2M-8

# 初学者のオブジェクト指向プログラムの問題点の分析

渡辺大貴† 松浦佐江子†

芝浦工業大学大学院 理工学研究科 電気電子情報工学専攻†

## 1.はじめに

ソフトウェア開発では仕様変更による機能変更や機能追加は繰り返し行われるものであるが，後の保守性を考慮せずに機能変更や追加を行うとコードの複雑度が増大し，変更箇所の特定や変更自体が困難になることが多い．

本稿ではオブジェクト指向プログラムの初学者に多く見られるクラスの責務を考慮しない開発に着目する．クラスの責務を考慮しない開発とは，クラス内のフィールドとメソッドの関係を強固にし，他クラスとの結合をシンプルにすることをせずに，必要とされる処理をその処理を呼び出す側に手続き的な視点で定義する開発方法である．

こうした開発の改善方法を検討するために，本学科の2年次前期，後期，3年次前期に行われているオブジェクト指向プログラミングの演習課題を分析し，初学者の改善すべき問題点を分析する．

## 2.保守性とオブジェクト指向

保守性とは，ソフトウェア品質の評価に関する国際規格 ISO/IEC 9126[1]の6つの品質特性の中の一つであり，保守性が低下すると，機能変更時に変更すべき箇所を特定できない，変更箇所が多いことにより作業時間が増大する，機能変更を行ったが変更してない箇所からバグが発見される等の結果に繋がる．オブジェクト指向は本来，クラスという概念に基づき保守性を向上させる技術として登場したが，開発方式によってはこの特徴を活かせないことが多い．

本稿ではクラスの概念に基づいていない問題点を分析の観点とし，本学科のオブジェクト指向プログラミングの演習課題を分析する．

## 3.分析する観点

### 課題の出題方法

各授業において課題の出題方法が違うため，これを考慮した上で分析する必要がある．

・2年次前期

オブジェクト指向プログラミングⅠ演習 5 を分析対象とする．学生はクラス構成の仕様を与えられ，それに合わせてコードを構築する．

・2年次後期

プログラミング演習Ⅱテーマ 2 を分析対象とする．学生は出題された課題に対して自由にプログラムを構築する．

・3年次前期

情報実験Ⅰ第 4 回を分析対象とする．学生は出題された課題に対して自由にプログラムを構築する．

### 緩いアクセス飾子の数とカプセル化

カプセル化とは，特定のフィールドやメソッドを他クラスからアクセスされないようにし，不正な操作から守り，コードの複雑さを隠蔽することである．カプセル化をすることで，部品化・再利用性の向上につながり，修正・変更の影響範囲を極小化することができる． カプセル化はアクセス修飾子を用いることで実現することができる．基本的にフィールドを private にすることで保護し，フィールドにアクセスする場合には set・get メソッド(後述)と呼ばれるアクセッサメソッドを使用する．

初学者のプログラムでは，カプセル化が適切ではないプログラムが多いと仮説を立て，アクセス修飾子が緩いフィールド・メソッドの数を分析の観点とする．

本稿では，緩いアクセス修飾子を以下のように定義する．

・アクセス修飾子が public であるが，他クラスからアクセスされておらず，自クラスからのみアクセスされているメソッド・フィールド

・どこからもアクセスされていないメソッド・フィールド

### データクラス

データクラスとは，フィールドとコンストラクタ，set および get メソッド以外には何も持たないクラスのことである．set メソッドと get メソッドは以下のように定義する．

・set メソッド

引数をフィールドに代入するのみのメソッド

・get メソッド

フィールドの値を返り値とするメソッド

初学者は，クラスの設計時にクラスの責務を考えないことが多いと仮説を立て，データクラスの数とそのデータクラスの振る舞いにあたるコード片はどこにあるのかを分析の観点とする．

### クラスの結合度[2]

結合度とは，あるクラスが他クラスにどれだけ依存しているかを測るメトリクスである．あるクラスでのフィールド・メソッドの引数返り値・メソッド内の変数の型(プリミティブ型，クラス型を含む)の個数を結合度と定義する．

初学者は結合度の高いクラスを作るのではないかと仮説を立て，全ての期間を通してどのような推移になっているのかを分析の観点とする．

### Eclipse Metrics Plugin[3]

Eclipse Metrics Plugin とは，Eclipse で使用することのできるメトリクス測定用のプラグインである．このプラグインでは，23 種類のメトリクスを測定することができ，本稿では初学者のプログラムに関係あると思われる3種類のメトリクスを分析の観点とする．

・MaCabe Cyclomatic Complexity[4]

メソッドあたりの循環的複雑度である．MaCabe の循環的複雑度は，判定条件の数をπとすると，π+1 で求めることができる．ただし，複合条件(||や&&)は別条件とする．

このプラグインでは許容値が 20 と設定されており，それを越えると複雑なプログラムであると警告がでる．

・Nested Block Depth(以下 NBD)

メソッドあたりのブロックのネスト数である．ネスト

---

Analysis of Problems on Beginner's' Object-oriented Programs

† Daiki Watanabe † Saeko Matsuura

† Department of Electrical Engineering and Computer Science, Graduate School Of Engineering And Science, Shibaura Institute of Technology



Copyright ©2014 Information Processing Society of Japan.
All Rights Reserved.

