

# 目次

---

日本語版によせて	v
訳者まえがき	vii
Effective Java によせて	xiii
まえがき	xv
謝辞	xix
第1章 はじめに	1
第2章 オブジェクトの生成と消滅	5
項目1 コンストラクタの代わりに static ファクトリーメソッドを検討する . . . . .	5
項目2 数多くのコンストラクタパラメータに直面した時にはビルダーを検討する . .	11
項目3 private のコンストラクタが enum 型でシングルトン特性を強制する . . . . .	17
項目4 private のコンストラクタでインスタンス化不可能を強制する . . . . .	19
項目5 不必要なオブジェクトの生成を避ける . . . . .	20
項目6 廃れたオブジェクト参照を取り除く . . . . .	24
項目7 ファイナライザを避ける . . . . .	27
第3章 すべてのオブジェクトに共通のメソッド	33
項目8 equals をオーバーライドする時は一般契約に従う . . . . .	33
項目9 equals をオーバーライドする時は、常に hashCode をオーバーライドする . .	45
項目10 toString を常にオーバーライドする . . . . .	51
項目11 clone を注意してオーバーライドする . . . . .	54
項目12 Comparable の実装を検討する . . . . .	62
第4章 クラスとインタフェース	67
項目13 クラスとメンバーへのアクセス可能性を最小限にする . . . . .	67
項目14 public のクラスでは、public のフィールドではなく、アクセッサメソッド を使う . . . . .	71
項目15 可変性を最小限にする . . . . .	73
項目16 継承よりコンポジションを選ぶ . . . . .	80
項目17 継承のために設計および文書化する、でなければ継承を禁止する . . . . .	86

項目 18	抽象クラスよりインタフェースを選ぶ	91
項目 19	型を定義するためだけにインタフェースを使用する	96
項目 20	タグ付クラスよりクラス階層を選ぶ	98
項目 21	戦略を表現するために関数オブジェクトを使用する	101
項目 22	非 static のメンバークラスより static のメンバークラスを選ぶ	104
第 5 章	ジェネリックス	107
項目 23	新たなコードで原型を使用しない	107
項目 24	無検査警告を取り除く	113
項目 25	配列よりリストを選ぶ	116
項目 26	ジェネリック型を使用する	121
項目 27	ジェネリックメソッドを使用する	126
項目 28	API の柔軟性向上のために境界ワイルドカードを使用する	131
項目 29	型安全な異種コンテナを検討する	138
第 6 章	enum とアノテーション	143
項目 30	int 定数の代わりに enum を使用する	143
項目 31	序数の代わりにインスタンスフィールドを使用する	153
項目 32	ビットフィールドの代わりに EnumSet を使用する	154
項目 33	序数インデックスの代わりに EnumMap を使用する	156
項目 34	拡張可能な enum をインタフェースで模倣する	160
項目 35	命名パターンよりアノテーションを選ぶ	164
項目 36	常に Override アノテーションを使用する	170
項目 37	型を定義するためにマーカーインタフェースを使用する	173
第 7 章	メソッド	175
項目 38	パラメータの正当性を検査する	175
項目 39	必要な場合には、防衛的にコピーする	178
項目 40	メソッドのシグニチャを注意深く設計する	182
項目 41	オーバーロードを注意して使用する	184
項目 42	可変長引数を注意して使用する	190
項目 43	null ではなく、空配列か空コレクションを返す	194
項目 44	すべての公開 API 要素に対してドキュメントコメントを書く	196
第 8 章	プログラミング一般	203
項目 45	ローカル変数のスコープを最小限にする	203
項目 46	従来の for ループより for-each ループを選ぶ	206
項目 47	ライブラリーを知り、ライブラリーを使う	209
項目 48	正確な答えが必要ならば、float と double を避ける	212
項目 49	ボックスされた基本データより基本データ型を選ぶ	214

---

項目 50	他の型が適切な場所では、文字列を避ける . . . . .	217
項目 51	文字列結合のパフォーマンスに用心する . . . . .	220
項目 52	インタフェースでオブジェクトを参照する . . . . .	221
項目 53	リフレクションよりインタフェースを選ぶ . . . . .	223
項目 54	ネイティブメソッドを注意して使用する . . . . .	226
項目 55	注意して最適化する . . . . .	227
項目 56	一般的に受け入れられている命名規約を守る . . . . .	230
第 9 章	例 外 . . . . .	233
項目 57	例外的状態にだけ例外を使用する . . . . .	233
項目 58	回復可能な状態にはチェックされる例外を、プログラミングエラーには実行 時例外を使用する . . . . .	236
項目 59	チェックされる例外を不必要に使用するのを避ける . . . . .	238
項目 60	標準例外を使用する . . . . .	240
項目 61	抽象概念に適した例外をスローする . . . . .	242
項目 62	各メソッドがスローするすべての例外を文書化する . . . . .	244
項目 63	詳細メッセージにエラー記録情報を含める . . . . .	246
項目 64	エラーアトミック性に努める . . . . .	248
項目 65	例外を無視しない . . . . .	250
第 10 章	並行性 . . . . .	251
項目 66	共有された可変データへのアクセスを同期する . . . . .	251
項目 67	過剰な同期は避ける . . . . .	256
項目 68	スレッドよりエグゼキューターとタスクを選ぶ . . . . .	262
項目 69	wait と notify よりコンカレンシーユーティリティを選ぶ . . . . .	264
項目 70	スレッド安全性を文書化する . . . . .	269
項目 71	遅延初期化を注意して使用する . . . . .	272
項目 72	スレッドスケジューラに依存しない . . . . .	276
項目 73	スレッドグループを避ける . . . . .	278
第 11 章	シリアライズ . . . . .	279
項目 74	Serializable を注意して実装する . . . . .	279
項目 75	カスタムシリアライズ形式の使用を検討する . . . . .	285
項目 76	防御的に readObject を書く . . . . .	291
項目 77	インスタンス制御に対しては、readResolve より enum 型を選ぶ . . . . .	297
項目 78	シリアライズされたインスタンスの代わりに、シリアライズ・プロキシを検討 する . . . . .	302
付 録	初版に対応する項目 . . . . .	307

参考文献	309
索 引	315