

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.	非オブジェクト指向開発の蔓延	I-4
3.	オブジェクト指向開発に関する誤解	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	そもそもオブジェクト指向とは	I-9
1.	オブジェクト指向とは	I-9
2.	誰にメリットのある技術なのか	I-10
I-iii	現状の打破	I-11
1.	現状を打破するには	I-11
2.	オブジェクターになるには	I-12
3.	振る舞いを重視する	I-13
4.	オーケストラモデル	I-14
5.	アンサンブルモデル	I-15
6.	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	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
3. オブジェクト指向とは	III- 5
4. オブジェクト指向のメリット	III- 6
5. クラス	III- 8
6. インスタンス	III- 9
7. カプセル化	III- 11
8. 汎化と継承	III- 15
9. 抽象クラス	III- 21
10. 継承関係におけるオブジェクトの抽象的な扱い	III- 22
11. ポリモフィズム	III- 23
III-ii 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. 依存	III- 30
III-iii UML	III- 31
1. UMLとは	III- 31
2. ユースケース図	III- 32
3. シーケンス図	III- 38
4. クラス図	III- 41
5. ステートマシン図	III- 43
III-iv クラス設計の概要	III- 45
1. クラス設計の考え方	III- 45
2. クラスのロール(役割)	III- 46

目次

ChapterIV 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.	拡張に対しオープンな設計とは	IV- 6
3.	修正に対しクローズな設計とは	IV- 7
IV-iii	クラス分割の原則	IV- 8
1.	1 fact in 1 place とは	IV- 8
2.	1 fact in 1 place の実現	IV- 9
IV-iv	システム構成要素を基にしたクラス分割	IV- 10
1.	システム構成要素を基にしたクラス分割	IV- 10
2.	境界に着目したクラス分割	IV- 11
3.	多重度に着目したクラス分割	IV- 13
IV-v	継承によるクラスの拡張	IV- 14
1.	継承の原則に基づくクラス拡張	IV- 14
2.	継承の深さに関する原則	IV- 16
3.	抽象クラスによるインスタンス作成制限	IV- 17
IV-vi	集約によるクラスの拡張	IV- 18
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- 19
3.	集約によるクラス拡張と委譲	IV- 20
IV-vii	コールドポイントとホットポイントの分離	IV- 21
1.	コールドポイント／ホットポイントとは	IV- 21
2.	コールドポイント／ホットポイントの分離	IV- 22
IV-viii	操作の外部仕様の設計	IV- 23
1.	操作の外部仕様とは	IV- 23
2.	操作の外部仕様の統一	IV- 24
3.	外部仕様が統一された操作の呼び出し	IV- 25
4.	変更に強い操作の外部仕様	IV- 26
IV-ix	ドメインのクラス群の重視	IV- 28
1.	振る舞いを重視する	IV- 28
2.	振る舞いのモデルを作成する	IV- 29
IV-x	デザインパターンの利用	IV- 31
1.	デザインパターンとは	IV- 31
2.	Template Method パターンの利用	IV- 32

目次

Chapter V 正しいクラスの分析・設計技法

V-i	開発工程の全体像	V - 3
	1. オブジェクト指向開発の流れ	V - 3
	2. 例題について	V - 4
V-ii	要件定義	V - 5
	1. フレームワークの選定	V - 5
	2. ユースケースの抽出	V - 6
V-iii	概要設計	V - 8
	1. ユースケースの詳細化	V - 8
	2. オブジェクトの抽出のポイント	V - 11
	3. オブジェクトの抽出	V - 12
	4. シナリオの分析	V - 14
	5. 状態遷移の分析	V - 14
	6. オブジェクトの分析	V - 19
V-iv	詳細設計	V - 18
	1. クラスの整理	V - 21
	2. 詳細シーケンス図の作成	V - 25
	3. 詳細ステートマシン図の作成	V - 28
	4. 詳細クラス図の作成	V - 29
	5. クラスおよび操作の詳細化	V - 30
V-v	実装	V - 32
	1. ソースコードの記述	V - 32
V-vi	テスト	V - 33
	1. 単体テスト〜クラス単位のテスト	V - 33
	2. 単体テスト〜層単位のテスト	V - 34
	3. 単体テスト〜各層間の疎通確認	V - 35
	4. 結合テスト〜ユースケース単位のテスト	V - 36
	5. 結合テスト〜ユースケース間のテスト	V - 37
	6. システムテストと運用テスト	V - 38