

オブジェクト指向の本質に基づいた  
アプリケーション設計  
～品質、コスト、改修を意識した設計～

# 目次

## Chapter I オブジェクト指向開発の現状

|       |                     |      |
|-------|---------------------|------|
| I-i   | オブジェクト指向開発の現状       | I-3  |
| 1.    | 広がるオブジェクト指向言語       | I-3  |
| 2.    | オブジェクト指向開発に関する誤解    | I-4  |
| 3.    | 非オブジェクト指向開発の蔓延      | I-5  |
| 4.    | オブジェクト指向設計の失敗事例     | I-6  |
| 5.    | オブジェクト指向設計の失敗事例の修正  | I-7  |
| 6.    | マーティン・ファウラーの嘆き      | I-8  |
| I-ii  | 現状の打破               | I-9  |
| 1.    | 振る舞いを重視する           | I-9  |
| 2.    | オーケストラモデルからの離脱      | I-10 |
| 3.    | アンサンブルモデルの採用        | I-11 |
| I-iii | そもそもオブジェクト指向とは      | I-12 |
| 1.    | オブジェクト指向とは          | I-12 |
| 2.    | 誰にメリットのある技術なのか      | I-13 |
| 3.    | オブジェクターの育成          | I-14 |
| 4.    | オブジェクター育成の観点        | I-15 |
| 5.    | オーケストラモデルとアンサンブルモデル | I-16 |

## Chapter II 問題のあるクラス設計例

|         |                        |       |
|---------|------------------------|-------|
| II-i    | 本章の位置付け                | II-3  |
| 1.      | 本章の目的                  | II-3  |
| 2.      | 本章の目標                  | II-4  |
| II-ii   | クラス設計の失敗例: 巨大な何でもクラス   | II-5  |
| 1.      | 巨大な何でもクラス              | II-5  |
| 2.      | 巨大な何でもクラスのデメリット        | II-6  |
| II-iii  | クラス設計の失敗例: 深すぎる継承      | II-8  |
| 1.      | 深すぎる継承                 | II-8  |
| 2.      | 深すぎる継承のデメリット           | II-9  |
| II-iv   | クラス設計の失敗例: 似て非なるクラス    | II-10 |
| 1.      | 似て非なるクラスの乱立            | II-10 |
| 2.      | 似て非なるクラスのデメリット         | II-11 |
| II-v    | クラス設計の失敗例: メンバ変数の乱用    | II-12 |
| 1.      | メンバ変数の乱用とは             | II-12 |
| 2.      | メンバ変数の乱用のデメリット         | II-13 |
| II-vi   | クラス設計の失敗例: ロジックの漏れ     | II-14 |
| 1.      | ロジックの漏れ                | II-14 |
| 2.      | ロジックの漏れのデメリット          | II-15 |
| II-vii  | クラス設計の失敗例: インタフェースの不揃い | II-16 |
| 1.      | インタフェースの不揃い            | II-16 |
| 2.      | インタフェースの不揃いのデメリット      | II-17 |
| II-viii | クラス設計の失敗例: クラス設計失敗例の棚卸 | II-18 |

