

Vol. 93

CONTENTS

- 【コラム】数えられることの幸せ—コンピュータ教師の父親に息子がくれたプレゼント—… 斎藤 俊則
- 【解説】高大接続改革とeポートフォリオ—資質・能力を多面的・多角的に評価し育成していくために—… 森本 康彦
- 【解説】プログラミング入門科目の指針と実践例（後編）… 久野 靖

COLUMN

数えられることの幸せ

—コンピュータ教師の父親に息子がくれたプレゼント—



SNS での情報発信は時に事故ともハプニングともつかない展開を生み出します。たとえばある SNS に私が書いた文章は、コメント欄でのやりとりを経てここに公開される運びとなりました。それは 6 歳の息子とのやりとりを記したこんな文章でした。

阿部先生の『わくわくプログラミング』¹⁾ を読んで FizzBuzz を気に入った息子にせがまれて、こども園の行きに 1 から 10 までを数え上げて遊んだら、帰りも続きをせがまれて、30 まで一緒に数え上げることができました。割り算（正確には割り切れるかどうかのチェックのみでしたが）は掛け算ができればよいこと、掛け算は足し算を繰り返せばできること、30 までに Fizz（3 で割り切れる数）の方が Buzz（5 で割り切れる数）よりも多く出てきて、それは 3 の方が 5 よりも小さいからだと思われること、などなど、彼は行き帰りの間に随分いろんなことを発見して面白がっていました。

さて、さらに家に帰ってからピアノの先生のところに向かう道すがら、FizzBuzz で遊ぶ息子にふとした出来心が生じた父親は……

父「実はね、1 と自分以外で割れない数ってのがあってね……」

と、悪魔の囁きのごとく、例の「あいつ」の扉を開くことに。2, 3, 5, 7, 11, 13 まで、一緒に数えてみました（そこで時間切れ）。キラキラした目でこちらを見ながら「数って面白い！」と笑顔の息子に向かって、

父「こういう数のことを、“そすう” っていうんだ」

と言い残し、じゃ、ピアノ頑張るねと彼を先生のもとに置いていきました。悪魔はいろんなところに潜んでいますね（笑）。息子の笑顔を思い出して、これって最高の誕生日プレゼントじゃん、と思った次第です。

このとき息子は前日に誕生日を迎えた父親に、ある発見をプレゼントしてくれました。それは数える力を行使できることはそれ自体が幸せである、ということです。この「幸せ」を共有することは、計算機を教える者として最も大切にすべきことの 1 つであるように思いました。

この書き込みに「できすぎで実話とは思われません（笑）」と即座に反応された久野靖先生（電気通信大学）は、この話を『『一杯のかけそば』みたいに日本中に流行らせたい！』と仰り、このコラムが誕生しました。

参考文献

1) 阿部和広：小学生からはじめるわくわくプログラミング、日経 BP 社 (2013).

斎藤 俊則 (星槎大学)

高大接続改革とeポートフォリオ—資質・能力を多面的・多角的に評価し育成していくために—

森本康彦

東京学芸大学

高大接続改革とは¹⁾

2020年度に実施する2021年度入学者選抜から大学入試が大きく変わる。これに伴い、今、高等学校では、学習指導の方法や評価の在り方などの議論が盛んに行われている。この背景には、戦後最も大きな教育改革と言われる「高大接続改革」がある。これは、単に大学入試の方法を変えるという単純なものではない。高等学校と大学での学びを一貫して、「知識・技能」、「思考力・判断力・表現力」、「主体性を持って多様な人々と協働して学ぶ態度」といった学力の3要素を確実に育成し評価することを目指した、高等学校教育—大学入学者選抜—大学教育の各改革を三位一体で進める大改革である²⁾。

高等学校教育では、カリキュラム・マネジメントを通して、主体的・対話的で深い学び（いわゆる、アクティブ・ラーニング）を充実させ、多面的・多角的に評価することが改革として行われる。

大学教育では、選抜（アドミッション・ポリシー）、教育（カリキュラム・ポリシー）、卒業（ディプロマ・ポリシー）の各段階の一体的な策定を行い、それに基づいた改革が行われる。また、高等学校教育での学びを踏まえ、学生を3つのポリシーに基づいて、学力の3要素で育成されるべき資質・能力をさらに高める必要がある。

そして、大学入学者選抜では、これらの2つの改革を接続する役割があり、大学が定めたアドミッション・ポリシーと高等学校教育での学びの評価を基に選抜されることが改革として行われる。

