の示唆
高齢化、地域間の経済格差	地方村落における農業生産従事者の減少と事業持続の困難をどのように解決するか？	生産過程の自動化や省力化、各種気象データの収集と活用、新たな流通販売経路の構築 etc.
社会参画の多様性拡大	障害者雇用と社会参画支援のための業務プロセスの改善をどのように行うか？	業務内容と障害の程度に応じた業務支援のシステム化、心身機能を補助するデバイスの開発 etc.

に、その知の活用が従来なかった新しい社会的な関係を生み出すことを含めて〈ソーシャルな知〉と呼ぶことができる。この考え方の背景には、社会とは人が為すあらゆる事柄（行為であれコミュニケーションであれ）に付随する意味や価値の源泉であり、社会の側もまた人が意味や価値を見出すあらゆる事柄を媒介にして編成され維持されるという認識がある。このような意味や価値と社会との関係についての認識は、学問分野でいえば社会情報学、さらにはその源泉の1つである社会学に由来する。ある社会情報学の文献は、社会学の古典である Max Weber による行為と社会との関係の解釈に触れつつ「行為とは、主体が主観的な意味に従い意味ある行動を取ることであり、また他者がその行為によって方向付けられることで成立するのが社会なのである。よって単なる生理学的な反応だけでは行為とはいえない。主観的な意味とは行為の動機であり、行為の動機を「理解する」ことが大切になる」と説明する⁵⁾。すなわち、社会集団やコミュニティにとって重要な意味を持つ課題解決への取り組みとは「主観的な意味」に動機付けられた「意味ある行動」の集積にほかならず、その中で生み出される知とは〈ソーシャルな知〉そのものであるといえよう。

デジタル・コンピテンシーとしてのプログラミングとは、社会集団やコミュニティの中で生ずる「主観的な意味」に動機付けられた「意味ある行動」であり、特に課題解決に必要な知識の創造や実現に与る点で、それは〈ソーシャルな知〉をデザインすることである。社会課題に直面する人がプログラミングを知ること、仮にコーディングをより高度な技量を持つ他者にゆだねる場合であっても、課題解決に向けてデジタル技術をより能動的に活用する道を開き、課題解決の当事者として、まだ存在しない解決方法の創造に主体的に参画することを可能にする。しかしこの可能性が現実のものとなるためには、社会課題に直面する当事者がプログラミングを知るにとどまらず、プログラミングを通して課題解決のた



めの知をデザインすることに対して、「行為の動機」としての「主観的な意味」を見出す必要がある。それ故に、デジタル・コンピテンシーとしてのプログラミングを学ぶことは、動機となる「主観的な意味」の発見を含めて〈ソーシャルな知〉のデザインを学ぶことなのである。

■ プログラミング授業における 〈ソーシャルな知〉のデザインの導入

プログラミング授業において教授者が〈ソーシャルな知〉のデザインの主題を意識することは、プログラミングを学ぶ理由の理解やプログラミングにかかわることへの納得感の形成に課題がある学習者との間に対話的な関係を開くための準備となる。そもそも、プログラミングを介した研究や職務の遂行が当然のことである教授者の多くは、すでに〈ソーシャルな知〉のデザインに深く参画し、新たな価値の創造に携わっている。この状況に長くある人は、そのような活動に参画することの成り立ちにくさを振り返ったり、活動の中心にある「プログラミングをすること」の意味や価値をあえて再考する機会は必ずしも多くはないのではないかと推察される。他方、学習の理由や納得感を見出せない学習者たちは、プログラミングが社会に不可欠な価値を生み出す活動であり、意味のある営為であることを、漠然とではなく具体的に理解するために、より多くの支援を必要とする。加えて、そのような学習者たちにとって、社会的な意味や価値の創造に与る〈ソーシャルな知〉のデザインへの参加の可能性が自らにも開かれていることを理解するには、さらに多くの支援が必要である。

したがって、上述のような学習者を対象とするプログラミング授業では、ただプログラミングを学ばせるだけではなく、「プログラミングを通して〈ソーシャルな知〉のデザインへの参画を促すこと」を正面から主題に掲げて授業を構成することが、学習者における学ぶ「理由」や「納得感」の発見に対する支援と

