

序文 .....	7
謝辞 .....	9
はじめに .....	11
本書について .....	11
謝辞 .....	12
イントロダクション .....	21
頭文字 M .....	22
本書について .....	29
想定読者 .....	33

## 第1部 メタプログラミング Ruby 35

第1章 月曜日：オブジェクトモデル .....	37
1.1 ビルと一緒の月曜日 .....	37
1.2 オープンクラス .....	38
1.2.1 クラス定義の中身 .....	40
1.2.2 オープンクラスの問題点 .....	42
1.2.3 猿マネとモンキーパッチ .....	43
1.3 クラスの真実 .....	44
1.3.1 オブジェクトの中身 .....	45
1.3.2 クラス再訪 .....	47
1.3.3 定数 .....	49
1.3.4 オブジェクトとクラスのまとめ .....	53
1.3.5 もう1つの学習機会 .....	54
1.4 クイズ：引かれていない線 .....	55
1.4.1 クイズの答え .....	56
1.5 メソッドを呼び出すときに何が起きているの？ .....	57
1.5.1 メソッド探索 .....	58

1.5.2	メソッドの実行	62
1.6	クイズ：絡み合ったモジュール	66
1.6.1	クイズの答え	67
1.7	オブジェクトモデルのまとめ	69
<b>第2章</b>	<b>火曜日：メソッド</b>	<b>71</b>
2.1	重複問題	72
2.1.1	レガシーシステム	72
2.1.2	ダブル、トリプル、トラブル	73
2.2	動的メソッド	74
2.2.1	メソッドを動的に呼び出す	75
2.2.2	メソッドを動的に定義する	78
2.2.3	Computer クラスのリファクタリング	79
2.3	method_missing()	83
2.3.1	method_missing() のオーバーライド	84
2.3.2	ゴーストメソッド	84
2.3.3	動的プロキシ	87
2.3.4	Computer クラスのリファクタリング (再び)	91
2.4	クイズ：バグ退治	94
2.4.1	クイズの答え	95
2.5	もっと method_missing()	96
2.5.1	メソッド名が衝突したら	96
2.5.2	まとめ	102
<b>第3章</b>	<b>水曜日：ブロック</b>	<b>105</b>
3.1	水曜日の過ごし方	106
3.1.1	今日の指針	106
3.1.2	基本に立ち返る	106
3.2	クイズ：Ruby#	108
3.2.1	using キーワード	108
3.2.2	挑戦	109
3.2.3	クイズの答え	110
3.3	クロージャ	110
3.3.1	スコープ	112
3.3.2	スコープのフラット化	116
3.3.3	スコープのまとめ	119
3.4	instance_eval()	119
3.4.1	カプセル化の破壊	121

3.4.2	クリーンルーム	122
3.5	呼び出し可能オブジェクト	122
3.5.1	Proc オブジェクト	123
3.5.2	Proc 対 lambda	126
3.5.3	メソッド再び	129
3.5.4	呼び出し可能オブジェクトのまとめ	130
3.6	ドメイン特化言語を書く	131
3.6.1	はじめての DSL	131
3.7	クイズ：より良い DSL	133
3.7.1	ビルスの逃亡	134
3.7.2	クイズの答え	135
<b>第4章</b>	<b>木曜日：クラス定義</b>	<b>139</b>
4.1	クラス定義のわかりやすい説明	140
4.1.1	クラス定義の中身	140
4.1.2	カレントクラス	141
4.1.3	カレントクラスのまとめ	143
4.1.4	クラスインスタンス変数	144
4.1.5	Bookworm の作業再び	146
4.2	クイズ：クラスのタブー	148
4.2.1	クイズの答え	148
4.3	特異メソッド	149
4.3.1	特異メソッドの導入	150
4.3.2	クラスメソッドの真実	152
4.3.3	クラスマクロ	153
4.4	特異クラス	155
4.4.1	特異メソッドの謎	155
4.4.2	特異クラスの出現	157
4.4.3	メソッド探索再び	159
4.5	クイズ：モジュールの不具合	167
4.5.1	クイズの答え	168
4.6	エイリアス	170
4.6.1	メソッドエイリアス	171
4.6.2	アラウンドエイリアス	172
4.6.3	Amazon の問題の解決	175
4.7	クイズ：壊れた計算	176
4.7.1	クイズの答え	176
4.7.2	一日の終わり	177

<b>第5章 金曜日：コードを記述するコード</b> .....	<b>179</b>
5.1 道案内 .....	179
5.1.1 上司の依頼 .....	180
5.1.2 開発計画 .....	181
5.2 Kernel#eval .....	181
5.2.1 Capistrano の例 .....	182
5.2.2 irb の例 .....	185
5.2.3 「コード文字列」対「ブロック」 .....	187
5.2.4 eval() の問題点 .....	188
5.3 クイズ：属性のチェック（手順1） .....	193
5.3.1 このクイズに答える前に .....	194
5.3.2 クイズの答え .....	195
5.4 クイズ：属性のチェック（手順2） .....	196
5.4.1 クイズの答え .....	196
5.5 クイズ：属性のチェック（手順3） .....	198
5.5.1 クイズの答え .....	199
5.6 クイズ：属性のチェック（手順4） .....	199
5.6.1 クイズの答え .....	200
5.7 フックメソッド .....	201
5.7.1 その他のフックメソッド .....	202
5.7.2 Merb の例 .....	204
5.7.3 クラス拡張ミックスイン .....	205
5.8 クイズ：属性のチェック（手順5） .....	206
5.8.1 クイズの答え .....	208
5.8.2 メタプログラミング道 .....	208
<b>第6章 エピローグ</b> .....	<b>209</b>

