

LB-3 オブジェクト指向入門教育における逸脱パターンの提示による効果  
Effects of Presenting Deviation Patterns in Introductory Object-Oriented Course

鹿糠 秀行† Hideyuki Kanuka 三部 良太†† Ryota Mibe 矢島 敬士††† Hiroshi Yajima

1. はじめに

近年 Web システムを中心として、オブジェクト指向(OO)を活用した情報システム開発が増えてきている。それに伴い、大学や企業のソフトウェア教育において、OO に関して教える機会が増えてきており、OO の教育法に関する研究の重要性も増してきている。

Pedagogical Patterns Project (<http://www.pedagogicalpatterns.org>) では、主に OO の教育法に関する問題とその解決法を、教育パターンとして収集している。教育パターンの一つである[2]では、OO 上級コースにおいて、過去の優れた OO 設計の成果であるデザインパターン(DP)[1]を活用した、OO の教育法を提案している。

我々は、OO 入門コースにおいても、DP のわかりやすい例と我々が提案している逸脱パターン(2 節で説明)[3]を提示し、それらの利用法を通じて、抽象化・継承・ポリモルフィズムなどの OO 特有で初心者には理解することが難しい事柄を、教授者は効果的に教えることができ、学習者の理解促進に役立つと考える。

本稿では、このアプローチの有用性を確認するために、OO に関しては初心者の学生数名に対して実験を行い、OO 入門教育における DP の教授と逸脱パターンの提示による効果について検討をする。

2. 逸脱パターンについて

DP の不完全な理解や誤った解釈のために、DP の作成意図からの逸脱を意味するような、DP の適用や拡張をしてしまいがちである。これは、DP の利点を台無しにするだけでなく、バグの原因に繋がる可能性もある。

特に初心者が、[1]から逸脱を読み取ることは、非常に難しいと考えている。そこで我々は、予め起こりうるだろう DP の作成意図からの逸脱とそれに対する解決策や防止策を考え、アンチパターン[4]の形式で整理した「逸脱パターン」を提案した[3]。

DP の例として、[1]で取り上げられている 23 個のパターンの一つである Template Method パターン(図 1)を、概説する。この DP は、親クラス(抽象クラス AbstractClass)の TemplateMethod でアルゴリズムの流れだけを予め定義し、その中で使われる具体的な処理内容(HookMethod)についてはサブクラス(ConcreteClass)で定義し、ポリモルフィズムを通じて動的に処理内容を決定することができる。この利点を応用して、OO フレームワークを構築するためによく利用されている重要な DP である。

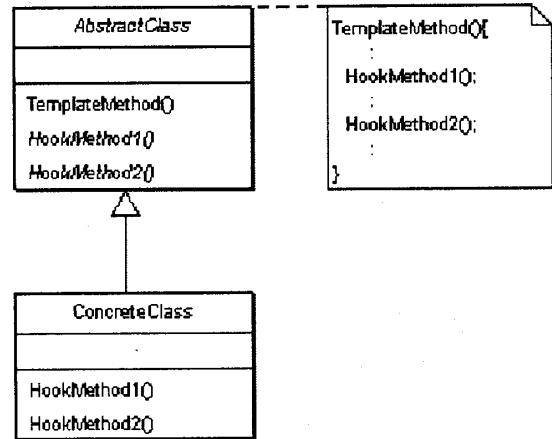


図 1 : Template Method パターンのクラス図

我々は現在までに、DP に対する逸脱パターンのいくつかを作成してきた。Template Method パターンの逸脱パターンの一つを、表 2 に例として示す。

(<http://www.ntt.dis.titech.ac.jp/home/kanuka/dp/top.html>)にて、他の逸脱パターンを公開している。

3. 実験

3.1 実験の目的

OO 初心者に対して DP の教授と演習、同時に逸脱パターンの提示とその解説を通じて、OO に関係する事柄や DP 自体の理解促進への効果を確認する。

3.2 実験の内容

被験者は、OO プログラミングの初歩を知っている程度の理工系大学院生 5 名である。実験に用いる DP として、2 節でも紹介した Template Method パターンを選んだ。

まず、被験者に DP の概要と実験の手順を説明する。次に、DP カタログ[1]と DP のわかりやすい例としての解説書[5]、そして Template Method パターンの逸脱パターンを資料として渡し、これらを活用して各自 Template Method パターンを利用する任意のプログラムを作成してもらった。プログラミング終了後、一旦アンケート(表 1)に 5 段階(5 点が最高)で解答してもらった。

表 1 : 実験アンケート

	質問内容
Q1	資料[1]をどのぐらい活用したか
Q2	資料[5]をどのぐらい活用したか
Q3	Template Method パターンを理解できたか
Q4	OO の事柄についての理解が深まったか (特に抽象クラスと継承の活用法)
Q5	逸脱パターンによって何らかの効果が得られたか

† 東京工業大学大学院 総合理工学研究科 知能システム科学専攻, CISS, IGSSE, Tokyo Institute of Technology.  
 †† (株)日立製作所 システム開発研究所, Systems Development Laboratory, Hitachi, Ltd.  
 ††† (株)日立製作所 研究開発本部, Research & Development Group, Hitachi, Ltd.