しての意味を持ち得る。なぜなら、〈ソーシャルな知〉のデザインへの参画を問うことは、その先にあるプログラミングの意味や価値への問いに必然性を与えるからである。これらの問いは翻って、学習者自身のプログラミングを学ぶ理由や納得感の発見を助けることになる。

「〈ソーシャルな知〉のデザインへの参画を促すこと」を授業の主題とすることは、すなわち、授業の事前準備の段階において、授業を成立させるすべての道具立て(たとえば、科目全体の目的と目標、個々の授業回の目的と目標、学習者に対する支援方針、教材その他の学習リソース、学生に課す課題や演習、学習成果の評価基準など)を、「学生に対して〈ソーシャルな知〉のデザインへの参画を促しているかどうか」という観点より吟味することから始まる(表-2参照)。そして、同じ観点で授業実践の評価や改善を繰り返すことによってそれは実現する(具体的な実践事例は次回の記事で紹介する)。学習者の支援の観点から特に留意すべきは、学習者に対して〈ソーシャルな知〉のデザインへの参画に向けた学習のコンテキスト、すなわち学習の開始から収束(たとえば最終成果物の提出や発表など)に向けた学習活動の脈絡、を作るための配慮である。明確な学習のコンテキストの存在は、特に、現在学んでいる事柄が目的とするゴールに向けてどのような意味や価値を持つのかを学習者自身が見出しやすくするために必要である。

プログラミング授業におけるこのような学習のコンテキストは、プログラミング授業を実施する科目全体、さらには科目体系全体の形作る学習のコンテキストと呼応することが望ましい。いわゆるカリキュラム・ポリシーの制定から始まるカリキュラム・マネジメントの話題は本稿の扱う範囲を超えるが、カリキュラム全体が学生に対して何らかの形で〈ソーシャルな知〉のデザインへの参画を促すものであれば、プログラミング授業の組み立ては、科目体系全体の作り出す学習のコンテキストへの呼応や調

和を意識することでより効果的となる。科目体系全体において〈ソーシャルな知〉のデザインへの参画に向けた学習のコンテクストを作ることができるならば、学習者がデジタル・コンピテンシーとしてプログラミングを学ぶことの意味や価値を知り、そこに自分なりの理由や納得感を見出すことはより容易となるであろう。

このような授業では学習者の何を評価すべきか

〈ソーシャルな知〉のデザインを主題とするプログラミング授業では、学習者の評価はプログラミング技術の獲得のみならず、プログラミングを含めたデジタル技術を活用して知の生産者として社会へ参画しようとする姿勢や、そのような社会参画の蓋然性を高める学習者としての素養の成長を総合的に評価することが重要である。しかし、そのような姿勢や素養の成長は客観的な評価材料のみでは確認し

づらいため、いわゆる形成的評価（学習の途上にある学習者に介入しつつ進捗や乗り越えるべき課題に関する質的なフィードバックを交えながら行う評価）を適宜用いながら総合的に成長を跡付けることが求められる。

その際に、特に注意深く観察すべき点の1つは、学習者による、プログラミングを学ぶことの意味や価値への言及の内容とその変化である。こと、プログラミングを学ぶ理由や納得感に課題がある学習者は、学習を開始する当初は、プログラミングを学ぶことの意味や価値を語る語彙をほとんど持たないか、非常に限定的であることが通常である。その語彙の変化（たとえば学習の意味や価値を語る語彙が豊かになった、内容が具体的になった、学習者の置かれた文脈との関係が表れてきた、など）を観察することで、学習者における〈ソーシャルな知〉のデザインの主題のもとでのプログラミングの学習の進捗や深化が理解できる。

