

オブジェクト指向プログラミングの概念理解を 支援する AnchorGarden システムの評価

三浦 元喜^{1,a)} 杉原 太郎^{2,b)}

概要: プログラミング初学者がオブジェクト指向プログラミング言語 (OOPL) の特性である「型・変数・オブジェクトとデータ参照」の概念を正しく理解するのは一般に困難さを伴う。我々は視覚モデルへの直感的な操作から対応するソースコードを生成して提示する手法を提案し、インタラクティブシステム AnchorGarden を構築した。大学 2 年生の C 言語学習者を対象とした評価実験を行ったところ、システムを実際に操作したとき、操作しない場合に比べてソースコード実行後のオブジェクトの状況を正確に把握できるようになることを確認した。

Evaluation of AnchorGarden System that Helps Understanding of OOPL Concepts

MIURA MOTOKI^{1,a)} SUGIHARA TARO^{2,b)}

Abstract: Learning of object-oriented programming language (OOPL) is difficult for novice programmer. We proposed a method to visualize concept models of OOPL, and allow the novice programmer to manipulate the visualized models by intuitive manner. The novice programmer can see snippets of source code corresponding to the manipulation. We developed a system named Anchor Garden (AG) to realize the method, and conducted an experiment at the end of CS1 course. We confirmed the system significantly increase the understanding of object and variable relationship.

1. はじめに

一般に、プログラミングは初学者にとって敷居が高い。その理由として、理解すべき概念が複雑であること、概念の表記が特殊であることなどが挙げられる。KidSim[1] や Viscuit[2] といった例示ベースのビジュアルプログラミング環境は、これらの概念と表記を意識させず、誰でも複雑なアニメーションやゲームを記述できるように工夫されている。一方、汎用のコントロールはシステムに依存するものが多く、また一般的なプログラミング言語で用いられる概念の理解を目的とするものではない。ドリトル [3] や

Alice[4], Nigari system[5], Scratch[6]などはビジュアル表現を援用しプログラミング状態を可視化することで、プログラムの振る舞いを理解しやすくする工夫をとりいている。しかし実践的なオブジェクト指向プログラミングを行ううえでは、クラスとインスタンス、メンバ変数とメソッド、コンストラクタといった抽象度の高い概念の理解が必要になる。

通常、これらの概念を理解するためには、書籍等の説明を読み、実際にソースコードを編集してプログラムを作りながら振舞いを調べる必要がある。しかし、プログラムを真似ながら作ることはできても一般に振舞いを調べることは難しく、結果としてプログラムとその振舞いを対応付けて理解することが困難であることが多い。特に高抽象概念の理解に到るためには何度もソースコードを編集して、実行する試行錯誤が必要となる。

オブジェクト指向プログラミングに必要な概念や知識を

¹ 九州工業大学 基礎科学研究系
Faculty of Basic Science, Kyushu Institute of Technology
² 北陸先端科学技術大学院大学 知識科学研究科
School of Knowledge Science, Japan Advanced Institute of
Science and Technology
a) miuramo@mns.kyutech.ac.jp
b) sugihara@jaist.ac.jp

初学者が修得しやすくするための手法として、我々は視覚的に表現したモデルを初学者に提供し、初学者がそのモデルに対する操作を行うと、振舞いに対応したソースコードを提示することを提案した [7]。これをシステム化した Anchor Garden(以降、AG) では、型・変数・オブジェクトとデータ参照の概念について、上記の操作「体験」と、対応する出力コードの「観察」によって、円滑かつ良質な試行錯誤を提供する。利用者は基本的にマウス操作を行うのみで、ソースコードを記述せずにオブジェクト指向プログラミングの理解に資する概念を修得することが可能となる。我々は 2011 年に、オブジェクトのメンバ変数やメソッドの概念を追加し、より具体的な事例をもとに初学者が学習できるように改良を加えた [8] が、この一連の研究では統計的な有効性は十分検証されてこなかった。そこで我々は、プログラミング初学者を対象に、システムを利用した場合と利用しなかった場合とで事後テストの点数に差があるかどうかを調べ、手法およびシステムの有効性について検証する。

2. AnchorGarden システム

本章では、学習者に提示する視覚モデルとその体験、および、それによってどのような概念獲得を促進するかについて、具体的な操作手順の例とシステムの動作を追いながら説明する。今回は分数を表現するシンプルなクラス (Frac) の例をもちいて説明する。

