

特集：プログラミング・情報技術教育に関する実践・支援システム

# プログラミングにおける構造的理解のための部品の 段階的拡張手法の提案とそのシステムの開発

古池 謙人<sup>\*,(注1)</sup>, 東本 崇仁<sup>\*</sup>

## Proposal of Expandable Modular Statements for Structural Understanding on Programming and Development of Learning Support System

Kento KOIKE<sup>\*</sup>, Takahito TOMOTO<sup>\*</sup>

### 1. はじめに

プログラミング教育においては、教師がサンプルコードの提示を行い、学習者が提示されたコードを入力して動かすという工程を課題形式で行うことが多い。この場合、サンプルコードについての説明を教師が一方的に行うものの、学習者はただ単純に課題をこなすためだけに学習を行うため、学習者自身が能動的にサンプルコードの意味や仕組みについて考える機会が不足する場合が多く存在することが指摘されている<sup>(1)</sup>。このような仕組みについての理解が不足した場合、学習者は獲得してきた知識を構造的に整理することができず、理解した内容を段階的に発展させることができないと筆者らは考える。また、個々の知識やアルゴリズムの理解が十分な学習者でも、実際の要求からおおよその流れを加味した手続き的な設計を行うことはできるものの、コードを有意な塊ごとに部品化を行い、今まで作成してきたコードを部品として、再利用性を高めるといった構造的な設計は苦手とする場合が多い。このような部品としての段階的な理解を進めるためには、複数行のコードの意味を理解したうえで、自ら構築し、再利用し、さらに大きな部品へと展開させることが重要である。しかし、学習者にとってみれば、コードの単一行は理解できていても、複数行になると理解が困難になることがある<sup>(2)</sup>。

Shneiderman は、プログラミングの理解において、単一の要素を複合して一つの大きな要素として理解することを繰り返して、構造的な理解に至ることの重要性を主張している<sup>(3)</sup>。本研究では、プログラミング学習においては、コードの有意な塊を部品として認識・習得し、複数の部品を組み合わせたより大きな部品を認識・習得、そして、それらの部品を組み合わせられる能力が重要であると考えられる。

よって、筆者らは、既有知識を用いてコードを有意な塊ごとに部品として認識を行い、それらの部品を再利用し、知識を組み合わせるより大きな部品の構築を構造的に行えるように学習者の既有知識が整理されることを「構造的理解」と位置づけた。

そこで本研究では、プログラミングにおける構造的理解を目指した「部品の段階的拡張手法」と提案手法を用いた学習支援システムの開発、また、開発したシステムを用いてソートを事例に学習効果の評価を行った。

### 2. 先行研究

学習者が知識によって各行を理解することはできるものの、複数行になった際に理解が困難になることから、学習者がコードの各行についての関係性を正しく理解することを目的とした研究が行われている<sup>(1)(5)</sup>。渡辺ら<sup>(5)</sup>の研究では、金森らが提案するプログラミ

\* 東京工芸大学工学部 (Faculty of Engineering, Tokyo Polytechnic University)

(注1) 現所属：東京工芸大学大学院工学研究科

受付日：2017年6月5日；再受付日：2017年9月27日；採録日：2017年10月14日