育成すべき資質・能力の3つの柱

学力の3要素で育成すべき資質・能力の3つの柱として、1) 個別の知識・技能、2) 思考力・判断力・表現力等、3) 学びに向かう力・人間性等³⁾、これらは氷山に例えられる。氷山は、水面から見える部分は小さいが、水面下はその何倍もの大きな塊からなっており、この部分が大きければ大きいほど安定する。資質・能力では、「知識・技能」が水面の上の部分に例えられ、テストで容易に測ることができるものである。実は、この「知識・技能」を実際に学びに活かしていく（活用する）ためには、水面下の大きな塊の部分にあたる資質・能力をしっかりと有していることが不可欠であり、それが「思考力・判断力・表現力等」と「学びに向かう力・人間性等」である。これらは、テストだけでは評価しにくい資質・能力であり、eポートフォリオを用いることで多面的・多角的に評価することが可能となる（図-1）

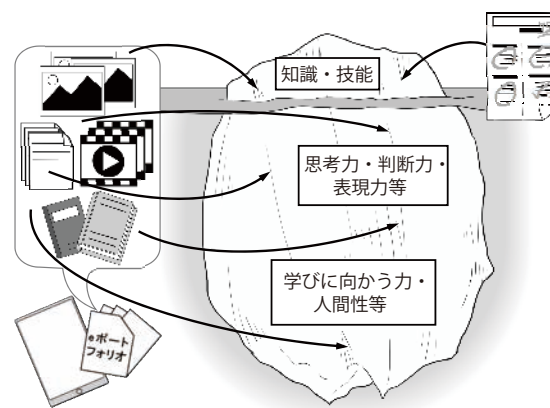


図-1 資質・能力の氷山モデル

eポートフォリオとは

生徒・学生自らが、学習を振り返って次につなげる主体的な学びの過程では、さまざまな学びの記録が生成されるが、これら記録を電子的に蓄積したもののすべてが「eポートフォリオ」である。学習過程において蓄積されたeポートフォリオは、生徒・学生の学びの教材として、それ自体が活用できるだけでなく、継続して蓄積・活用していくことで、過去と現在の学習状況を把握・評価し、未来の伸びしろまで見える化できる「学びのアルバム」となる。特に、生徒・学生は、いつでもどこでもeポートフォリオを記録し、見返して活用することができるため、自問自答による学びの振り返り（自己評価）が促され、仲間同士で時空を超えて相互に学び合ったり（相互評価）、教員などからアドバイスがもらえたりする（教員評価、他者評価）。つまり、eポートフォリオは、アクティブ・ラーニングを行う“必須のツール”であり、学びを促進させるとともに、学力の3要素で育成すべき資質・能力を多面的・多角的に評価することを容易にする。

育成すべき資質・能力とは

中央教育審議会答申では、「教育課程全体を通じてどのような資質・能力の育成を目指すのかは、各学校の学校教育目標等として具体化されることになる」とされている⁴⁾。しかし、各教科等の学習指導要領解説では、育成すべき資質・能力は、文章中のさまざまな箇所に埋め込まれて記述・表現されているため、具体的にそれらを把握することは難しいのが現状である。

そこで、本稿では、平成30年改訂高等学校学習指導要領解説(国語, 地理歴史, 公民, 数学, 理科, 保健体育, 芸術, 外国語, 家庭, 情報, 総合的な探究の時間, 特別活動)から、育成すべき資質・能力に関する記述を抽出・整理することで、具体的な育成すべき資質・能力を明らかにすることを試みた。

各教科等で育成すべき「知識・技能」のほか、「思考力・判断力・表現力等」と「学びに向かう力・人間性等」に関する記述を抽出したところ、全8,782件の項目が得られ、〈思考力〉は2,048件、〈判断力〉は275件、〈表現力〉は579件、〈課題解決〉は937件、〈主体性〉は1,367件、〈協働性〉は1,300件、〈多様性〉は956件、〈豊かな心〉は1,320件分類され、それぞれ表-1, 2の通りに整理された。各表の小カテゴリは、学習指導要領の文言を参考に「力」と「態度」でまとめている。なお、〈豊かな心〉の小カテゴリは、平成29年改訂小学校及び中学校学習指導要領解説「特別の教科 道徳編」を参考に「力」と「態度」を包括した表現としてそのままの形とした。これにより、「生徒に対してどのような資質・能力の育成を目指すのかを指導のねらいとして設定」し³⁾、生徒一人ひとりが「前の学びからどのように成長しているか、よ

表-1 「思考力・判断力・表現力等」に含まれる資質・能力

大枠	中カテゴリ	小カテゴリ
思考力	ひろげる力	考えをひろげる力
		発展的に考える力
		創造する力
	予想する力	予想する力
		推論する力
	まとめる力	考えをまとめる力
		構造化する力
		抽象化する力
	分析する力	順序づける力
		関連づける力
		分類する力
		比較する力
		理由づける力
具体化する力		
とらえる力	多面的・多角的にとらえる力	
	批判的にとらえる力	
判断力	選択する力	選択する力
		判断する力
	決定する力	意思決定する力
表現力	伝達する力・発信する力	文字で表現する力
		言葉で表現する力
図で表現する力		
身体で表現する力		
	プレゼンテーション力	プレゼンテーション力
課題解決	課題設定を行う力	課題を発見する力
		課題を設定する力
	方略を立てる力	方略を立てる力
解決の検証をする力	解決の検証をする力	



