

制御構造における変数の値変化と役割の理解を目的とした 独自の図表現によるプログラミング学習支援の研究

飯田和佐[†] 大沢卓[†] 新井聖也[†] 水谷晃三[†] 荒井正之[†]

帝京大学理工学部ヒューマン情報システム学科[†]

1. はじめに

プログラミング初学者を対象にして、プログラムの制御構造の理解促進を目的とした研究が行われている[1]. 本研究では、条件分岐の中に反復処理があったり、多重の反復処理があったりするなどの複雑な制御構造の理解を促進するために、独自の図を考案してこの図を用いた学習支援ツールを開発する。

2. 研究背景と新たな表現方法の検討

2.1 制御構造の理解のレベル

初学者がプログラムの制御構造を理解するとき、その理解度に適した学習支援が必要となる。本研究では、プログラムの制御構造の理解のレベルを次のように定義した。

- ①プログラムの全体構成の理解：フィールド変数、メソッド、クラスを列挙できる程度の理解
- ②全体的な処理の流れの理解：プログラムの開始から、終了までの大まかな流れを把握できる程度の理解
- ③制御の概念の理解：接続、条件分岐、反復のそれぞれの制御構造についての（特定のプログラム言語に依存しない）概念的な理解
- ④プログラムにおける制御構文の（文法上の）意味の理解：プログラムソース上でそれぞれの制御構造の理解
- ⑤変数の値の変化や制御構造の役割の理解：それぞれの変数や制御構造の役割、変数の値がどのような影響を受けて変化するかを理解

本研究では⑤のレベルに到達するための方法を検討している。

2.2 トレース表やフローチャートによる表現方法の問題点

既存研究の例として、トレース表やフローチャートをわかりやすく表現することを試みた研究がある[2,3,4]. トレース表ではプログラムソースの1ステップごとに変数値を書きだしていく方法で変数の値の変化を可視化する。ステップ単位で理解できるようになるが、①②に対しては直感的にはわかりにくく理解しにくい。また、

④についても1ステップごとにプログラムソースと対応させながら理解する必要があり、初学者にとっては負荷が大きい。複雑な制御構造になるとトレース行が増えるデメリットもある。

フローチャートでは①②の理解は比較的容易であるが、複雑な制御構造になると図も複雑になりかえって理解しにくくなる。④についてもフローチャートだけでは理解しにくく、フローチャートとソースコードを対応付けて理解する能力も必要になり初学者には負担が大きい。

また、トレース表やフローチャートに共通する問題点として変数のスコープの表現がないことがあげられる。変数のスコープを正しく把握していないと、制御構文とそれによって影響を受ける変数の関係や、変数の値の変化を理解しにくく⑤の理解に到達できないと考えられる。

2.3 独自の図による表現の検討

前述の問題点を踏まえ、独自の図による表現方法を検討した。具体例を図1に示す。

本提案の図は、変数名、この変数と関係のある行を示すマーク（以下関係行マーカと呼ぶ）、スコープの3つの基本要素から構成される。これらを一組としてプログラムに必要な変数の個数分、横に並べて表現する。また、ブロックごとに図の背景を色分けして表現する。本案では、メソッドは橙、条件分岐はピンク、反復処理は黄緑で表現しているが、多重の反復処理などでは色の明度を変えるなどの工夫も考えられる。

さらに、デバッガによるプログラム実行の際には、変数の値の変化と実行位置を表現する。実行位置を赤線で表現し、更にその位置での変数の値を、スコープを表す縦棒と赤線が交わる位置に表示する。例えば図1において、(b)のステップ実行では変数 `fact` に1が代入されたことがわかる。このとき、変数 `x` と `n` の変数値も表示されているが、`fact` には関係行マーカがあるため把握しやすくなっている。

以上により⑤の理解を促す効果が期待できる。

3. 支援ツールの試作

2章で述べた独自の図による学習支援を実現するために、Processing言語を対象としたPDE (Processing Development Environment) 上の支援ツールを試作することとした。支援ツールの動

Programming Learning Tool for Understanding the Value Changing and the Role of a Variable in Control Structures.

[†]Kazusa Iida, Suguru Ohsawa, Seiya Arai, Kozo Mizutani, Masayuki Arai, Department of Human Information Systems, Faculty of Science and Engineering, Teikyo University.

