

プログラミング教育における 学習者の到達度に応じた傾向 Patterns Depending on Students' Achievement Degree on Programming Education

永吉 翔*, 山岸 秀一*, 松本 慎平*, 沖中 涼*, 松岡 伸彦†

Sho Nagayoshi*, Shuichi Yamagishi*, Shimpei Matsumoto*, Ryo Okinaka* and Nobuhiko Matsuoka†

*広島工業大学情報学部知的情報システム学科

†広島工業大学大学院工学系研究科

〒 731-5193 広島県広島市佐伯区三宅 2 丁目 1-1

Email: {ba09180, s.yamagishi.if, s.matsumoto.gk, ba09048, m161206}@cc.it-hiroshima.ac.jp

Abstract—A web based programming learning support system is now planned to be developed aiming for the holistic improvement of programming technique. This software is for students with insufficient understanding, and aims to improve their technique by after class hour learning. Before the authors develop the software, this paper addresses to give the part of specification of the software. Especially as the first step, this paper examines what kind of students are no good at programming, and what needs to improve the understanding of the students. The authors consider that the trends and characteristics of students with low programming skill need to be shown for supporting their learning. The short exams were conducted in several times in lectures related with programming for the first and second grade students currently being held, and the results were examined depending on the achievement degree of students. From the analysis, the valuable speculation about students who are no good at programming was obtained.

I. はじめに

現在、著者らは、大学1年生と2年生を対象としたプログラミング・アルゴリズム関連科目に携わっている。まず、1年生の前期に開講されているアルゴリズム導入科目では、フローチャート記述とトレースを重点的に取り上げている。1年生の後期に開講されるプログラミング導入科目では、C言語の基本文法教示を目的としている。2年生の前期に開講されるアルゴリズム応用科目では、データ操作や探索といったアルゴリズムの概念を述べると共に、そのアルゴリズムのプログラミング実装法を紹介している。その後、2年生の後期に開講される2コマ連続の実験では、これまでの学習の集大成として、複雑なデータ操作や知的処理に基づいた探索プログラムを開発している。以上述べた教養課程でのプログラミング導入講義では、従来、一般的な指導方法に基づいて開発技法を教示してきた。具体的には、各回ひとつの重要な機能を話題として取り上げ、その機能を利用するための基本文法や応用事例(使い方)をいくつか紹介した後、時間内外での演習問題を課していた。しかしながら、従来の教示法に基づく限りは、プログラミングを苦手とする学習者に対して、苦手意識を克服するまでの十分な指導を行うことができなかった。プログラミングを極端に不得手とする学生は各学年一定数存在しており、また、こ

の傾向は全学年で同様であるように感じられる。情報系の専門領域を専攻する学生にとっては、プログラミング技能は特に不可欠である。事実、卒業研究のみならず就職活動でもプログラミングスキルは必要であり、多くの一般企業は、情報系学生に対して、高度な水準でのプログラミング技能を求めている。よって、プログラミング関連講義に対しては、一層の工夫が必要であると著者らは考えている。

こうした背景を踏まえて、著者らは、プログラミング技能の全体的な向上を目指して、プログラミング関連科目の教育研究に着手した。具体的には、プログラミングを不得手とする学習者を対象とし、プログラミングに対する苦手意識を少しでも軽減できるような自学学習環境の構築運用を最終的な目標に設定して、研究プロジェクトを開始させた。自学学習環境としては、インターネット上で利用可能なプログラミング学習支援システムの開発と導入を計画している。娯楽的要素を取り込むことで、楽しみながらプログラミングの概念を学習できるソフトウェアを考えている。

本稿は、ソフトウェア開発に先立って、ソフトウェアに実装すべき機能や特徴を明確にすることを目的とする。試験によってプログラミング力が客観的に低い学習者を選出し、彼らのプログラミングに対する特徴や傾向をデータ分析から明らかにする。現在開催されている大学1年生と2年生のアルゴリズム関連科目内で小テストを複数回実施し、学習者の到達度に応じた傾向を把握した。その結果、プログラミング学習支援ソフトウェア開発のための有益な知見を得ることができた。

II. 現在の取り組み

著者らは現在、大学1年初期から2年後期までを対象として、連続的なプログラミング導入教育構築に向けて取り組んでいる。特に、考える習慣やプログラムに触れる機会を少しでも多く用意したいと考えている。ソフトウェア開発導入を目的とした教育法の改善に先立って、現在、プログラミングに対して少しでも苦手意識を持たないようにすることを意識して、講義を計画している。具体的な取り組みとしては、今年度から、プログラミング導入教育の初期段階において、アルゴリズムと呼ばれるソフトウェアを利