## 第2部 Rails におけるメタプログラミング 211

<b>第7章 ActiveRecord の設計</b> .....	<b>213</b>
7.1 観光ツアーの準備 .....	214
7.1.1 Rails のインストール .....	214
7.1.2 Rails のソースコード .....	215
7.2 ActiveRecord の設計 .....	216
7.2.1 ActiveRecord の簡単な例 .....	217
7.2.2 ActiveRecord::Base .....	218

7.2.3 ActiveRecord::Validations .....	219
7.2.4 alias_method_chain() .....	222
7.2.5 ActiveRecord::Base を最後にもう一度 .....	224
7.3 学んだ教訓 .....	225
7.3.1 Java から離れる .....	225
7.3.2 モジュールで考える .....	227
7.3.3 自分のやり方でやる .....	228

## 第8章 ActiveRecord の中身 229

8.1 動的属性 .....	230
8.1.1 ActiveRecord::Base#method_missing() .....	230
8.1.2 ゴースト属性の転生 .....	232
8.1.3 アクセサの定義 .....	233
8.1.4 動的なままの属性 .....	235
8.1.5 ActiveRecord::Base#respond_to?() .....	237
8.2 動的ファインダ .....	238
8.2.1 ActiveRecord::Base.method_missing() .....	239
8.2.2 ActiveRecord::Base.respond_to?() .....	243
8.3 ここで学んだこと .....	243
8.3.1 パフォーマンスを気にしすぎない .....	243
8.3.2 自分なりの境界線を引く .....	245

## 第9章 安全なメタプログラミング 247

9.1 メタプログラミングのテスト .....	247
9.1.1 コントローラのフィルタ .....	248
9.1.2 コントローラフィルタのテスト .....	252
9.2 モンキーパッチを防ぐ .....	255
9.2.1 モンキーパッチを明示的にする .....	256
9.2.2 モンキーパッチの防止 .....	258
9.3 学んだこと .....	261
9.3.1 あなたのメタプログラミングのコードをテストする .....	261
9.3.2 メタプログラミングもプログラミング .....	262

## 第3部 付録 263

<b>付録A よく使うイディオム</b> .....	<b>265</b>
A.1 ミミックメソッド .....	265
A.1.1 Camping の例 .....	267

A.2	nil ガード	268
A.3	メソッド引数のトリック	269
A.3.1	名前付き引数	269
A.3.2	配列引数とデフォルト値	271
A.3.3	引数のイディオムを組み合わせる	271
A.3.4	Rails の find() の例	272
A.4	自己 yield	273
A.4.1	RubyGems の例	273
A.4.2	tap() の例	274
A.5	Symbol#to_proc()	275
<b>付録B</b>	<b>ドメイン特化言語</b>	<b>277</b>
B.1	ドメイン特化言語の事例	277
B.1.1	DSL を使う	278
B.2	内部 DSL と外部 DSL	279
B.3	DSL とメタプログラミング	280
<b>付録C</b>	<b>魔術書</b>	<b>281</b>
C.1	魔術	281
C.1.1	配列引数	281
C.1.2	アラウンドエイリアス	282
C.1.3	ブランクスレート	282
C.1.4	クラス拡張	283
C.1.5	クラス拡張ミックスイン	283
C.1.6	クラスインスタンス変数	284
C.1.7	クラスマクロ	284
C.1.8	クリーンルーム	284
C.1.9	コードプロセッサ	285
C.1.10	コンテキスト探査機	285
C.1.11	遅延評価	286
C.1.12	動的ディスパッチ	286
C.1.13	動的メソッド	286
C.1.14	動的プロキシ	287
C.1.15	フラットスコープ	287
C.1.16	ゴーストメソッド	288
C.1.17	フックメソッド	288
C.1.18	カーネルメソッド	289
C.1.19	遅延インスタンス変数	289

C.1.20	ミミックメソッド	289
C.1.21	モンキーパッチ	290
C.1.22	名前付き引数	290
C.1.23	ネームスペース	290
C.1.24	nil ガード	291
C.1.25	オブジェクト拡張	291
C.1.26	オープンクラス	291
C.1.27	パターンディスパッチ	292
C.1.28	サンドボックス	292
C.1.29	スコープゲート	293
C.1.30	自己 yield	293
C.1.31	共有スコープ	294
C.1.32	特異メソッド	294
C.1.33	コード文字列	295
C.1.34	Symbol の Proc 変換	295

<b>付録D</b>	<b>参考書</b>	<b>297</b>
------------	------------	------------

<b>付録E</b>	<b>から騒ぎ</b>	<b>299</b>
------------	-------------	------------

E.1	Null オブジェクト	301
E.2	Ruby の強みを活かす	302
E.3	Null オブジェクトの向こう側	303

<b>著者・訳者紹介</b>	<b>305</b>
----------------	------------

索引	306
----	-----