り深い学びに向かっているか」を評価し見取る際の参考になると考えられる⁴⁾。

eポートフォリオを活用した資質・能力の評価方法

(1) 資質・能力の多面的・多角的な評価方法

資質・能力の評価については、「単元や題材など内容や時間のまとまりを見通しながら評価の場面や方法を工夫して、学習の過程や成果を評価する」とし、「指導と評価の一体化を図る中で、論述やレポートの作成、発表、グループでの話し合い、作品の制作等といった多様な活動を評価の対象とし、ペーパーテストの結果にとどまらない、多面的・多角的な評価を行っていくこと」とされている³⁾。つまり、資

質・能力を多面的・多角的に評価するために、eポートフォリオの活用が有効的であるということが読み取れる。しかし、生徒の学習活動の、こういったタイミングで、どのeポートフォリオを用いて、どのように評価するかを明らかにする必要がある。

そこで、本稿では、この解決の手立てとして、eポートフォリオを活用した学習評価の方法（以下、アセスメント方法）をまとめた（表-3）。アセスメント方法は、生徒の学習活動の何に焦点を当て、どのeポートフォリオを用いて、資質・能力の育成状況を把握し、どう学習支援するかの方法であり、これらはハンドブックとして公開している⁵⁾。これらのアセスメント法を組み合わせることで、あらゆる学習活動に対して、適切なeポートフォリオを蓄積・活用しながら多面的・多角的な評価を実施していくことが可能になると期待できる。

表-2 「学びに向かう力・人間性等」に含まれる資質・能力

大枠	中カテゴリ	小カテゴリ
学びに向かう力	主体的に学習に取り組む態度	主体的に学習に取り組む態度
		主体的になる態度
		関心を持つ態度
		粘り強く学ぶ態度
	自らの学習を調整する態度	粘り強く取り組む態度
		学習を見通す態度
	キャリアを形成する態度	学びを振り返る態度
		次につなげる態度
		キャリアを形成する態度
		自己実現する態度
人間性等	コミュニケーション力	伝え合う力
		交流する態度
	協働する力	チームワーク力
		リーダーシップ力
		合意形成する力
		よりよい人間関係を形成する態度
	理解する態度	集団・社会に貢献する態度
		自己を理解する態度
		他者を理解する態度
		歴史や伝統、文化を理解する態度
尊重する態度	自己を尊重する態度	
	他者を尊重する態度	
	歴史や伝統、文化を尊重する態度	
	感性	
豊かな心	豊かな心	自主、自立、自由と責任
		節度、節制
		思いやり、感謝
		礼儀
		家族愛、家庭生活の充実
		生命の尊さ
		自然愛護
		感動、畏敬の念
		よりよく生きる喜び
		創造性

表-3 eポートフォリオを活用したアセスメント方法⁵⁾

アセスメント方法	説明
テスト法	テストを受け、テストの解答と解き直しから学習状況を把握する方法。
質問法	質問（アンケート等）の回答から学習状況を把握する方法。
作品法	学習成果としての作品から学習状況を把握する方法。
レポート法	レポートやエッセイ、小論文の記述から学習状況を把握する方法。
日誌法	日々の出来事やその活動、思いの記述などから学習状況を把握する方法。
実技法	実技の記録などから学習状況を把握する方法。
体験法	体験の記録などから学習状況を把握する方法。
プレゼンテーション法	プレゼンテーションの記録などから学習状況を把握する方法。
議論法	議論や討論の過程の記録およびその結果から学習状況を把握する方法。
思考・判断法	思考・認知過程を文章や言語で言語化・図式化することで外化させ、その内容から学習状況を把握する方法。
ノート法	気づきや思考が外化されたノートの記述から学習状況を把握する方法。
演習法	演習した解答等の記述などから学習状況を把握する方法。
実習法	実習の記録から学習状況を把握する方法。
課題解決・探究法	課題解決や探究の遂行過程の記録およびその結果から学習状況を把握する方法。
観察法	学習過程における生徒の活動を観察することで、学習状況を把握する方法。
面談法	教員と生徒（ただし、保護者や第三者が加わることもある）との面談（カンファレンス）から学習状況を把握する方法。
ショーケース法	生徒の特徴的な学びをまとめたショーケース・ポートフォリオを用いて学習状況を把握する方法。

(2) 横断的な学びにおける資質・能力の評価方法

『主体性を持って多様な人々と協働して学ぶ態度』など、筆記試験や技能試験によって評価しにくい資質・能力については、特に日々の高等学校の学習活動等を通じて多面的な評価を行う²⁾、「これからの時代に求められる資質・能力を育むためには、各教科等の学習とともに、教科等横断的な視点に立った学習が重要であり、各教科等における学習の充実のもとより、教科等間のつながりを捉えた学習を進める必要がある⁴⁾」としている。つまり、特に「学びに向かう力・人間性等」の育成とその評価は、横断的な学びの視点が必須であることが分かる。

