

オブジェクト指向設計

原則（2）

オブジェクトモデリングスペシャリスト

土屋 正人

Masato Tsuchiya

m-tsuchi@sra.co.jp

前号では、ロバート・C・マーチンの「アジャイルソフトウェア開発の奥義」で取り上げられている 11 のオブジェクト指向原則から「単一責任の原則 (SRP)」を紹介しました。今回は「オープン・クローズドの原則 (OCP)」を紹介します。

◆オープン・クローズドの原則

OCP (Open-Closed Principle) と呼ばれます。定義は、「ソフトウェアの構成要素 (クラス、モジュール、関数など) は拡張に対して開いて (オープン) いて、修正に対して閉じて (クローズド) いなければならない」です。

オブジェクト指向言語 Eiffel の設計者であるバートランド・メイヤの名著「オブジェクト指向入門」にモジュール性に関する 1 章があります。その中でメイヤは、**モジュール性が高いことを評価するための 5 つの基準と 5 つの原則**を説明しています。5 つの基準とは次の通りです。

- ・ モジュールの分解しやすさ
- ・ モジュールの組み合わせやすさ
- ・ モジュールのわかりやすさ
- ・ モジュールの連続性
- ・ モジュールの保護性

モジュールの連続性とは「仕様に変更を加えた時にその変更が 1 つあるいは 2~3 のモジュールにとどまるか」ということです。とどまる場合は「その手法はモジュール

の連続性を満たしている」ということになります。

モジュールの保護性とは「実行時にあるモジュールの中で発生した異常の影響がそのモジュールだけ、あるいは周辺の 2~3 のモジュールだけにとどまるか」ということです。とどまる場合は「その手法はモジュールの保護性を満たしている」ということになります。また **5 つの原則**とは、次の通りです。

- ・ 言語としてのモジュール単位
- ・ 少ないインタフェース
- ・ 小さいインタフェース
- ・ 明示的なインタフェース
- ・ 情報隠蔽

言語としてのモジュール単位とは「使用する言語はモジュール性をサポートするものでなければならない」ということ。少ないインタフェースとは「モジュールはできるだけ少ないモジュールと対話すべき」、小さいインタフェースとは「モジュール間でやり取りする情報量はできる限り少なくなければならない」ということで、構造化設計の結合度と同じです。さらに、5 つの基準の 1 つである「モジュールの分解しやすさ」を満たすためにはもうひとつの原則を満たさなければならない、として、開放／閉鎖原則をあげています。これは、モジュールは拡張に対して開いて (Open) おり、修正に対して閉じて (Closed) いなければならない、というもので、要は、

- ・ 拡張容易性
- ・ 修正局所化

を同時に満たすことがオープン・クローズド原則 (OCP) です。オブジェクト指向の効果についていろいろな意見がありますが、メイヤは、

- ・ 継承などを使って拡張容易性を満たすことができる
- ・ カプセル化によって修正局所化ができる

というように OCP を満たせる点が他の手法に比べて優れていることとしています。オブジェクト指向の真髓がこ

ここにあると言えるでしょう。

◆リファクタリング

前号でもふれましたが、書き上げたコードを改良する手段としてリファクタリングがあります。リファクタリングとは「外から見えるふるまいを変えずに、ソフトウェアを分かりやすくし、安いコストで変更できるようにするためにソフトウェアの内部構造に加えられる変更」あるいは、これを行なうことです。設計原則を満たすコードを書くことは大切ですが、最初から完璧を目指すのではなく、早めにコーディングに着手して動く成果物を作成し、それをリファクタリングによって改良するアプローチがアジャイルに開発を進める上で現実的かと思います。

マーチン・ファウラ等の著書「リファクタリング」は、オブジェクト設計原則を知らなくとも原則を遵守するようにコードを導いてくれるものです。OCP を満たすためには、前号でも取り上げた「オブジェクト間でのメンバの移動」というカテゴリにあるものが「修正局所化」の実現のために有効です。これらは「拡張容易性」の実現にもつながりますが、ここでは「一般化の処理」というカテゴリを紹介します。このカテゴリには、リファクタリングカタログには次のように命名されたリファクタリングがあります。

- ・ メソッドの上位階層への移動
- ・ メソッドの下位階層への移動
- ・ モジュールの抽出
- ・ モジュールのインライン化
- ・ サブクラスの抽出
- ・ 継承の導入
- ・ 階層構造の統合
- ・ テンプレートメソッドの作成
- ・ 継承から委譲へ
- ・ 委譲から継承へ
- ・ 抽象クラスからモジュールへ

「一般化」は UML では「汎化」と呼ばれています。汎化によって必ず「拡張容易性」が手に入るわけではありませんが、ひとつの手段として有効です。例えば、テンプレートメソッドの作成は、同じようなステップを同じ順序で実行するメソッドが別々のサブクラスにあるがステップに微妙な違いがある、といった事実が動機となり、

- ・ サブクラスの該当するメソッドから、違いのあるステップを抽出する
- ・ 抽出したステップ群を、それぞれメソッドにしてシグネチャを同じものにする
- ・ 元のメソッドではそのメソッドを呼び出すようにしてスーパークラスに移す
- ・ 新たにサブクラスに作ったメソッドのシグネチャ部のみをスーパークラスに移す

といった施策をとることで「拡張容易性」を実現できるように設計実装を改良していきます。「テンプレートメソッド」とは、多相性(多態性)を利用したデザインパターンとして知られているものです。ただし、汎化のために継承というメカニズムを使うと、クラス間の構造が強固になるため、かえって修正がやりにくくなる危険性があります。そのような場合は「継承から委譲へ」を使うことで継承のデメリットを回避することができます。

前述したように「拡張容易性」と「修正局所化」を「同時に」実現できることが、オブジェクト指向の真髄だと思います。OCP は設計原則の中でも特に重要なものと言えるでしょう。

◆◆◆ ミュージカル A COMMON BEAT ◆◆◆

コンサルタントファシリテータ 野島勇

ファシリテータ修行ということでミュージカルに出演します。え？ミュージカル？と思われるかもしれませんね。コモンビートは、情熱と感受性を育て創造活動を通じ互いに学び合う「共育（共に育む）」活動を行う、教育・人材育成のNPO法人です。日本を元気にしよう！という団体です。そのNPOの第15期100人100日プログラムに参加しています。7/30, 31に公演がありますので、ぜひお越しください！続きはWebで！

<http://www.commonbeat.org/musical/15.html>

夢を。



GSLetterNeo Vol. 22

2010年5月20日発行

発行者 ●株式会社 SRA 産業開発統括本部

編集者 ●土屋正人、柳田雅子、小嶋勉、野島勇

ご感想・お問い合わせはこちらへお願いします ●gsneo@sra.co.jp

株式会社SRA

〒171-8513 東京都豊島区南池袋2-32-8

夢を。Yawaraka Innovation
やわらかいのべーしょん