# 目次

## Chapter III オブジェクト指向の基礎知識

|                           |         |
|---------------------------|---------|
| III-i 本章の位置付け             | III- 3  |
| 1. 本章の目的                  | III- 3  |
| 2. 本章の位置付け                | III- 4  |
| III-ii クラス設計とは            | III- 5  |
| 1. クラス設計の考え方              | III- 5  |
| 2. クラスの役割(ロール)を決める        | III- 6  |
| III-iii クラス、オブジェクト、インスタンス | III- 7  |
| 1. オブジェクトとは               | III- 7  |
| 2. クラス                    | III- 8  |
| 3. インスタンス                 | III- 9  |
| <<参考>>システム化対象領域とクラス       | III- 10 |
| III-iv カプセル化              | III- 11 |
| 1. カプセル化の必要性 その1          | III- 11 |
| 2. カプセル化の必要性 その2          | III- 12 |
| 3. カプセル化                  | III- 13 |
| 4. カプセル化のメリット             | III- 14 |
| 5. カプセル化の本質               | III- 15 |
| III-v 汎化と特化               | III- 16 |
| 1. 汎化                     | III- 16 |
| 2. 特化                     | III- 17 |
| 3. オーバーライド                | III- 18 |
| <<参考>>汎化によるメリット           | III- 19 |
| 4. クラスの階層構造               | III- 20 |
| 5. 抽象クラス                  | III- 21 |
| III-vi ポリモフィズム            | III- 22 |
| 1. 汎化/特化におけるオブジェクトの抽象的な扱い | III- 22 |
| 2. ポリモフィズムとは              | III- 23 |
| III-vii クラス間の関係           | III- 25 |
| 1. クラス間の関係とは              | III- 25 |
| 2. 関連                     | III- 26 |
| 3. 集約                     | III- 28 |
| 4. コンポジション集約              | III- 29 |
| 5. 依存                     | III- 30 |

# 目次

## Chapter IV クラス設計の原則

|                              |        |
|------------------------------|--------|
| IV-i 本章の概要                   | IV- 3  |
| 1. 本章の目的                     | IV- 3  |
| 2. 本章の位置付け                   | IV- 4  |
| IV-ii クラス設計の基本原則             | IV- 5  |
| 1. Open-Close Principalとは    | IV- 5  |
| 2. Open-Close Principalのポイント | IV- 6  |
| 3. 拡張に対しオープンな設計とは            | IV- 7  |
| 4. 修正に対しクローズな設計とは            | IV- 8  |
| IV-iii クラス分割の原則              | IV- 9  |
| 1. 1 fact in 1 place とは      | IV- 9  |
| 2. 1 fact in 1 place の実現     | IV- 10 |
| IV-iv システム構成要素を基にしたクラス分割     | IV- 11 |
| 1. システム構成要素を基にしたクラス分割        | IV- 11 |
| 2. 境界に着目したクラス分割の着眼点          | IV- 12 |
| 3. 境界に着目したクラス分割              | IV- 13 |
| 4. システム構成要素とその境界によるクラス分割     | IV- 14 |
| 5. 多重度に着目したクラス分割             | IV- 15 |
| IV-v 特化によるクラスの拡張             | IV- 16 |
| 1. 特化と「is～a」関係               | IV- 16 |
| 2. 特化の原則に基づくクラス拡張            | IV- 17 |
| 3. 特化の深さに関する原則               | IV- 18 |
| 4. 抽象クラスによるインスタンス作成制限        | IV- 19 |
| IV-vi 集約を用いたクラスの拡張方法         | IV- 20 |
| 1. 集約を用いたクラスの拡張方法            | IV- 18 |
| 2. 集約の原則に基づくクラス拡張            | IV- 21 |
| 3. 集約によるクラス拡張と委譲             | IV- 22 |
| IV-vii フローズンスポットとホットスポットの分離  | IV- 23 |
| 1. フローズンスポット/ホットスポットとは       | IV- 23 |
| 2. フローズンスポット/ホットスポットの分離      | IV- 24 |
| IV-viii 操作の外部仕様の設計           | IV- 25 |
| 1. 操作の外部仕様とは                 | IV- 25 |
| 2. 操作の外部仕様の統一                | IV- 26 |
| 3. 外部仕様が統一された操作の呼び出し         | IV- 27 |
| 4. 変更に強い操作の外部仕様              | IV- 28 |
| IV-ix ドメインのクラス群の重視           | IV- 30 |
| 1. 振る舞いを重視する                 | IV- 30 |
| 2. 振る舞いのモデルを作成する             | IV- 31 |
| IV-x デザインパターンの利用             | IV- 33 |
| 1. デザインパターンとは                | IV- 33 |
| 2. Template Method パターンの利用   | IV- 34 |