とは入れ子のことであり，この数が大きいほど複雑なプログラムとなる．

・Lack of Cohesion of Methods(以下 LCOM)

あるクラスの凝集性の欠如を表す．0 に近いほど LCOM は大きく、クラスの強度が高いことを表す．Eclipse Metrics Plugin では Henderson-Sellers[5]の定義を用いている．計算式は以下の通りである．

$$LCOM = (<p> - |M|) / (1 - |M|)$$

ただし，

M =クラスのメソッドの集合
F =クラスのフィールドの集合
p(f) =F の要素であるフィールド f をしようしているメソッド数
<p> =(p(f)の合計)/|F|

とする．

## 4.プログラムの分析

### 4.1. 2 年次前期

2 年次前期では，データクラスをもつ学生が 2 人いた．データクラスの出現した学生のコードを見ると，データクラスの LCOM が 0.5 であり結合度が 1 である学生とデータクラスの LCOM が 0.75 であり結合度が 1 である学生であった．LCOM が高いためクラスの強度は低いと考えられる．また結合度が 1 であるため，依存している型は 1 つだけである．また，これらのデータクラスは他のクラスからの参照も少ない．このことから，これらのデータクラスは今後拡張する上で重要となってくる．データクラスに振る舞いをもたせれば，LCOM も 0 に近づくと考えられる．

循環的複雑度を見ると，最大値は 8 であった．この数値のメソッドを見ると似たようなコードが同じメソッド内に存在したため，これは 1 つのメソッドにまとめることができると考えられる．

NBD は，どのコードも 3 以下であったため，読みやすいコードであると考えられる．

### 4.2. 2 年次後期

2 年次後期では，データクラスをもつ学生がいなかった．LCOM から学生らのコードを見ると，LCOM が最大の学生は 1 であった．そのクラスを見ると，どこからも参照・更新されていないフィールドがあり，そのフィールドを削除した LCOM を見ると，LCOM は 0.5 となっていた．未使用のコードが存在すると，LCOM は増加すると思われる．他に LCOM が 1 である学生のコードがあったが，こちらも未使用のコードが存在し，削除すると LCOM は 0.25 に減少した．

循環的複雑度をみると，最大値は 14 であった．またこの数値のメソッドは NBD が 6 であり，他のメソッドの NBD に比べて高い数値であった．NBD が高いと循環的複雑度も高くなると考えられる．

### 4.3. 3 年次前期

3 年次前期では，データクラスをもつ学生が 5 人いた．どの学生の結合度も 2 であり，LCOM は 0(2 人)か 0.8(3 人)であった．LCOM が 0 の 2 人の学生のデータクラスを見ると，set メソッドをコンストラクタとして定義しており，フィールドのアクセス修飾子は省略されていた．これらのフィールドは他クラスから直接アクセスされており，カプセル化されていない保守性の低いコードとなっている．また，データクラスのフィールドは他クラスで参照され，演算などの処理を行った結果を再びデータクラスのフィールドへ代入している．このような処理が存在した場合，その処理をデータクラスへ移動することで，保守性の高いクラスになると考えられる．

循環的複雑度を見ると，最大値は 42 であった．この数値のメソッドが含まれるクラスは Panel クラスを継承しており，GUI の一部となっている．リスナーを使用する上で条件判断が複雑になり，循環的複雑度が高くなっていた．

NBD は，最大が 5 であったため，そのコードは読みにくいと考えられる．

### 4.4. 全体の考察

2 年次前期はクラス構成が指示されるため，それに機能変更を加える授業形式となっている．学生は 1 からコードを構築する必要がないため，問題点がでにくいと考えられる．2 年次後期になると課題だけが与えられ，学生は自由にコードを構築できる．そのためどこが問題点なのかがわからなくなり，各数値が上昇したと考えられる．3 年次前期では 2 年次後期までオブジェクト指向を学習しているが，規模が大きいため自分のコードを整理しきれず，各数値が上昇したと考えられる．

学年が上になるにつれてコードの規模が大きくなるため，循環的複雑度の数値は上昇している．規模が大きいため，学生は自分の書いたコードが整理できなくなり複雑になると思われる．

2 年次前期と 3 年次前期のデータクラスを比べると，3 年次前期はデータクラスのフィールドを参照して演算し代入する処理がないのに対して，2 年次前期はそのような処理が存在しない．データクラスは処理を持たせることで保守性を高めることができるが，初学者にはどの処理をデータクラスに持たせればいいのかわからないため，それを指摘すればデータクラスは改善されると思われる．

## 5.まとめと今後の方針

初学者は，コードの規模が大きくなるにつれて循環的複雑度と NBD が増加し，複雑で読みにくいコードを書くことがわかった．NBD は，ブロックとなっている処理を自クラスの private メソッドとして定義することで，読みやすくなると考えられる．

データクラスは，LCOM が低く結合度が低いため，改善する必要がある．このクラスを使用している処理をメソッドとしてデータクラスに付加させることが改善策として考えられる．なお，低い学年ではこのような処理が存在しないため，この段階では改善する必要がないと思われる．

今後はこのような観点から改善手法を検討し，開発の中で問題点を改善できないかを考えていく．

## 6.参考文献

[1] ISO/IEC 9126-1:2001，http://www.iso.org/iso/
[2] Efferent Couplings http://eclipse-metrics.sourceforge.net/descriptions/pages/EfferentCouplings.html
[3] Eclipse Metrics Plugin http://eclipse-metrics.sourceforge.net/
[4] Thomas J. McCabe，“A Complexity Measure，”IEEE Transactions On Software Engineering，pp. 308-321，Dec 1979
[5] Lack of Cohesion of Methods (Henderson-Sellers) http://eclipse-metrics.sourceforge.net/descriptions/pages/cohesion/HendersonSellers.html



Copyright ©2014 Information Processing Society of Japan.
All Rights Reserved.