各教科等では主に「知識・技能」の育成とその評価が重要となるが、たとえば、各教科等の学びの中に、授業の内容を変えずに「協働する力」の「チームワーク力」や「合意形成する力」を養うための学習活動として【学び合い／議論】を定期的かつバランスよく埋め込み、実施していく際に学んだ成果や振り返りの記録をeポートフォリオとして蓄積・活用していく。その後、蓄積されたeポートフォリオを用いて、生徒の学習活動を時系列でつなげて見直すと、一連の学びが浮かび上がる(図-2)。さらに、上手くeポートフォリオにより見える化することで、「生徒一人ひとりの持つよい点や可能性などの多様な側面、進歩の様子などを把握し、学年や学期にわたって生徒がどれだけ成長したかという視点」で評価することができるようになる³⁾。

このように、求める資質・能力とそれらを育成す

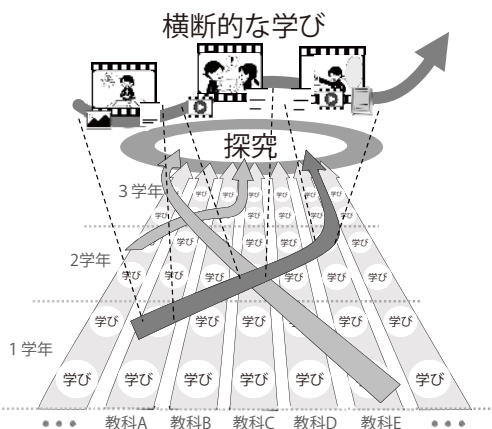


図-2 横断的な学びにおける資質・能力の育成・評価のイメージ

る学習活動を各教科等の学びの中に散りばめ、バランスよく埋め込み、eポートフォリオを活用し実施していくことで、「各教科等の教育内容を相互の関係で捉え、学校教育目標を踏まえた教科等横断的な視点で、その目標の達成に必要な教育の内容を組織的に配列していくこと」によるカリキュラム・マネジメントを実現するための一助になると考えられる⁴⁾。

高大接続改革におけるeポートフォリオの役割

高大接続改革において、高等学校教育、大学入学者選抜と大学教育をつなげることができるツールがeポートフォリオである(図-3)。

生徒は、高等学校のすべての学びを通じてeポートフォリオを蓄積・活用し、その多くのeポートフォリオの中から自ら精選したものの一部を大学入学者選抜で活用する。また、それに引き続き、大学教育でもeポートフォリオを活用し、さらに、学生は、新たな学びを通して、eポートフォリオを蓄積・活用し、学びを深めていく。そうすることで、eポートフォリオは、高等学校教育から大学教育の7年間を通した学びを、あたかも1つのシームレスな学びのようにつなげることができ、生徒・学生のすべての学びを支えることを可能にしてくれる。

特に、大学教育においては、高等学校教育とつながることにより、従来のように入試という見えない壁によって学びが切り離されることなく、高校教育での学びを引き継いで、資質・能力をさらに向上さ

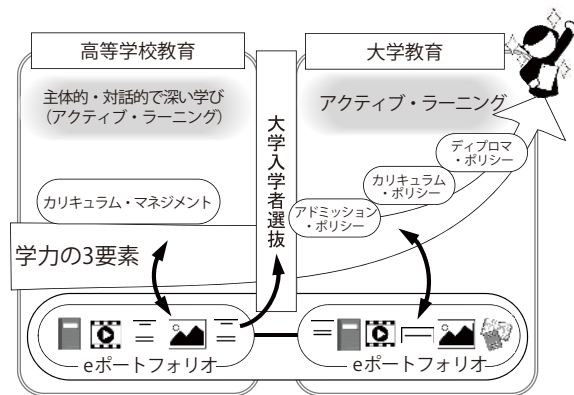


図-3 eポートフォリオがつなぐ7年間の学び



せ、著しい成長を遂げることが可能になると期待できる。まさに、この7年間の学びは、キャリア形成そのものであり、大学卒業時点で、社会の各分野で活躍できる人材に成長できるだろう。

生徒・学生は、eポートフォリオをツールとして活用することで、7年間を通して、見通しを持ち、粘り強く、振り返りながら主体的に学び続けることができ、さらに、蓄積された多量のeポートフォリオを用いることで、7年間を通じた進歩の状況やよい点、その先の可能性まで見える化できるようになる。これが、高大接続改革におけるeポートフォリオを活用した学びのカタチである。

よく、大学入試の出願書類、または、そのための記録をeポートフォリオと呼ぶケースが見受けられるが、それはeポートフォリオのごく限定された一側面にすぎない。eポートフォリオを貯めることが目的であってはならない。それは、手段であり学び