システムを起動すると初期画面 (図 1) になる。ここで Type の Frac をクリックして選択する。次に中央の Variable 領域をクリックして Frac クラスの変数 frac1 を作成する。その後つづけて Object 領域を SHIFT キーを押しながらクリックして Frac クラスのオブジェクトを作成する。SHIFT キーを押しながらクリックすると、図 2 で示す画面が表示され、コンストラクタに与える引数を指定できる^{*1}。分数オブジェクト生成後は、変数 frac1 の右のリンクタブをドラッグして分数オブジェクトにドロップし、オブジェクトにリンクを張る。このように学習者は変数を作成したり、オブジェクトを作成したり、変数からオブジェクトへのリンクを張る作業を行うと、右のソースコード画面に対応するコードが随時表示され、モデルとプログラムの関係を知ることができる。

AnchorGarden システムでは、オブジェクトのメソッドを呼び出す際には、変数上で右クリックしてメニューを表示し、method public Frac mul_new(Frac) を選択する (図 3)。すると、図 4 のようにメソッドを呼び出す際に必要となる引数オブジェクトの選択画面となる。ドロップダウンリストから引数 (ここでは frac2) を選択して OK を押す。すると Object 領域にメソッド実行によって生成され

^{*1} SHIFT キーなしの単純なクリックのみの場合は、引数指定なしのデフォルトコンストラクタを利用する。

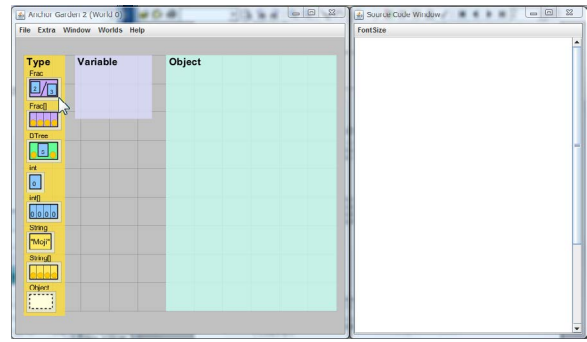


図 1 初期画面

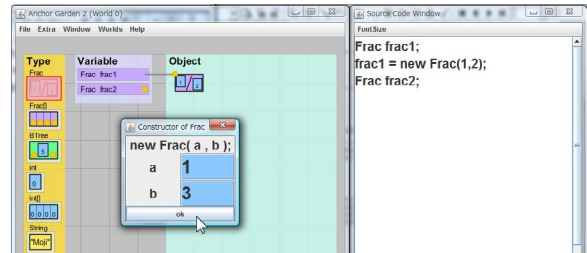


図 2 オブジェクトの生成

た、新しいオブジェクトが生成されるので、変数 frac3 からリンクする (図 5)。なお、new が付かない public void mul(Frac) を選択すると、メソッドを呼び出した分数オブジェクト (例えば frac1) の値が変更される。ほかには引数をとらないメソッド public void reduce() を呼び出すと、その分数オブジェクトの約分をおこなう。

この一連の操作と、右側の画面に出力されるソースコードを対比することによって、(1) 変数の宣言は型名+変数名で行うこと、(2) コンストラクタの呼び出しは new 型名 (引数) で行うこと、(3) 変数からのオブジェクトへの参照は = で行うこと、(4) メソッド呼び出しの表記と意味、(5) メソッド呼び出しの結果として、計算結果が入った「新しい分数オブジェクト」が生成される場合と、呼び出したオブジェクトの自分自身の値を直接書き換えるメソッドが存在すること、を体験的に理解できる。また、Frac クラスの具体的な定義を見ることで、コンストラクタや、メソッドにおける引数・返却値の定義はどのように行えばよいのかを例を通じて理解することが可能となる。

3. 実験

2012 年 8 月に、提案手法の有効性を検証するための実験を行った。工学部の 2 つの学科 (機械系の A 学科と、化学系の B 学科) に共通で開講している、C 言語の基礎を教える科目の最終回の授業において、発展としてオブジェクト指向プログラミングの基本的な考え方を 20 分程度でスライドおよび AnchorGarden を用いて説明した。具体的な説明内容を表 1 に示す。