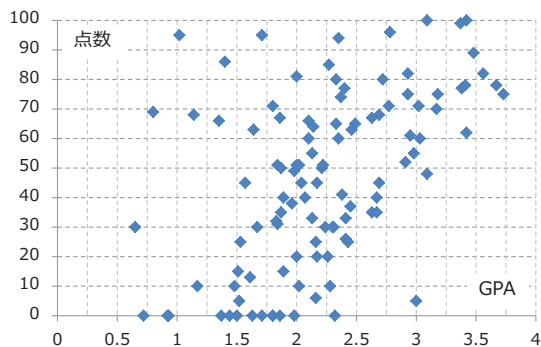


図 1. 試験の得点と GPA との関係

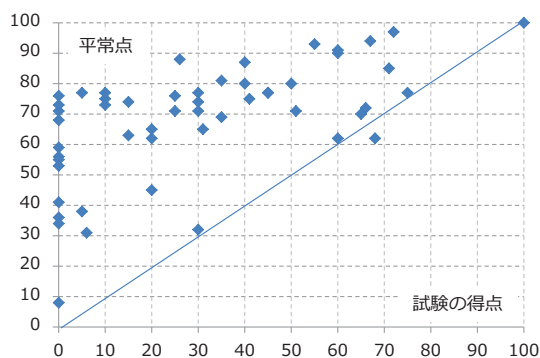


図 2. 試験の得点と授業の取り組み状況との関係

用している [1]. アルゴリズムは、プログラミングの基本となる論理的思考をゲーム感覚で修得するための課題解決型ソフトウェアであり、著者らの構想に近い機能を実装している。アルゴリズム導入の結果、学習者からは好意的な評価を得ているものの、授業の進行に伴って、プログラミングの講義に追いつけない学習者が目につくようになってきた。講義の終盤に差し掛かり、到達度確認テストを実施した結果、プログラミングの根本的な概念を十分に理解できていない学習者の存在が明らかとなった。

プログラミングは向き不向きが強く出るように経験的に感じられるが、Dehnadi らは、プログラミングができるかできないかは比較的簡単なテストによって、プログラミングの訓練を始める前の段階で分かると主張している [2]。Dehnadi らの主張の中では、プログラミングの素質がある人間の割合は、年齢や性別や教育レベルでは変わらないと述べている。考え方を一貫していればそこから意味不明な結論が出てもしとする素質はプログラミングにとって重要であると Dehnadi らは結論付けている。そして、プログラミングを楽しみながら取り組める素質のある人間は 40% ぐらいであろうと述べている。Dehnadi らの主張を検証することは重要であるが、プログラミング自体には楽しみを見出す必要はなく、プログラミングをできるだけ最低限の力さえあれば十分であると著者らは考えている。

プログラミングに関連する先行研究としては、プログラミング学習の教授法に関する報告と、プログラミング学習支援ソフトウェアの開発に関する報告の 2 種類に大別される。例えば深町は、文系の学生がコンピュータプログラミングを習得する際に陥りやすい間違いを具体例を挙げながら紹介し、それらに対する取り組みを紹介した [3]。深町によると、数式を読み解く能力や、考えていることを数式として表す訓練が不十分であることがプログラミングの習得を阻んでいる要因のひとつであると指摘している。また、コンパイル時および実行時のエラーの発見ができない点も問題であると述べている。

III. 分析の結果

本稿では、プログラミング教育の初期段階において、事前調査的な試験によりプログラミング力が客観的に低いであろうと考えられる学習者を選出し、特徴や傾向を掴むことを課題とした。まず、講義内で理解度の中間到達度試験を行い、学習者個々の得点と全体的な成績 (GPA) との関係性を明らかにした。講義では C 言語のプログラミングを扱っ

ており、配列までの内容で、ポインタや関数など応用処理は一切取り上げていない。結果を図 1 に示す。相関係数は 0.55 であり、弱い関係性が確認できる。GPA が低い学習者であっても試験の得点が特に良かった学習者も存在しており、GPA が低いからといってプログラミング力が低いとは限らない結果が得られた。ただし、GPA が高い学生は、プログラミング力も高い傾向にある。次に、試験の得点と授業の取り組み状況との関係を図 2 に示す。授業の取り組み状況とは、課題や講義内の小テストの点数を累積したものである。講義内の小テストは、当日内に進んだ内容の確認を行うためのもので、講義内容とほぼ同様の問題が与えられる。一方中間到達度試験では、講義の内容を踏まえた応用の問題が 25 点分出题されていた。図 2 から、平常点が高い、すなわち真面目に取り組む学習者であっても、応用問題ができなかったことにより、試験の得点が低くなった学習者が多く確認できる。応用問題は 30 行程度のプログラムであったが、複雑な分数の計算処理を要求する問題であった。以上から、深町の指摘の通り、考えていることを数式として表す力が重要ではないかという仮説を得た。

IV. おわりに

本稿では、どのような学習者がプログラミングを不得手とするか、また、対象の学習者が理解を向上させるために何が必要かを考察した。

謝辞

本研究を遂行するにあたり、一般社団法人 電子情報技術産業協会が開発したアルゴリズムを利用させて頂きました。

参考文献

- [1] 一般社団法人 電子情報技術産業協会, アルゴリズム体験ゲーム アルゴリズム, <http://home.jeita.or.jp/is/highschool/algo/>, 2012 年 6 月 25 日参照.
- [2] Saeed Dehnadi and Richard Bornat, The camel has two humps (working title), <http://www.eis.mdx.ac.uk/research/PhDArea/saeed/paper1.pdf>, 2012 年 6 月 25 日参照.
- [3] 深町修一, 文系の学生に対するコンピュータプログラミング教育の一考察, 福岡国際大学紀要, No.23, pp.39-45, 2010.

問い合わせ先

〒731-5193

広島県広島市佐伯区三宅 2 丁目 1-1

広島工業大学情報学部知的情報システム学科

山岸 秀一