表-2 「〈ソーシャルな知〉のデザインへの参画」の主題を導入したプログラミング授業の企画例（抜粋）

<p>授業の目的と目標</p> <p>(1) 目的 すべての学習者がデジタル技術による課題解決の当事者になり得ることを認識した上でプログラミングの基礎を学び、かつそれぞれの立場においてプログラミングを学ぶ意味や価値を発見できること</p> <p>(2) 目標 学習者が自分の設定した小さな課題解決に役に立つプログラムを作成し、その性能や改善点を評価できること 学習者が社会課題解決の事例に基づいてプログラミング技術を学習することの意味や価値を他者に説明できること</p>
<p>授業進行</p> <p>第1, 2回に具体的な課題解決へのプログラミングの利用事例を取り上げてディスカッションを行う。第3回に学習者に小さな課題解決のプロジェクトの立案を促す。その後の回で基礎的なプログラミング技術の学習に取り組み、立案した課題解決に役立つプログラムの設計、実装、評価を経験させる。最終回に成果発表を設けプロジェクトの成果の振り返りと、実際のプログラム開発経験を踏まえた課題解決事例(最初の2回で取り上げたもの)に対する評価を発表させる</p>
<p>今年度取り上げる課題解決の事例</p> <p>(1) 課題の内容 気候変動によって亜熱帯化が進む島嶼部での伝染病の発生</p> <p>(2) デジタル技術（特にプログラミング）を使った課題解決の可能性 ドローン、リモートセンシング、画像解析技術の組合せによる島内の水場の位置の特定とその変化の観測</p>
<p>上記事例を手掛かりにした授業展開のための問いの例</p> <p>(1) ディスカッション 課題解決事例においては、どのような場面でどのような「知」が創出されるか？ その「知」は誰にとってどのような意味や価値を持ち得ると考えられるか？ 当事者が持つデジタル・コンピテンシーは上記事例の解決にどのような影響を与えると考えられるか？</p> <p>(2) 課題解決の企画立案： このような社会・コミュニティの課題解決にプログラミングが用いられる例はほかにないか？ 自分（たち）がかかわる社会やコミュニティにおける課題解決にプログラミングを応用できる例はないか？ (特に限られた予算、時間、技術レベル等を想定した) 小規模の実行可能な課題解決としてどのような事例が考えられるか？</p> <p>(3) 技術的な課題の明確化と学習目標化： 選んだ課題の中で特にプログラミングが役立つような部分はどこか？ 授業で扱う基本的な技術の組合せを課題解決のどの部分にどのように応用できそうか？</p>



また、学習者同士の協働性が問われる課題を置くことで、それぞれの学習者のプログラミングへのかかわりの自律性や、それを〈ソーシャルな知〉としてグループのために行使することの積極性を観察することができる。この観点による観察においては、教授者は学習開始時点におけるプログラミングの技術や知識の個人差を念頭に置きつつ、適宜助言やフィードバックを交えながら、特に技術や知識の面で不安がある学習者が、課題解決において積極的な役割を果たせるように支援することが重要となる。そして、協働の過程で、技術や知識の面で異なるスタートラインに立つそれぞれの学習者が、〈ソーシャルな知〉のデザインへの参画を通じたデジタル・コンピテンシーの発揮の面で見せた成長を積極的に評価することが求められる。

次回に向けて

本稿ではプログラミングを学ぶ理由の理解やプログラミングにかかわることへの納得感の形成に課題のある大学情報系科目を想定して、学習目標としてのデジタル・コンピテンシーの獲得と、授業の主題としての〈ソーシャルな知〉のデザインへの参画を手掛かりに、学習者が「プログラミングを学ぶ理由」

や「プログラミングにかかわることへの納得感」を自ら見出させることに重きを置いたプログラミング授業の作り方の骨子を解説した。今回は筆者がかかわってきた実践の成果を例に、〈ソーシャルな知〉のデザインへの参画の主題を取り入れたプログラミング授業のもとで、学習者がプログラミングの学習にどのように動機付けられたのかを具体的に解説する。

