

プログラミング入門教育における教育補助システムの開発

石井 侑太[†] 小野 郁真[†] 河野 美波[†] 高野 辰之^{††} 宮川 治^{†††}

東京電機大学情報環境学部情報環境学科[†] 東京電機大学大学院先端科学技術研究科^{††}
東京電機大学情報環境学部^{†††}

1. はじめに

プログラミング教育において、初学者がプログラムの動作を理解することは難しい[1]。この理由として挙げられるのが、プログラムの実行結果からプログラムの流れを読み取る力の不足である。教科書などではプログラムの流れを図示化し、わかりやすく記述しているが、その図から流れを理解するのは困難である。そのため、教授者は逐次トレースしながら説明するが、それでも引数・返却値の説明は複雑になってしまい、初学者が適切に理解するのは容易ではない。

そこで我々は、初学者の理解度向上を目的とし、教授者が利用する教育補助システムを提案する。本システムは Java 言語を対象とし、初学者が理解することの難しいプログラムの流れを標準出力と共に逐次可視化するものである。

2. システム概要

プログラムの可視化を行うためには、プログラムの流れや標準出力等の情報を得るための解析が必要となる。プログラムから、この情報を得るために動的解析を行う。本システムでは、この解析部分と可視化部分から、この情報を分離するための中間層を設ける。このシステムは Java 言語で実装を行っている。

2.1 解析部分

解析部分はプログラムを解析し、可視化するために必要な情報の取得を行う。解析は Java の標準のデバッガ API である JDI (Java Debug Interface) を利用した。JDI を利用することにより、仮想マシンの内部情報へのアクセスや操作をすることが可能となる。

また、本システムでは解析によって得られたデータのモデル化を行なっている。モデル化を行うことにより、複雑なデータ構造を単純化でき、その再利用も促進される。このモデルは解

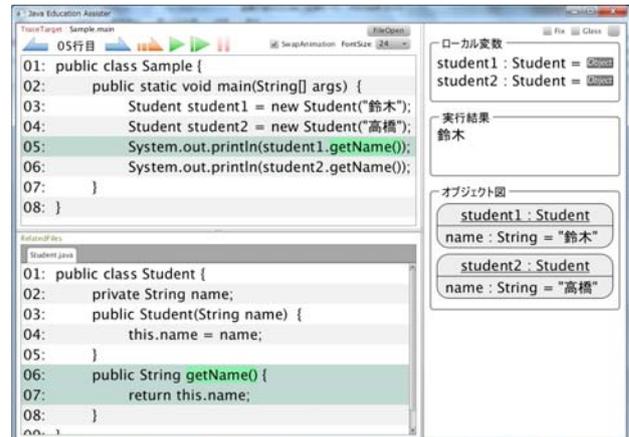


図1 可視化部分の画面

析部分及び可視化部分で使用される。

2.2 中間層

中間層は、解析部分で得られた情報を元に作成された XML 形式のデータと、そのモデル化されたクラスである。このデータは、解析部分でモデル化された Java オブジェクトを、JAXB を用いて XML 形式にシリアライズしたものである。そして、このデータを可視化部分で JAXB を用いて Java オブジェクトにデシリアライズし、利用する。中間層を用いることにより、解析部分と可視化部分の関係を疎結合に保つことができ、一方の変更がもう一方に影響することが無い。

2.3 可視化部分

可視化部分は JavaFX2.0 を用いて実装した。JavaFX2.0 とはリッチクライアントを構築することが可能な Java の GUI ライブラリである。JavaFX2.0 の多彩な GUI 機能を用いて可視化を行うことが可能である。

可視化部分の画面を図1に示す。画面右側に標準出力等の情報を示す状態部を表示する。画面左側上部にトレースを行うソースコードを示すトレース部を表示し、下部に上部のソースコードから利用するクラスを示す呼び出し部を表示する。これにより、振る舞いの呼び出し元と呼び出し先を分割して表示でき、引数・返却値の関係を明確に可視化することが可能となる。

