

## 手続き型プログラムのオブジェクト指向化について

1J-5

鈴木 崇文\* 山本 晋一郎† 阿草 清滋‡

\* 名古屋大学工学研究科 † 愛知県立大学情報科学部 ‡ 名古屋大学情報メディア教育センター

## 1 はじめに

近年、ソフトウェアの規模は増大し複雑になってきている。このようなシステムを従来の方法で開発・保守することは困難になってきている。この問題を解決するために、オブジェクト指向による方法など、より効率的で扱いやすい方法が提案されている。これらの方法は、特に保守の労力を軽減するが、一般的に開発段階から運用されていないからではない。

一方で、過去に作られた膨大な量のソフトウェア資産が存在する。これらの多くは手続き型言語で書かれており、新しく提案された方法を十分に活用することができない。過去に開発された手続き型言語のプログラムをオブジェクト指向言語に変換することができれば、オブジェクト指向の技術を用いることによって、保守などの管理が行ないやすくなると思われる。

## 2 オブジェクト指向の特徴

オブジェクト指向の技術要素とその効果には、次の点が挙げられる。まず、オブジェクト化により情報隠蔽が行なえること、データとそれに対する操作をまとめて記述できるため、強度の強いモジュールを記述しやすいこと、継承によってプログラムの再利用が行ないやすいことなどである。

手続き型言語からオブジェクト指向言語へ変換するための指針として、ここではそれぞれの言語は構造上類似しており、また、データとそれを操作する部分をまとめて構造型を一般化したものがオブジェクトであると見なせる点を利用する。この変換により、オブジェクト指向へ変換できるが、そのためには、利用すべき情報を特定しなければならない。

## 3 手続き型プログラムのオブジェクト化

手続き型言語とオブジェクト指向言語を構成する要素の間には、ある程度の対応をとることができる。具体的には、型とクラス、手続きとメソッド、変数とインスタンス変数であり、これらは類似した機能を持っている。

Conversion from Procedural Program to Object Oriented Program.

Takafumi Suzuki\*, Shinichirou Yamamoto†, Kiyoshi Agusa\*

\*Nagoya University, †Aichi Prefectural University

基本的には、このような置換えによってオブジェクト指向のプログラムへ変換できるが、要素間の意味は異なるため、そのままの変換では適切なオブジェクトは得られない。まず、これらの間に存在する関係を調べ、これらのうち、考慮すべき関係を明らかにする。

## 3.1 考慮すべき関係

手続き型言語における手続きは、単に実行すべき処理を列挙したものである。しかし、何らかの処理が行なわれるときは必ず、何らかのデータが付随する。オブジェクト指向言語では、あるオブジェクトに属すデータに対しオブジェクト自身に属す手続きのみがデータを変更や参照可能なように制限することができる。このように、データとそのデータを操作する処理を明確にできるので、手続き型言語から関係の深い変数と手続きの組を抽出することにより、オブジェクト指向言語に変換することができる。

手続きとメソッドは、実際には必ずしも一対一になる必要はない。しかし、手続きを分割・統合して適切に構成することにより、メソッドと対応する手続きを作成できるので、ここでは、手続きをメソッドへの変換の単位として考える。

この場合、重要となる処理は、プログラム全体に影響を及ぼすような処理である。手続きの内部で行なわれた変更は、必ずしも、直接プログラム全体に影響が及ぶわけではない。例えば、局所変数を変更しても、その変更で直接依存して変化する処理は、手続きの内部に限られる。手続き内部の変化が、直接、外部に伝わるのは、次のようながいずれかの処理が存在する時である。

## ● 大域変数へ値を代入する場合

大域変数の場合、ある大域変数に値を代入している手続きはすべて、その大域変数を操作するための手続きであるとみなすことができる。したがって、代入している手続きと代入される変数をまとめることによって、ひとつのオブジェクトとすることができる。

## ● 仮引数に変数への参照を含み、その参照している先の値が変更される場合

仮引数を介して外部に手続き内部での変更が反映されるためには、引数が参照渡しまたは参照を含む構造型などであり、手続きの中で、その参照先

に書き込む処理があることが必要である。仮引数自体は手続き内部でのみ有効であり、実際に仮引数が外部に示している情報は、その引数に要求される型の情報である。

したがって、手続きが参照を含む仮引数を持ち、さらに参照先が変更を受ける場合、その手続きは仮引数の型に対する処理とみなせ、型をクラスに変換したオブジェクトのメソッドとして扱うことができる。