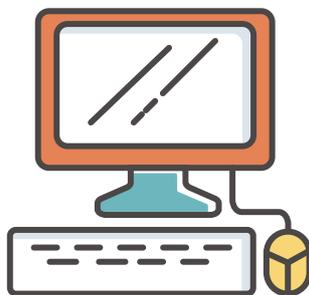
参考文献

- 1) Woodruffe, C. : What is Meant by a Competency?, Leadership & Organization Development Journal, 14(1), pp.29-36 (1993).
- 2) Magenheimer, J., Nelles, W., Rhode, T. and Schaper, N. : Towards a Methodical Approach for an Empirically Proved Competency Model, In Teaching Fundamentals Concepts of Informatics : 4th International Conference on Informatics in Secondary Schools - Evolution and Perspectives Proceedings, pp.124-135 (2010).
- 3) 文部科学省, OECDにおける「キー・コンピテンシー」について, http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/039/siryu/attach/1402980.htm
- 4) OECD, OECD Future of Education and Skills 2030 - Organisation for Economic Co-operation and Development, <http://www.oecd.org/education/2030-project/>
- 5) Scardamalia, M., Bransford, J., Kozma, R. and Quellmalz, E. : New Assessments and Environments for Knowledge Building, In Griffin, P., McGaw, B. and Care, E. (Eds.) : Assessment and Teaching of 21st Century Skills. Dordrecht, Springer, pp.231-300 (2012).
- 6) 石井和平 : 社会情報学—情報技術と社会の共変—, 学術出版会, p.20 (2007).

(2019年9月2日受付)

齋藤俊則 (正会員) t-saito@gred.seisa.ac.jp

星槎大学大学院教育実践研究科准教授。本会誌編集委員会専門委員会(教育分野/ EWG) 幹事。本会 IFIP 委員会 TC3 (教育) 代表。WCCE 2021 開催準備委員会委員長として同会議の広島開催の準備に取り組む。



大学と高校教科「情報」のかかわり

永松礼夫

神奈川大学

高校教科「情報」は元気に育っているか？

情報処理とその教育に関心を持つ大学人としての雑感を書かせていただく。高等学校の教科として「情報」が登場してから久しいが、新参者の教科として何となく肩身の狭い印象はいまだに否めない。大昔からある英語や数学や理科(物理・化学・生物)と比べてみると、教育の根幹をなすような堂々とした基本的教科になれたとは言いにくい。大学受験科目として広範に採用されてはいないので、受験指導を通して真剣に指導される機会が少ないことも影響しているのだろう。また「情報科学・情報工学・ICT」そのものが分野として歴史が浅く日々の進歩が激しいため、受験勉強をしようにも定番の問題集や参考書が存在しないこともある。さらに、パソコン教室のような文書作成や表計算のスキルを教えるだけの実技科目であるとみなされてしまうなど、わざわざ教科として教える価値のある深みを持つ理論・知識の体系がないようにも思われている。その結果、現代社会を生きる上で重要な、高校卒業までにしっかり身に付けておくべき知識を扱っている科目であるという意識が薄く、学校運営の中で何かと低い優先度で扱われているようにも見える。

教科「情報」がもっと「元気な」教科になってほしいと思いつけている。数年前から「情報入試研究会」や「文部科学省・大学入学者選抜改革推進委託事業(情報分野)「情報学的アプローチによる『情報科』大学入学者選抜における評価手法の研究開発」(2016～2018年度)¹⁾にかかわっている。また、高大接続に関する文部科学省の会議の傍聴に行ったり、「大学教育の分野別質保証のための教育課程編成上の参照基準：情報学分野」²⁾の策定の過程を関連委

員会のメンバとしてウォッチしてきた。

我が国の高等教育の特徴として、多くの生徒が受験して大学に入るまでにエネルギーを費やし、入学後は真剣に学ばない風潮があることは事実であろうし、それをまず改革するのが本筋であろう。また、近年では「上位校」ではない大学では推薦入試など、「受験」を要しない選抜で入学する者が半数近くになっている。高校全体の教育を活性化するゴールに対しては、国立で5科目、私立でも3科目以上を課するような大学入試の比率が低下し、入試改革の教育改革への影響は限定的になっている³⁾。