そのものである。eポートフォリオは、生徒・学生の学びを深め、学びの軌跡とその成果を見える化してくれる、とてもポジティブなものである。

参考文献

- 1) 森本康彦：高大接続改革におけるeポートフォリオの役割と活用方法、情報処理学会研究報告、Vol.2018-CLE-26, No.4, pp.1-6 (2018).
- 2) 高大接続システム改革会議：高大接続システム改革会議「最終報告」(2018).
- 3) 文部科学省：高等学校学習指導要領解説 総則編 (2018).
- 4) 中央教育審議会：幼稚園、小学校、中学校、高等学校及び特別支援学校の学習指導要領等の改善及び必要な方策等について(答申) (2016).
- 5) <https://sun.u-gakugei.ac.jp/ePortfolio/assessment/> (参照 2019.02.22)

(2019年2月25日受付)

森本康彦 (正会員) morimoto@u-gakugei.ac.jp

1991年三菱電機(株)・情報技術総合研究所。1996年広島市立牛田中学校 教諭(数学)、2004年千葉学芸高等学校 教諭(情報)、その後、富士常葉大学・准教授を経て、2009年東京学芸大学・准教授、2017年同大学・教授。博士(工学)。eポートフォリオ、ICT活用教育、教育AI活用を専門とする。



プログラミング入門科目の指針と実践例(後編)

久野 靖

電気通信大学

前編のあらまし

前編¹⁾では大学のプログラミング科目について「単位を取得したのにプログラミングができない」という問題意識と、これを克服する指針「離陸ファースト」「多様な水準の演習問題」「演習の重視」「苦手な人の学びを促す」「プログラムが書けるという目標の明示」「プログラミングに唯一の正解はない」「プログラムは自分の頭で作り出す」を提案し、FP(電気通信大学初年次科目「基礎プログラミングおよび演習」)でそれをどう具体化したかを解説した。本稿ではこれらの指針に基づいたFPの具体的なカリキュラム、実施時の知見、結果について述べる。

使用言語とカリキュラム構成

前編で述べた通り、FPでは最初の回ですぐ離陸でき演習に取りかかれるようにRubyを採用しているが、後半はC言語を使用している。これは、本学の2年次以降の各科目がC言語を採用しているため、C言語も学んでおく必要があったことによる。

入門科目であるのに言語を2つ学ばなければならないことは学習者にとって大きな負担となるが、一方で複数の言語を学ぶことの有用性は一般に言われている。そこでFPでも、何とか工夫して「RubyからC」を実現することにした(詳細は後述)。

表-1にFPのカリキュラムを示す(2017年度と2018年度で同一、ただし細かい改良はある)。#11~15がC言語の内容となる。理系の大学であるこ

とを活かし、なじみのある数学に関係する題材を積極的に取り入れた。構成で工夫したのは、「離陸ファースト」のためにその回の基本的な例題を学んだらすぐに演習に入れるように題材を組み合わせた点と、中間部分については個々の内容が「行き止まり」になっていて、そこが理解できなくてもその先の学習に支障がないようにした点である(詳細は後述)。

もう1つの特徴は、全体として入門科目としては異例にコンピュータサイエンス(CS)の内容を多く盛り込んだ点である。これは東京大学が初年次プログラミング科目で取り入れた方針であり²⁾、筆者も非常勤講師としてその内容を教えた際³⁾、「CSのさまざまな題材こそ知的関心につながり学ぶ価値のある内容である」と思うようになった。

学内でシラバスの承認をもらうときに「この内容は2年次以降で扱うのでは」という意見もあったが、「全員が必須内容として身につけるのは2年次以降としても、興味を持てる題材として触れてもらうのは構わないはずで、複数回触れてもらう方が後の科目にとっても好ましい」と述べて了承を得た。

- | |
|--------------------------|
| # 1 プログラミング入門; 様々な誤差 |
| # 2 分岐と反復; 数値積分 |
| # 3 制御構造(2); 配列とその利用 |
| # 4 手続きと抽象化; 再帰呼び出し |
| # 5 2次元配列; レコードと画像 |
| # 6 画像の生成(総合実習) |
| # 7 整列アルゴリズム; 計算量 |
| # 8 計算量(2); 乱数とランダム性 |
| # 9 オブジェクト指向 |
| # 10 動的データ構造; 情報隠蔽 |
| # 11 C言語入門; $f(x)=0$ の求解 |
| # 12 様々な型と配列; 動的計画法 |
| # 13 文字列の操作; パターン探索 |
| # 14 構造体; 表と探索 |
| # 15 チームによる開発(総合実習) |

表-1 各回の内容



1～4：入門部分

入門部分の目標は、まずは普通にプログラムが書けるだけの材料を身につけることとした。中でも # 1 の役割は重要である。この回は「三角形の面積」から始めて、直線的なコードだけで書けるさまざまなプログラムを課題として演習してもらおう。初心者の学生でも 4 行のプログラムを打って動かすことはすぐにでき、また理解もできるので、それなりに熱心に演習に取り組んでいた。