説明の後、B 学科の学生にのみ、AnchorGarden を各自

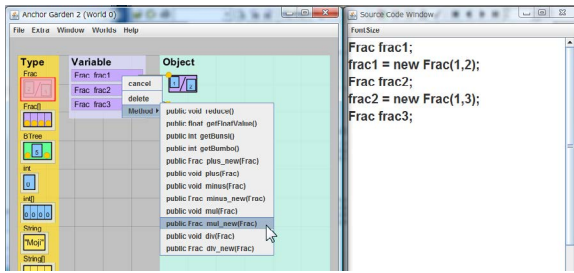


図 3 Variable 右クリックメニュー メソッド選択

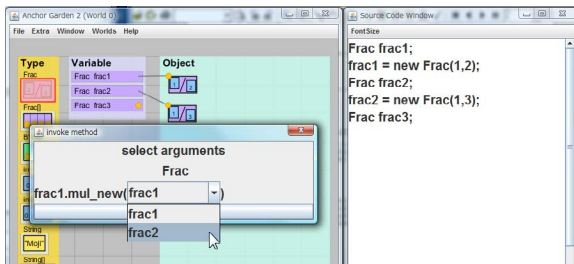


図 4 メソッド引数オブジェクトの選択

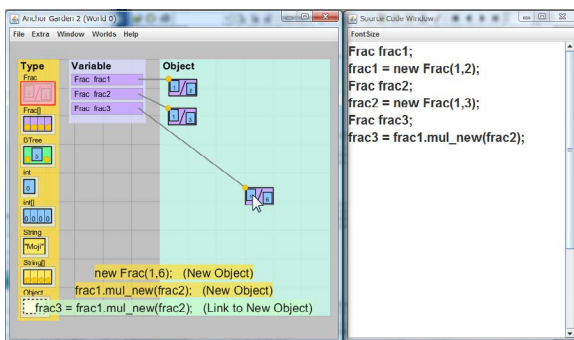


図 5 返却値のオブジェクトヘリンク

表 1 説明内容 (20 分程度)

Table 1 Lectured Contents (20 minutes)

説明概要	内容
C 言語の構造体とは	分数計算で例示
C 言語の構造体の問題点	分母=0 を抑止できない
ООP での型とは	クラス=構造体+手続き
ООP での型の具体例	分数クラス Frac
ООP での変数宣言と Object	変数と Object の分離
= の意味	変数から Object を参照
null の意味	変数からの参照を解除
new の意味	コンストラクタと新しい Object
メソッドの例 (1)	分数計算結果を新 Object で返す
メソッドの例 (2)	分数計算結果を自己書換え

起動し、操作説明を聞きながら試行する時間を 10 分程度与えた。その後、両学科とも A4 用紙の片面に印刷した問題を小テストとして解いてもらった。その際、小テストは講義全体の満点 100 点のうちの 2 点分の成績として加味することを伝えた。問題の回答時間は A 学科は 40 分、B 学科は 30 分とした。回答に際し、A 学科にはヒントとしてプロジェクトに AnchorGarden システムの画面を投影した。B 学科については各人が AnchorGarden システムを継続し

表 2 小テストの結果

Table 2 Scores of mini-test

	未使用群 (51 名)	使用群 (48 名)
問題 1 (3 点満点) 平均	2.88	2.96
標準偏差	0.32	0.20
問題 2 (5 点満点) 平均	3.31	4.33
標準偏差	1.82	1.28

て使用することを許諾した。

小テストの問題を付録に示す。問題 1 は分数クラスを使った計算結果を問う問題 3 問からなる。問題 2 は分数クラスを使ったプログラムの実行に伴うオブジェクトの状況を図示する問題 5 問からなる。

採点の結果、合計が 0 点の答えは回答意志が希薄であるとして以降の検定から除いた。最終的な有効回答数は A 学科 51 件、B 学科 48 件であった。問題ごとの結果を表 2 に示す。F 検定の結果、問題 1、問題 2 とともに等分散が仮定できなかった。分散が等しくないと仮定した t 検定の結果、問題 1 は有意差がみられなかった ($t(84) = 1.40, p = .16$)。一方、問題 2 については使用群の成績は未使用群に比べて有意に高いことがわかった ($t(90) = 3.24, p < .002$)。

問題 1 で有意差がみられなかった理由として、簡単な分数の計算ができればある程度の点数がとれる問題であったため、天井効果が現れたことが考えられる。問題 2 で有意差がみられた理由として、AnchorGarden システムの操作により、ソースコードの表す意味を短時間のうちに、直感的に理解することができたことが考えられる。なお AnchorGarden システムはソースコードの入力をサポートしていないため、問題 2 のようなソースコードの実行結果を直接確認することはできない。そのため、AnchorGarden を使用しても問題の解答を直接得ることはできない。それにもかかわらず未使用群との差が生じた理由として、オブジェクトの状況を操作して生成されるソースコードを観察することを通じ、学習者が概念モデルを確認・獲得することができたことが考えられる。