しかし、「情報を元気な科目にする」ための1つの手段であると割り切って、情報入試(大学の入学試験で教科「情報」を取り入れること)をここ数年來は応援してきた。最近(と言ってもこの1年ほど)は少し流れが変わり、首相直下の会議などでIT立国とそれを支える教育の重要さが強調され、デジタル人材・AI人材の語を頻繁に目にするようになった。折しも、次の指導要領策定や大学入試センター試験の改革のタイミングに合わさったこととも呼応して、センター試験への導入や小学校へのプログラミング教育の取り組みなど、教科「情報」を直接応援するような流れが加速している。

本稿ではこういった経験をもとに、個人的な見解も交えて、高大連携や入試や初年次教育の話題を主に、高校と大学の「情報」教育に関してミスマッチを感じる個所や課題とを感じる個所について述べてみたい。

「～的」思考

知識とは別に、思考力、とりわけ「情報的な」考え



方を育むことが強調されている。本稿の読者の多くは、「CT—Computational Thinking」⁴⁾が頭にかぶであろう。一語で内容が正しく伝わる良い日本語訳がなく、中島先生は「計算論的思考」を充てている。プログラミング教育が強調される流れの中で「プログラミング的思考」という言葉もよく使われる。Wingの元論文を見ると、「計算機科学者のように問題解決する」思考法となっている。その流れで、ある問題を、計算する機械や情報を処理するエージェントを用いて解くためにどのような方法を選択するかを決める際の考え方として、問題の分割・階層化・手順化などを強調している。今後のコンピュータに基づく情報処理とそれを応用した社会システムの構築にも役立つような、「コンピュータ・サイエンス」における「科学的方法」(客観的事実に基づく考察や論理的思考に基づいた判断)と言えよう。

教育の文脈では少し違ってくるようで、イギリスの初等教育へのガイドライン (Computing At School (CAS) による⁵⁾) では、ポイントとして、(1) think algorithmically ; (2) think in terms of decomposition ; (3) think in generalizations, identifying and making use of patterns ; (4) think in abstractions, choosing good representations ; (5) think in terms of evaluation を挙げている。あえて英語のままで表記したが、これらを輸入して日本の教育を論じる際に、日本語への置き換えが十分でなく、意味が微妙に違って議論されていることもあるので危惧を感じている。(3)の後半は、ソフトウェア工学での「デザイン・パターン」を意識した記述と思われるが、単に「パターン認識」と紹介された例がある。日本語では同じ「認識」と片づけてしまいがちだが、recognitionは「存在を認知する、あるいは、知られた分類のどれに該当するかを判断する」、identifyは「同じ種類・類型のものであるかを判断する」、discriminationは「ある基準に照らして該当するか否かを判断する」であり、それぞれ違った概念に結びついている。カリキュラムの設計やシラバス(講義要目)の記載を扱うときもそうであるが、ま

だその分野を学習していない人が当該分野の知識の詳細を示しているキーワードを議論・評価することには、困難が伴うことを自覚しておきたい。

「小学校段階におけるプログラミング教育の在り方について(議論の取りまとめ)」⁶⁾の中では、基本命令を組み合わせてプログラムを作成する活動を意識して「プログラミング的思考」を「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、一つひとつの動きに対応した記号を、どのように組み合わせるか、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」としている。これを見ると、「Computational Thinking」が情報科学的な問題解決の方法論を広くカバーするものであったのに、「プログラミング的思考」では学校でできる範囲のプログラミング実践に限定された枠組みに寄っているようだ。

設計と「デザイン」

情報科学・情報工学の立場では「デザイン」の語は、「設計(何かを作る際に前もって計画し、そこで定めたことを記録に残し、それに従って作ること)」と「意匠(色や形や見た目)」の2つの意味で使われる。やや極端な例だが、意匠は本来の機能(電気のスイッチなら、人による操作と回路のON/OFFを対応させること)とは別の製品の属性(部材の形や硬さや表示の分かりやすさ)を工夫して、本来の機能をより使いやすくすることを考えることであり、「ユニバーサルデザイン」の例はこれに当たる。指導要領に「情報デザイン」の項目があるが、意匠の側面として捉えている。