後半の題材として実数計算の誤差を取り上げ、さまざまな誤差の出方を観察する課題を含めている。普通であれば初心者が初回から学ぶことはない内容であるが、一直線のコードで扱え、また「なぜそうなるのか」を考える題材になることから、このようにした。プログラミング経験者である学生にとっても、この内容は初めてのようで、興味を持って取り組んだ様子である。この内容を十分理解するまでに至らなくても、「コンピュータによる計算は 100%

```
$s = []
def e(x) $s.push(x); return $s end
def add
  y = $s.pop; x = $s.pop; $s.push(x+y)
  return $s
end
```

```
irb> e 1
=> [1]
irb> e 3
=> [1, 3]
irb> add
=> [4]
```

図-1 Ruby による RPN 電卓と実行例

```
Pixel = Struct.new(:r, :g, :b)
$img = Array.new(200) do Array.new(300) do
  Struct.new(255, 255, 255) end end

def writeimage(filename)
  open(filename, 'wb') do |f|
    f.puts("P6\n300 200\n255")
    $img.each do |a| a.each do |p|
      f.write(p.to_a.pack("ccc")) end end
  end
end
```

図-2 Ruby による画像の表現と出力

正確などではない」という基本的な事項を記憶にとどめるだけで価値はあると考える。

2 と # 3 では if 文と繰り返し (while ループ、計数ループ) を一通り学び、これらを組み合わせてコードを構成する課題 (fizzbuzz^{☆1} など) を演習してもらおう。制御構造を工夫するとそれに応じて動作が作り出せることは多くの学生の関心を引いていた。配列についてはカリキュラムの都合上ここで一度取り上げているが、あまり重きは置いていない。

4 は手続きが題材であるが、初回から「プログラム」と称して手続き (Ruby では「メソッド」) を作成している。ここではそれを見直し、手続きにまたがったデータ (広域変数) や再帰の話題を取り上げている。RPN (逆ポーランド記法) 電卓の例題 (図-1) などは呼び出し時のかっこが省略できるという Ruby の特性上それらしく見えるため、多くの学生の興味を引いていた。再帰は「数式の定義をそのまま再帰プログラムに対応づける」形で導入することで多くの学生が自然に受け止めていた。

5～6：2次元配列と画像の生成

前編でも述べたように、画像の生成は「自分の頭で作出す」体験を持てる題材としてできるだけ早い回に設定している。# 5 で画像を「RGB 値を表すレコード型の 2 次元配列」として導入し、200 × 300 ピクセル 24 ビットカラーの PPM (Portable PixMap, Unix のツールで多く使われる簡潔なフォーマット) 形式の画像を出力できるようにする (図-2)。すべて自前で書くことで、画像の原理や扱いを「謎」なしに理解してもらおうことを目指した (実際には writeimage の動作は謎のままという学生も多かった)。

その上で基本的な線分、円、長方形、三角形などの塗りつぶしを例示し、# 6 でそれらを利用して (または類似のものを自分で独自に作って) 「美しい画像」を生成する総合課題を課している。

.....
☆1 1, 2, 3……と数値を表示する。ただし、3 の倍数なら fizz, 5 の倍数なら buzz, 両方なら fizzbuzz と (数値の代わりに) 表示する。

レポートとともに提出された作品の一部を図-3に示す。技術的な難しさのレベルは学生によりさまざまだが、どの学生もそれなりに工夫して「自分が作ろうと思う絵」を作っていると感じる。

6 ~ 10 : 中間部分

中間部分は「整列(と時間計算量)」「疑似乱数」「クラス定義」「単連結リスト」とそれぞれ異なる題材を取り上げている。どれもやや高度だがその回で完結し、次回に影響しないため、できなかった場合でもあまりダメージはない。

整列は問題設定が「小さい順に並べる」と明快で、多様なアルゴリズムがあること、またそれらで大きく速度が違ってくことを体験する題材として適している。例題としてバブルソートを提示した後、選択ソート、挿入ソートを誘導つきで書くことを課題としたが、配列に十分慣れていない学生が多く苦戦していた。選択ソート用の「指定範囲での最小値の位置を返す」メソッドの作成が一番やさしい問題であり、これは配列の回に「最大値の位置を返す」演習が既習のため、苦手な学生でも大半がなんとかでき



図-3 提出された画像の例

ていた(ヒントは与えている)。

疑似乱数は「サイコロやコインの利用をシミュレーションする」題材が直観的に分かりやすかったようで、多くの学生が楽しんで実施していた。

クラス定義はクラス中にメソッドを置くという構造を知ってもらうことに力点を置き、「値を put すると覚えていて get で取り出せる」などごく簡単な課題を含めた結果、学生からも「構造は目新しいがプログラムを作ることができた」と肯定的なコメントを多くもらうことができた。

単連結リストは Ruby の最終回であり、やや高度でもチャレンジと思って取り入れた。2017年度はテキストがやや難しい内容から始まり、例題も再帰が分からず苦戦した学生が多かった。