4. まとめと今後の課題

本研究では、初学者が視覚的に表現したモデルに対する操作を行うと、振舞いに対応したソースコードを提示する環境を利用した場合、プログラミングのソースコードを実行した際に計算機がどのように振舞うのかをより正確に認識できるようになることを実験により示した。その結果、本手法およびシステムは、短時間のうちにオブジェクト指向プログラミングで必要となる概念を修得するのに役立つことがあきらかとなった。

提案システムは <http://ist.mns.kyutech.ac.jp/miura/anchorgarden/> から実行可能である。

謝辞 本研究の一部は文部科学省科学研究費補助金(課

題番号 22650204) の支援によるものです。

参考文献

- [1] Smith, D. C., Cypher, A. and Spohrer, J.: KidSim: programming agents without a programming language, *Communication of the ACM*, Vol. 37, No. 7, pp. 54–67 (1994).
- [2] 原田康徳: Viscuit: 柔らかい書き換えによるアニメーション記述言語, 情報処理学会インタラクシオン 2004 予稿集, pp. 183–184 (2004).
- [3] 兼宗 進, 中谷多哉子, 御手洗理英, 福井眞吾, 久野靖: 初中等教育におけるオブジェクト指向プログラミングの実践と評価, 情報処理学会誌プログラミング, Vol. 44, No. 13, pp. 58–71 (2003).
- [4] Powers, K., Ecott, S. and Hirshfield, L. M.: Through the Looking Glass: Teaching CS0 with Alice, *Proceedings of the 38th SIGCSE technical symposium on Computer science education*, pp. 213–217 (2007).
- [5] 長 慎也, 甲斐宗徳, 川合 晶, 日野孝昭, 前島真一, 箕捷彦: プログラミング環境 Nigari: 初学者が Java を習うまでの案内役, 情報処理学会論文誌 プログラミング, Vol. 45, No. 9, pp. 25–46 (2004).
- [6] Maloney, J. H., Pepler, K., Kafai, Y., Resnick, M. and Rusk, N.: Programming by Choice: Urban Youth Learning Programming with Scratch, *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, pp. 367–371 (2008).
- [7] 三浦元喜, 杉原太郎, 國藤 進: オブジェクト指向言語における変数とデータの関係を理解するためのワークショップ, 情報処理学会論文誌, Vol. 50, No. 10, pp. 2396–2408 (2009).
- [8] 三浦元喜, 杉原太郎: オブジェクト指向言語におけるクラス定義の意味とオブジェクトの振舞いを理解するためのワークショップ, 情報処理学会情報教育シンポジウム (SSS2011), pp. 43–49 (2009).

付 録

問題 1 以下のプログラムを実行したら, 画面になにが表示されるか答えなさい

リスト 1 問題 1-1

```
Frac a,b,c; //変数の定義
a = new Frac(1,5);
b = new Frac(2,5);
c = a.plus_new(b);
c.print();
```

リスト 2 問題 1-2

```
Frac a,b; //変数の定義
a = new Frac(1,8);
b = a;
b.setBunsiBumbo(1,4);
a.print();
```

リスト 3 問題 1-3

```
Frac a,b,c; //変数の定義
a = new Frac(1,4);
b = new Frac(2,4);
c = b.plus_new(a);
a = c;
a.print();
```

問題 2 プログラムをすべて実行した直後の変数とオブジェクトの状況を図示しなさい

リスト 4 問題 2-1

```
Frac x,y,z;
x = new Frac(2,5);
y = new Frac(4,5);
z = a.plus_new(y);
```

リスト 5 問題 2-2

```
Frac m,k;
m = new Frac(3,8);
k = m;
k.setBunsiBumbo(2,9);
```

リスト 6 問題 2-3

```
Frac p,x,y;
p = new Frac(2,1);
x = new Frac(3,1);
y = x.plus_new(p);
x = y;
```

リスト 7 問題 2-4

```
Frac c,h,m;
c = new Frac(1,2);
h = c.plus_new(c);
m = h.plus_new(h);
```

リスト 8 問題 2-5

```
Frac a,b,c;
a = new Frac(2,3);
b = new Frac(3,2);
c = b.plus_new(a);
b = null;
```