プログラミング学習における部品の段階的拡張手法の提案

Proposal of Granularity Expand Method in Parts for Learning on Programming

古池 謙人[*1], 東本 崇仁[*1]
Kento KOIKE[*1], Takahito TOMOTO[*2]
[*1] 東京工芸大学工学部
[*1]Faculty of Engineering, Tokyo Polytechnic University
Email: c1418030@st.t-kougei.ac.jp

あらまし：プログラミング学習では，プログラミングに対する構造的な理解が不十分になると部品ごとの特徴を理解した本質的な設計を行うことが難しくなる．そこで本研究では，学習者が一行ずつのコードや，より小さな部品を組み合わせ，段階的に大きな部品へ発展させていくことで，プログラミング学習における構造的な理解を目的とした学習手法の提案を行った．加えて，提案手法を用いたシステムを試作した．
キーワード：部品の段階的拡張手法，段階的な思考

1. はじめに

個々の知識やアルゴリズムの理解が十分な学習者でも，個々の知識の関係性や特徴を理解していないと，コードの有意な塊ごとに部品化を行い，部品の再利用性を高めるといった構造的な設計を行うことは難しい．加えて，個々の知識の関係性や特徴の理解が十分でない学習者は，与えられた複数行のコードを有意な塊ごとに部品として認識することができておらず，単一行として理解できていても複数行のまとまりになると，そのまとまりが示す意味が理解できない場合がある．

著者らは，要求から構造的な設計を行うために，拡充した知識からコードの有意な塊ごとに部品として認識を行い，個々の部品の関係性や特徴を理解し，より大きな部品の構築を行えるように知識を整理することが重要であると考えた．

よって本研究ではプログラミングにおける構造的な理解を志向した，「部品の段階的拡張手法」の提案を行う．

2. 関連研究

学習者が自律的に個々の知識の関係性や特徴を理解するのは容易ではなく，その重要性も金森ら[(1)]やArai et al.[(2)]によって主張されている．また，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ための研究が渡辺ら[(3)]によって行われている．渡辺らの研究では，金森らが知恵案するプログラミングプロセスの中で，プログラムの読解と意味理解を行うことがプログラミング学習において有意義であると主張しており，プログラムの読解と意味理解を支援する手段として段階的抽象化プロセスを提案している．この段階的抽象化プロセスは，与えられたコードの中で一連の処理だと考えられる部分をまとめ，その意味を考えるということを繰り返すことで，“段階的に読む”学習を支援している．例えば，(1)c に a を代入，(2)a に b を代入，(3)b に c を代入というコードが与えられたときに，これらのコードを一行ずつみれば単純な代入の繰り返しだが，3 行まとめて意味を考えると，a と b のスワップを行っているという，新しい概念の発見ができるとしている．著者らは，この渡辺らの段階的抽象化プロセスによって，コードの各行の関係性を正しく理解することができ，有意な塊としての認識が行えると考える．

しかし，渡辺らの段階的抽象化プロセスでは，プログラムを構成している部品について着目しているものの，部品同士を組み合わせてより大きな部品を作るといった，部品の発展については扱っていない．

そこで著者らは，コードの有意な塊ごとに部品として認識するのと同時に，その部品を段階的に発展させ組み合わせることによって，どうすれば部品を再利用できるかなどを考える“段階的に作る”学習に焦点を当てることで，よりプログラミングにおいてコードの有意な塊ごとの構造的な理解につながるのではないかと考えた．

3. 提案手法

通常，熟達者はプログラミングの際に他人のプログラムを部品ごとに構造的に読む行為と，自ら部品ごとに再利用性などを考慮しながら構造的にプログラムを設計する行為を日常的に行っている．熟達者は他人のプログラムの構造を理解し，自らの知識に単一のコードごととして見た新たな知識としてだけでなく，複数行の有意なまとまりごとに新たな部品として加えることで，加わった部品を用いた新たな設計を行えるようになる．

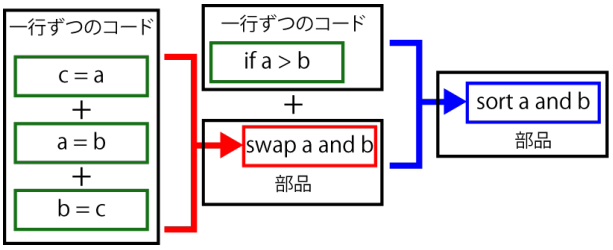

図 1　提案する「部品の段階的拡張手法」

一方，初学者など個々の知識の関係性や特徴を理解できていない学習者は，他人のプログラムを読む際に単一のコードごととしてしか新たな知識に加えられず，複数行の有意なまとまりとして構造的に理解する場合と比較して，再利用の際に利便性が高いとは言い難い．