手続き型プログラミングでは主に引数・返却値の関係を、オブジェクト指向プログラミングでは引数・返却値の関係に加えてオブジェクト

Education Support System in Introduction to Programming Courses

[†]Yuta ISHII, Ikuma ONO, Minami KONO,
Graduate School of Information Environment,
School of Information Environment, Tokyo Denki University
^{††}Tatsuyuki TAKANO, Graduate School of Advanced Science
and Technology, Tokyo Denki University
^{†††}Osamu MIYAKAWA, School of Information Environment,
Tokyo Denki University

図の可視化を行う。

3. 機能

3.1 可視化するプログラムの選択

本システムは、プログラムを選択し、解析を行う。そのため、任意のプログラムを可視化することが可能である。

3.2 逐次トレース

選択されたプログラムのトレースを逐次進めることが可能であり、その時々でのローカル変数、標準出力、オブジェクト図を画面右側の状態部で表示する。また、トレースを逐次進めるだけでなく、戻すことも可能である。

3.3 引数・返却値のアニメーション

逐次トレース中の振る舞いの呼び出し時に、引数・返却値が存在した場合、振る舞いの呼び出し元と呼び出し先に着目し、引数・返却値の関係を、アニメーションを用いて表現する。

i 引数

振る舞いに引数が存在する場合、呼び出し元の実引数と呼び出し先の仮引数の受け渡しを、アニメーションを用いて表現する。まず呼び出し元の実引数の値をトレース部から呼び出し部に遷移させる。そして、呼び出し先の仮引数が実引数の値に変わる様子を表現し、2つの関係を表す。

ii 返却値

振る舞いに返却値が存在する場合、引数が存在する時と同様に呼び出し元と返却値の関係を、アニメーションを用いて表現する。まず、返却値を呼び出し部からトレース部に遷移させる。そして、呼び出し元が返された値に変わる様子を表現し、2つの関係を表す。

4. 試験運用

手続き型プログラミングの講義中に引数・返却値を説明するプログラムの可視化を行い、アンケートを取った。対象は引数・返却値について説明を受けた学習者41名である。内容には引数・返却値を「今まで理解していたか」と「今回のシステムを利用しながらの説明でより理解できたか」の項目がある。アンケート結果を表1、表2に示す。

また、「普段の説明と比べてどう思ったか」「自分のソースコードで利用したいか」の質問に対しては表3、表4に示す。

5. 考察

引数・返却値共に「より理解できた」が多く、学習者の理解を促進させることができた。しかし、質問には「今まで理解していたか」と「より理解できたか」の項目のみのため、本システ

表1 引数の理解

	より理解できた	それ以外
今まで理解していた	25人 (61.0%)	7人 (17.1%)
今まで理解していなかった	7人 (17.1%)	2人 (4.9%)

表2 返却値の理解

	より理解できた	それ以外
今まで理解していた	18人 (43.9%)	8人 (19.5%)
今まで理解していなかった	10人 (24.4%)	5人 (12.2%)

表3 普段の説明と比べてどうだったか

大変良い	10人 (24.4%)
良い	30人 (73.2%)
悪い	1人 (2.4%)
大変悪い	0人 (0.0%)

表4 自分のソースコードで利用したいか

是非利用したい	14人 (34.1%)
利用したい	26人 (63.4%)
利用したくない	1人 (2.4%)
絶対利用したくない	0人 (0.0%)

ムで理解が下がってしまった学生がいる可能性は否定できない。「普段の説明と比べてどう思ったか」では普段の説明より良いと思っている人が多く、「自分のソースコードで利用したいか」では利用したいという人が多数いた。

6. おわりに

本研究ではプログラムを可視化する教育補助システムを開発した。アンケート結果より、本システムを利用することで初学者の理解促進が確認できた。今後、手続き型プログラミングだけでなく、オブジェクト指向プログラミングでもシステムを運用し、初学者の理解度の検証を行いたい。

また、本システムはJava標準ライブラリのArrayListなどオブジェクトを集約可能なものに対応しきれていない。今後の課題として、この問題を改善し、より利便性の高いシステムを目指したい。

参考文献

[1]Andy, O. and Greg, W. (著), 久野 禎子, 久野 靖 (訳): Making Software —エビデンスが変えるソフトウェア開発, pp.107-120, オライリー・ジャパン (2011)