- 手続きの内部でデータが生成され、それが戻り値によって返される場合  
この場合でも、戻り値として渡される手続き内部の局所変数はやはり手続き内部でのみ有効である。したがって、手続きの戻り値を持つ情報は返す値の型である。つまり、この手続きは、返す型に対するオブジェクトのメソッドであるとみなすことができる。

以上で述べた、手続きと大域変数・型の間の関係を調べ、互いに関係を持つものをまとめることにより、オブジェクトをつくることができる。ひとつの手続きが複数の大域変数・型と関係をもつ場合や、ひとつの大域変数が多くの手続きと関係を持つことが起こる。それらは互いに関係を持つと考えられるので、基本的に、まとめてひとつのグループとして扱い、同一のオブジェクトに変換する。

### 3.2 オブジェクトへの変換方法

それぞれの場合について、手続き型言語から、オブジェクト指向言語への変換の方法を述べる。基本的に、上で述べた、関連する手続き・変数・型のグループに対し、それぞれひとつのオブジェクトを生成する。

大域変数をオブジェクトに変換する場合、もとの大域変数はクラスのインスタンス変数に変換する。そして、大域変数を変更している手続きを、そのクラスのメソッドとする。これにより、直接大域変数に変更を加える手続きはすべてクラス内のメソッドとなる。また、別の手続きの中で、大域変数を参照していた式は、クラスに対応するインスタンス変数の値を返すメソッドを作成し、このメソッドを介して値を参照するように変更する。

引数に関する関係をオブジェクトに変換する場合は次のようになる。まず、仮引数に表れていた変数をインスタンスとしてクラスに追加し、手続きは、メソッドとしてクラスに追加する。このとき、インスタンス変数に変換した仮引数は、手続きからは削除する。この仮引数を利用している部分は、直接、インスタンス変数を利用する式に変更する。また、別の手続きの中でこの手続きを呼び出していた式は、引数に渡していた変数の宣言を、このクラスのオブジェクトを生成する宣言に変更し、オブジェクトのメソッドを呼出す式に変換する。

表 1: 解析結果

プログラム	L	F	G	C	M	m
dhryostone-2.1	578	11	17	6	1.7	1.4
nkf-1.6	1582	16	40	3	2.0	1.0
gnugo-1.2	2857	20	75	3	4.0	1.0
gzip-1.2.4	7323	52	24	15	2.3	1.5

L: 行数, F: 手続きの数, G: 大域変数の数, C: 生成クラス数, M: 1 クラスの平均メソッド数, m: 1 クラスの平均インスタンス変数の数

戻り値に関する関係をオブジェクトにする場合も、引数の場合と同様に、その手続きをメソッドとしてクラスに追加する。そして、手続きの戻り値がそのクラスの实体を返すようにする。

それぞれの処理を各手続きに行なうことにより、手続き型のプログラムをオブジェクト指向のプログラムに変換することができる。

## 4 実装・評価

C 言語に対し、以上で示した変換方法を Sapid[1] を利用して実装した。現在、必要な関係を調べ、要素を分類をする部分まで実装されている。いくつかのプログラムに対し、このアルゴリズムを適用して解析を行ない、分類の状況を調べた。解析結果は表 1 のようになった。

この結果から、手続きと大域変数・型の関係を利用することにより、ある程度の判別が可能であるといえる。大多数のプログラムは大域変数を利用しており、関係抽出の基準とすることが可能である。しかし、大域変数をあまり持たないプログラムや定義された型を持たないプログラムでは、分類が困難である。また、全体的に、ひとつのグループに手続き・変数が偏りやすい。

## 5 まとめ

手続き型言語から保守しやすいオブジェクト指向言語へ変換するための方法について述べた。現在の問題点としては、前提として、ある程度適切に手続きが作られていなければならないこと、変換によって作られたオブジェクトがどの程度妥当かの判定が必要な点が挙げられる。また、ここでは手続きとデータの関係に基づいて変換を行なったが、これだけでは十分にオブジェクト指向になったとはいえない。より構造的な面での解析が必要だと考えられる。

## 参考文献

- [1] 山本晋一郎, 阿草清滋, “細粒度リポジトリに基づいたツール・プラットフォームとその応用,” 情報処理学会研究報告 95-SE-102, pp.37-42, 1995.