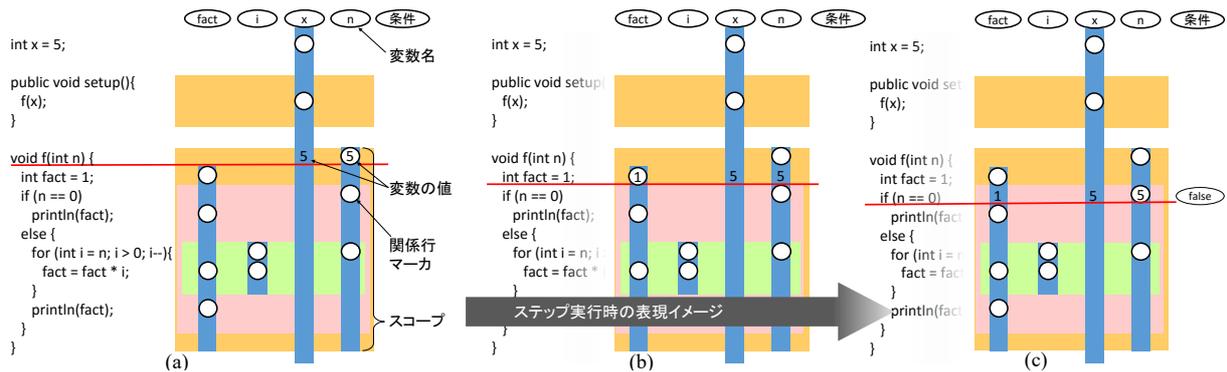


図 1. 独自の図表現によるプログラム構造の可視化

作概要を図 2 に示す. PDE 上でソースを作成し, 支援ツールを起動すると, 支援ツール上で構文解析を行い, 変数名や行番号を取得し, その情報を用いて図を描画する. PDE のデバッガ機能を使用すると実行が図中に表示され, 同時に変数の値が表示される. 現段階では Processing の静的スケッチに限定しているが, 将来的には動的スケッチも対象にする予定である.

4. 考察

フローチャート, トレース表, 本提案の図について制御構造の理解①~⑤に対する理解の容易性を主観的に評価した. その結果を表 1 に示す.

前述したように, ①②においてトレース表は各行の処理を追っていかねばならず全体的な制御構造は把握しにくい. フローチャートは複数のメソッド呼び出しを伴う場合把握しにくくなるが, 本提案図は変数スコープやブロックごとの色分けにより直感的に把握しやすい.

⑤においてフローチャートは変数の値の変化までは図そのものからは理解できないと考えられる. トレース表は値の変化は分かっても, この制御で変化しているかを判断するにはソースコードの行番号と照らし合わせて判断しなければならない. また, いずれの表現方法においても各変数と制御構造の役割までは理解できないと考えられるため△とした.

しかしながら, 提案する図では各変数のスコープやブロックを示しているほか, 関係行マーカーがあることにより, 他の表現方法よりも⑤のレベルの理解を促進する効果が高いと考えられる. 以上のことから, 本提案の図による表現方法は優位性があると考えられる.

5. おわりに

本研究では, プログラミング初学者を対象とした, プログラムの制御構造の理解促進を目的とする独自の図による新しい表現方法を検討した. Processing の静的スケッチを対象とした支援ツールを試作するとともに, 本提案手法の優位

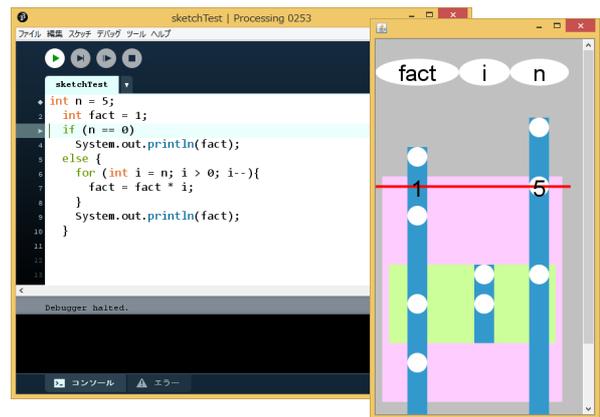


図 2. PDE と試作ツールの実行画面例

表 1. 制御構造の理解における比較

	フローチャート	トレース表	提案する図
①	△	×	○
②	○	△	○
③※1	—	—	—
④	○	○	○
⑤	×※2	△	△

○: 理解できる △: 理解しにくい ×: 理解できない

※1: ③はプログラムにおける制御構造の理解の前提となる理解レベルであるので比較対象外とした.

※2: フローチャートを発展させたビジュアルトレースツールなどでは△.

性について述べた.

参考文献

- [1] Juha Sorva, Visual Program Simulation in Introductory Programming Education, Aalto University publication series, 2012.
- [2] 古宮誠一, 今泉俊幸, 橋浦弘明, 松浦佐江子, プログラミング学習支援環境 AZUR-ブロック構造と関数動作の可視化による支援-, 研究報告ソフトウェア工学(SE), pp.1-8, 2014.
- [3] 森田昌樹, 香川考司, WebGL とジェスチャー認識デバイスを用いたプログラミング例題提示システムの開発, 第 39 回教育システム情報学会全国大会講演 論文集, pp391-392, 2014.
- [4] 荒井正之, プログラミング初学者のトレース能力の修得を目的とした学習支援システム, E-Learning 学会 論文誌, Vol.10, pp.54-62, 2010.