デザインの「設計」の側面はまだまだ扱いが不十分に思える。たとえば、スタンフォード大学のd.schoolの教育は、「ものづくり」をチームでの議論を通して設計から制作・実用化まで行う活動を実践する場であり、初期の段階から意匠も含めて総合的に計画することを強調している。最近の「ものづくり」ではアップル社の製品のように、形や動きに

よって初めて成立する機能もあるので、設計と意匠を総合的に扱う流れは自然である。

「情報」の教育では、設計の側面はあまり強調されていないので、プログラミング言語を習って少し経った学生が、やや大きめの作品を作ろうとすると、やみくもにプログラムを入力する作業から始めてしまう。教育者としては「いきなりコードを書くな！まず設計図を書け」と言いたい。感覚的なものでなく、事実と論理的思考に基づいた「科学的方法」に則ったシステムやプログラムの作成が求められている。情報システムやITを専門とする道に進むならば、自分以外の人が使ったり・保守したりすることを想定した「お客様にお出しできる」プログラムを作ることがゴールである自覚を持ってほしいものだ。

入試について

入試改革の文脈、特に大学入学共通テスト（新テスト）に関して、コンピュータ化された試験の導入と思考力を測る試験の導入の両者は一緒に語られることが多いが別のものである。

□ コンピュータで行う試験

知識を問うようなCBT（コンピュータ化された試験）は資格試験等ではすでに広範に実施されている。紙で行っていたことを電子化（キーボードとディスプレイ）するだけにとどまらない。「適応型テスト」では、試験の進行を積極的に制御して試験時間中の受験者の応答の内容（どんな解答をしたか）に応じて以降の出題が変化するようにしたり、解答順序を強制して特定の問題は一度答えたら「戻れない」ようにすることができたりする。たとえば、語学などでレベル判定を行いたい場合、それまでの正答率が高ければより難しい問題を・低ければよりやさしい問題を出題することで、少ない設問数で受験者のレベルをより効果的に絞り込める。医療系の試験などでは、「戻れない」フローを使い、1問目では症状から病名を答えさせ、2問目では病名を明らかにした上で治

療法を解答させるといった問題を出題している。

試験を行う側では、大量の問題ストックを用意する必要があり、問題の作成はもちろん、作成した問題の質の管理（特に難易度を揃えること）が難しい。難易度の揃った類題を効率的に（可能ならば自動的に）数多く作成できる方法があるかが課題になっている。

□ 思考力を測る入試

知識を問う問題よりも一問あたりの解答時間がかかると予想されるので、限られた試験時間の中にかに収めるかが課題となる。複数の種類の思考力を別々に測ろうとすればなおさらである。

思考力を測る試験として、記述式が良いとされているが、限られた試験時間・採点時間の中で処理できるように配慮すると問題の作り方が制限されてしまう。解答欄として用意するマス目の数を減らして文字数に強い制限をかける、構成（二文を逆接の接続詞でつなぐ、など）や使用するフレーズ（問題文で示した傍線部の単語を使う、など）に制約をつけて採点を容易にする、などである。このような制限をすると副作用として、あまり思考力を使わなくても、定型的な解答方法のコツに従えば点が取れるような問題になってしまい、思考力を測る趣旨から乖離してしまう。また、米国の共通テストでの記述式エッセイについても、内容よりもテクニックで点が取れる、入学後の成績と相関が少ないなどの問題が浮上している。

委託事業¹⁾の一環として文字数を指定したマス目に自由記述させた答案の採点を試みたが、試験を終えて答案を見るまでどんな単語が来るか分からない、辞書にない若者言葉や造語が入る可能性もある。このような状況で単語の選択についても評点するなら、あらかじめ採点基準を定めることはできない。

別のアイデアとして、出題側であらかじめ解答文の構成要素となる単語やフレーズを書いたカード（短冊）を提示しておき、それらの中から使うものを選択して並べ替えることで解答文を作成させる方式を導入