表2：逸脱パターンの例

逸脱パターン名	テンプレートメソッドの再定義
対象デザインパターン	Template Method パターン
逸脱タイプ	構造
一般形式	ConcreteClass で TemplateMethod() を再定義
症状と結果	TemplateMethod() でアルゴリズムの流れを固定するのが Template Method パターンの意図である。ConcreteClass で Template Method() をオーバーライドすることは、アルゴリズムの流れを変える行為であり、Template Method パターンの目的からの逸脱を意味する。
解決策	1. TemplateMethod() のアルゴリズムを再検討する。 2. TemplateMethod() のアルゴリズムごとに変更したいのであれば、Strategy パターンへの変更を考える。
防止策	TemplateMethod() のオーバーライドをさせないようにする。例えば、Java では AbstractClass で定義される TemplateMethod() を final 宣言することでサブクラス(ConcreteClass)での再定義を禁止することができる。
逸脱例	省略
関連する逸脱パターン名	省略
備考	省略

次に、現在までに確認している Template Method パターンに対する5つの逸脱パターンの解説と其中で様々なDPやOOに関係する事柄を説明した。

例えば、表2の逸脱パターンの解説を通じて次のようなことを確認した。なぜ TemplateMethod() の再定義が逸脱行為なのかを説明することによって、Template Method パターンの意図を再確認した。また、逸脱の解決策の項目2を通じて、Strategy パターンの概要と Template Method パターンとの違いを解説し、さらにそれに関連して継承と集約の違いを再確認した。そして、防止策を通じて Java の final 宣言の活用法とその必要性を確認した。

最後に再度、アンケートのQ3～Q5を質問した。

#### 4. 実験結果及び検討

実験アンケート(表1)の結果を図2に示す。縦軸は、5人の被験者の平均点とする。

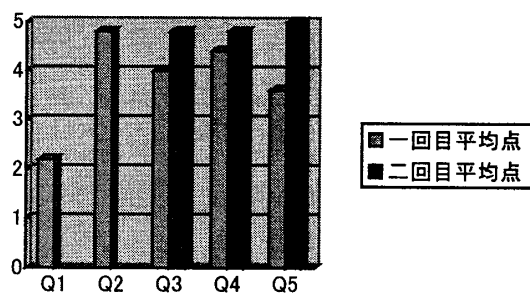


図2：アンケート結果

Q1とQ2の結果より、初心者にとって[1]は理解することが難しかったようであり活用せず、ほとんどの人が解説書[5]を主に活用してDPを理解し、全員がJavaを用いて実装した。作成した各自のプログラムをチェックしたが、一人を除いて[5]のサンプルプログラムに基づいて作成していたために、逸脱を含むプログラムは無かった。

Q3とQ4の一回目の結果より、逸脱パターンを説明する前に、[5]とプログラム作成を通じて、Template Method パターンとOOの事柄(特に抽象クラスと継承の活用法)に関して、すでにある程度の理解が得られたようである。

Q5の一回目の結果より、初心者にとって、実験で用いた逸脱パターンの説明記述が、独力で理解するには少々難しかったようで、その効果までを全員が十分に感じることはできなかったようである。逸脱パターンの説明記述については、図や具体例を充実させるなどして、可読性を向上させることが、今後の課題である。

しかし、解説後に行った二回目のアンケートのQ5では、全員が満点と解答しており、確かにDPやOOに関する事柄の理解が深まったということは、Q3とQ4の結果やその後の被験者へのヒアリングからも確認している。

#### 5. むすび

本稿では、OO初心者に対する、DPの教授と演習、我々が提案している逸脱パターンの提示と解説を通じて、OO特有で初心者には理解することが難しい事柄を、教授者は効果的に教えることができ、学習者の理解促進に役立つということを、実験とその結果より確認することができた。

今後、どのDPとその逸脱パターンを、どのようにOO入門教育へ取り入れることが、さらに効果的であるかを、検討する必要があると考えている。

#### 参考文献

- [1] Gamma, E., Helm, R., Johnson, R., Vlissides, J. (監訳; 本位田真一, 吉田和樹): "オブジェクト指向における再利用のためのデザインパターン", ソフトバンクパブリッシング(1995).
- [2] Jane Hung: "Using Design Patterns as a Context in which to teach Fundamental Components of Object Oriented Systems", <http://sol.info.unlp.edu.ar/ppp/pp28.htm>
- [3] 鹿糠秀行, 三部良太, 矢島敬士: "ソフトウェアパターンの適用における逸脱パターンの提案", 第137回情報処理学会ソフトウェア工学研究会, pp33-39, (2002.5.10).
- [4] William J. Brown, Raphael C. Malveau, William H. Brown, Hays W., III McCormick, and Thomas J. Mowbray: "AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis", WILEY, (1998).
- [5] 結城浩: "Java言語で学ぶデザインパターン入門", ソフトバンクパブリッシング, (2001).