そこで2018年度は、まず irb でリストのセルの連鎖を作り、途中で nil を入れて短くしたり2つのリストをつなげるなどの体験をしてもらった(図-4)。その後「数値のリストで合計」「文字列のリストで全連結」など単純な課題を含む形で演習したが、学生の得手不得手に対応するため例題コードでループ版と再帰版を併記した上、両方でリストをたどって処理するコードのテンプレートもヒントとして提示した。この結果、多くの学生はこれら基本的な課題がこなせ、肯定的なコメントをもらうことができた。2つの版の併記については「ループ版なら分かった」等のコメントもあったが、「再帰を理解する題材となり良かった」というものも多数あった。

11 ~ 15 : C 言語

Ruby で入門後、途中で C 言語に移行することが、本カリキュラムの最大のチャレンジの1つである。特に、Ruby では irb のおかげで入出力を書かなくても済んだが、C ではそうはいかない。

2017年度は main でコマンド引数を受け取り atoi や atof で数値に変換する形で例題を作ったが、これまでと大幅に構造が違うため混乱があった。また、コマンド引数の個数検査や、呼び出される関



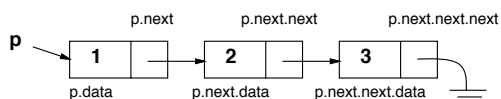
数を下に置くためのプロトタイプ宣言の記述を含めたことも混乱を増大させた。# 11と# 12の間に冬休みが3週間入るため、# 12のときに# 11の内容を覚えていないことも問題だった。

2018年度はこの反省に立ち、図-5を提示して「いままでは本体の関数だけ書いたが、C言語ではmainで入力と出力をする、ただしmainの内容はどのプログラムでも類似」と説明し、scanfとprintfで入出力する形とした。当面はmainを最後に置かせることで、プロトタイプ宣言を# 14の分割コンパイルの説明まで保留した。

```

irb> Cell = Struct.new(:data, :next)
irb> p = Cell.new(1, Cell.new(2,
      Cell.new(3, nil)))
irb> p

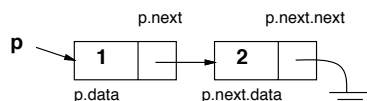
```



```

irb> p.next.next = nil
irb> p

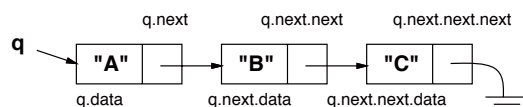
```



```

irb> q = Cell.new("A", Cell.new("B",
      Cell.new("C", nil)))
irb> q

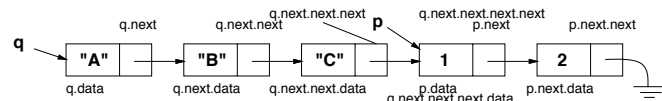
```



```

irb> q.next.next = p
irb> q

```



Quiz: 次のようにできますか?

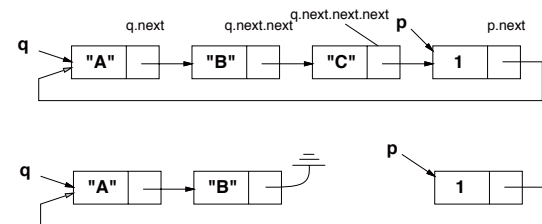


図-4 irbを用いた単連結リストの練習

もう1つの課題として、C言語はメジャーなので既習者も混在している点がある。そこで総合課題を除く# 11～14では内容を「基礎的な前半」「高度な後半」に明確に分け、後半は試験範囲外で経験者のみでよいと明示することで、初級者が挫折を感じないようにしつつ、経験者にもチャレンジしがいのある演習を提供した（予習は全範囲で義務づけているので、初級者でも関心を持ち後半の演習に取り組んでももらえることは多くあった）。

11はC言語の初回なので図-5を提示した後「三角形の面積」を最初の例題とし、Rubyで学んだ題材をCでも書いてみるという演習を中心とした。ループや枝分かれ等の制御構造はCとRubyの対比を示し、機能は同じで書き方だけ変えれば済むという形でまとめて説明した。そして「冬休みに忘れる」問題に対処するため、Rubyですでに取り上げたやさしい課題を多く用意し、通常は「1個以上提出」であるレポートをこの回のみ「10個以上提出」とし、冬休み中に練習するよう呼びかけた。

12は配列が中心だが、C言語では配列アクセスがポインタ演算に基づくため、ポインタも併せて扱う。この内容は一般に「難しい、分かりにくい」とされており、学生にもそのような意見が多かった。配列の扱い自体はRubyと同様でできるが、Ruby部分での配列の練習が不足で身につけていない学生が多いという問題があった。

13は文字列の内容で、文字列がナル文字で終端された文字の配列であることを説明した後、「文字列の長さを求める」「文字列中の文字 c_1 を c_2 に置き換える」など基本的な演習を中心に行った。文字列はイメージしやすく分かるとの感想もあったが、課題は（ヒントを活用して）できても、配列が不得手で自信がないという感想も多かった。