した。この「短冊方式」では、未知の単語問題を回避しつつ、解答パターンは十分多くあるが採点しやすい程度に少ない有限な数に絞れることが分かった。

情報機器を扱うスキル

大学に進むまでに身に付けておいてほしい「情報学」を知識・スキルの側面から考える。大学の初年次セミナーの担当者から、「レポートの作成法」といったタイトルで「レポートの書き方」を教えたいのに「文書作成ソフトの使い方」の教育に終始してしまった、という声を聞く。コンピュータ操作を教えることに煩わされずに、論説文の構成、証拠を示した客観的な議論の進め方、資料の引用の仕方を教えることに専念したいという意味である。

学生が初めて自分用のPCを持つタイミングは変化しているようで、少し前のように大学入学の時点で一斉にPCを買わなくなっている。スマホの高機能化で大概のことは足りるし、高校時代からコンピュータを用いた授業を受けた者も多く、入学前からタブレット端末やPCを所有している者も多くなってきた。

学生が端末を持っていることを前提に、「コンピュータ教室」で行っていたことを、一般教室で行おうとする動きもある。BYOD (Bring Your Own Device) と呼ばれ、従来型のコンピュータ教室の削減と絡めて語られることもある。

授業運営側からみれば、文具としてのPC (学部・学科を問わず、Webとメールへのアクセス・文書作成・資料提示・簡単な表計算などの標準的な知的生産活動の機能)と、教育機材としてのPC (専門科目に固有であったり、大規模な情報を扱ったり、特定の装置と関連するようなもの：実験実習設備に相当)の境界があいまいで、個人端末ですべてのコンピュータ演習が代替できるとか、PCを持つる学生には文書作成は指導不要などの誤解もある。

「必要・共通」な端末スキルの範囲は、教育の電子化の進展と歩調を合わせるべきで、現状は発展途上であることをしっかり自覚することが大事である。情

報機器を用いた勉強法についてコンセンサスがなく、紙媒体のカルチャーから「BYODカルチャー」への移行ができていない^{☆1}。紙媒体での文化の蓄積の上に構築された「〇〇大生の超ノート術」に相当するものの電子デバイス文化版が登場するのはまだ先だとすれば、それを創りだせるような学生が育ってほしい。

未来に向けて

「情報学」のカバーする内容は、情報システムに関する知識・スキルの側面とコンピュータに基づく情報処理の土台となる客観的考察や論理的思考の側面の2つがある。そのため、この分野ではルーチン化した流れに乗るだけでなく、そのルーチン自体を見直して改良につなげる思考法こそが重要であり、我々も実践を怠ってはいけないし、そこが面白いのだと思っている。

参考文献

- 1) 文部科学省・大学入学者選抜改革推進委託事業 (情報分野)：情報学的アプローチによる『情報科』大学入学者選抜における評価手法の研究開発 (2016～2018年度)、http://www.mext.go.jp/a_menu/koutou/senbatsu/1413650.htm
- 2) 浜中淳子：2020年度の入試改革 高校生「学習離れ」防げず、日本日本経済新聞 2019年8月12日付。
- 3) 日本学術会議 情報学委員会 情報科学技術教育分科会 大学教育の分野別質保証のための教育課程編成上の参照基準 情報学分野、2016年3月23日、<http://www.scj.go.jp/ja/info/kohyo/pdf/kohyo-23-h160323-2.pdf>
- 4) 中島秀之：Computational Thinking 計算論的思考、情報処理、Vol.56, No.6, pp.584-587 (May 2015) (Jeannette M. Wing: Computational Thinking, Communications of the ACM, Vol.49, No.3, pp.33-35 (Mar. 2006)の翻訳)。
- 5) CAS, Computational Thinking A Guide for Teachers (Nov. 2015), <https://www.computingschool.org.uk/computationalthinking>
- 6) 小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議 (2016)、小学校段階におけるプログラミング教育の在り方について (議論の取りまとめ)、2016年6月16日、http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm (2019年9月17日受付)

永松礼夫 (正会員) lnag@kanagawa-u.ac.jp
神奈川大学・理学部情報科学科・教授。

^{☆1} 第14回神奈川大学メディア教育シンポジウム「大学教育における必携PC活用を考える」(帝京大学・宮崎誠先生の講演)、2019年6月22日、https://www.kanagawa-u.ac.jp/event/details_18581.html