そこで著者らは，プログラミング学習において，知識の拡充だけでなく，知識の関係性を正しく理解する構造的な理解が重要だと考え，この構造的な理解を支援するための手法を提案する．提案手法は，渡辺らの段階的抽象化プロセスを用いて，まず，学習者がプログラムにおける一行ずつのコードから有意な塊ごとに部品の構築を行う．さらに，構築した部品に対して新たな処理や既存の部品を追加したり，すでに構築された部品に対して部分的な修正を行ったりなどして部品を変化させる（図 1）．こうした部品の変化を繰り返すことで，部品の構造を段階的に拡張していく．このようにして，学習者に部品の構造を段階的に拡張することで部品ごとにおける関係性の理解を促し，学習が行える．加えて，部品を段階的に拡張する行為そのものによって，学習者が自身で大きな部品を作る際の練習となり，実際に設計を行う際に役立つと考えられる．この過程を，著者らは「部品の段階的拡張手法」と定義した．

段階的に「読む学習」の有用性については金森ら[(1)]によって主張されており，そこに加えて，従来の学習行程である「作る学習」によって段階的にプログラムを発展できれば，設計を行う能力だけでなく，アルゴリズムの構造的な理解につながり，プログラミング学習としても有用ではないかと著者らは考える．

4.　システムの試作

本研究では，上述の「部品の段階的拡張手法」を用いた学習支援システム（以下，本システム）の試作を行った（図 2）．本システムでは，各部品をブロックと位置づけ，「部品の段階的拡張手法」の活用として，構築すべきブロックを目標としてあらかじめ設定し，ブロックの組み合わせや発想が容易なものから順に習得を目指すことで，学習者におけるプログラムの構造的な理解を目的としたプログラミングの学習の支援を図る．

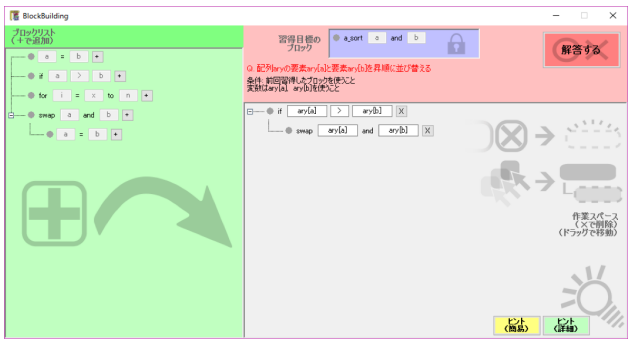

図 2　本システムのインタフェイス

5.　学習方法

本システムを用いて学習者が学習を行う際の手順を示す．まず，(1) 習得目標とされるブロックが提示され，次に，(2) 構築に利用できるブロックの一覧から，実際に習得目標のブロックを構築するために，必要なブロックを複数選択，追加を行う．(3) 追加されたブロックごとの順序や，入れ子構造などの階層を整理，設定する．最後に (4) 解答を行い，正解であった場合は次の課題に，不正解であった場合にはシステムからのフィードバックを元に修正を行う．これらの手順を通して学習者は学習する．

加えて，本システムでは学習者が部品について段階的な発展を行うために系列的な課題を用意し，必ず前後の課題が関連するように設定している．例えばスワップの習得を行なった後に，そのスワップを用いて 2 変数の並び替えの習得を行なうなどである．

6.　まとめと今後の課題

本研究では，プログラミング学習において，知識の拡充だけでなく，拡充した知識の関係性を理解する構造的な理解を行うことで，コードの有意な塊ごとに部品化を行い，部品の再利用性を高めるといった構造的な設計が行えると考えた．そこで，構造的な理解を支援するために「部品の段階的拡張手法」を提案し，システムを試作した．

しかし，現状のシステムでは部品の追加や，その組み合わせについて実装しているものの，部品の修正については取り扱っていない．また，問題の系列についても詳細な検討は行なっていない．

そのため，今後の課題として，システムで部品の修正が扱えるように追加実装を行い，問題の系列においては，詳細化と共に問題系列ごとの分岐・統合などについても検討していきたい．

参考文献

(1) 金森春樹, 東本崇仁, 米谷雄介, 赤倉貴子： “プログラミングプロセスにおける「プログラムを読む学習」の提案及び 「意味理解」 プロセスの学習支援システムの開発,” 電子情報通信学会論文誌 D, vol. 97, no. 12, pp. 1843‐1846, (2014).

(2) T. Arai, H. Kanamori, T. Tomoto, Y. Kometani, T. Akakura: “Development of a learning support system for source code reading comprehension,” in International Conference on Human Interface and the Management of Information, pp. 12–19, (2014).

(3) 渡辺圭祐, 東本崇仁, 赤倉貴子： “段階的抽象化を用いたプログラムを読む学習の支援システムの開発とその評価 (教育工学),” 電子情報通信学会技術研究報告= IEICE Tech. Rep. 信学技報, vol. 115, no. 50, pp. 49‐54, (2015).