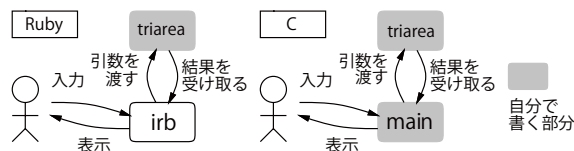


図-5 RubyとCの対比の図

14 は構造体で、Ruby の画像でも使用した RGB 値を表す構造体を定義し、複数の色を混ぜる関数を例題として示した後、1つの色を受け取り明るく／暗くした色を返す、補色（実際には RGB それぞれを 255 から引く）を返すなどの演習問題を課した。16進6桁の RGB 値を入力するとその色が見られる実習ページを用意して実際の色を確認することで、演習の内容が分かりやすいようにした。学生の感想は、よく分かったという意見と（おそらく C 言語の構造体の書き方になじんでいないため）難しいという意見とに二分されていた。

15 は2回目の総合課題で、2～3名のグループで分担してアニメーションを製作する内容である。例題として、図-2と同様の機能を C 言語の API として実装したものを提供し、ただしファイルは書き出すごとに連番のファイル名で作成されるようにし、多数のファイルを生成した後、ツールで動画表示させた。説明時はアニメーションという高度そうな内容に不安を示す学生もいたが、授業時に例題をコピーして動かすことで個々の絵

は # 6 と変わらないことに納得し、グループで作品内容を話し合っ取り組む様子が見られ、多くの力作が提出されている。

結果とまとめ

表-2、表-3に、2018年度授業最終回のアンケート (n=709) 結果抜粋を示す。多くの学生がこの科目を有用と考え、(プログラミングの科目なのでプログラミングについて) 多く学んだと回答している。

期末試験はすべて CBT (Computer Based Test) による短冊問題で、80分の試験で28問を課した。採点は全問、正解(複数の場合もある)と一致で2点、1カ所の欠落、余分、入れ替わりで部分点1点とした(素点満点56点)。素点合計を算出後、ある程度できた学生が50点になるように換算を行い、レポート点と合算している。図-6に2017・2018年度の試験素点合計の分布を示す。前述した授業内容の手直しと、2017年度にやや難しい問題があったものをやさしい問題に変更したことから、全体に成績は向上している。たとえば、95%の学生が8点(4問相当)以上、77%の学生が16点(8問相当)以上の得点を得ている。短冊型の試験はプログラムが書けなければ得点は難しいため、本科目は多くの学生について「基礎的なプログラムが書ける」という目標を達成したのと考えている。

一方で、2018年度の内容についても「配列の練習が不足」など、改良が必要な点は複数見つかっているため、科目内容は今後とも手直ししていきたい。

表-2 学習内容は今後自分の役に立つと思うか?

とても そう思う	そう思う	どちらで もない	思わない	まったく 思わない	未回答
186 26.2%	331 46.7%	145 20.5%	20 2.8%	12 1.7%	15 2.1%

表-3 どれくらい多く新たなことを学んだか?

多数学んだ	まあ学んだ	どちらで もない	少しだけ	まったく ない	未回答
279 39.4%	283 39.9%	92 13.0%	26 3.7%	15 2.1%	14 2.0%

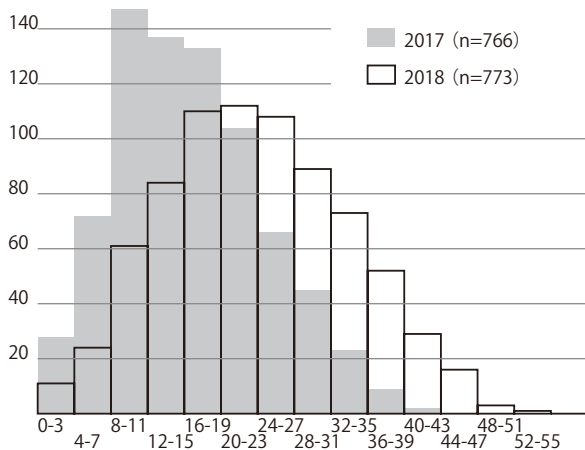


図-6 試験素点の分布

参考文献

- 1) 久野 靖：プログラミング入門科目の指針と実践例（前編），情報処理，Vol.60, No.3, pp.244-247 (2019).
- 2) 森畑明昌：東京大学における全学プログラミング教育，情報処理，Vol.57, No.4, pp.362-365 (Apr. 2016).
- 3) 久野 靖：Ruby による情報科学入門，近代科学社 (June 2018).

(2019年2月5日受付)

久野 靖 (正会員) y-kuno@uec.ac.jp

1984年東京工業大学理工学研究科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、電気通信大学情報理工学研究科教授。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、情報教育に関心を持つ。

