

目 次

第1章 はじめに	1
1.1 インターネット環境でのプログラミングと Java	1
1.2 大規模プログラミングとオブジェクト指向	2
1.3 仮想マシン	2
1.4 さらなる進化	5
1.5 準備 — ソフトウェアなどの入手 —	5
第2章 オブジェクトの生成とメソッド呼び出し	7
2.1 オブジェクトとクラス	7
2.2 最初の例題 — オブジェクトの生成とメソッド呼び出し —	8
2.3 コンパイルと実行	11
2.4 TurtleFrame と Turtle の API の仕様	12
2.5 次の例題 — プリミティブ値, コンストラクタ, ライブラリの利用 —	14
2.6 インポート宣言	17
2.7 さらにもう一つの例題 — インスタンス変数 —	18
2.8 オブジェクトとは?	22
第3章 処理の流れ	24
3.1 for 文による繰り返し	24
3.1.1 同じことを繰り返す	24
3.1.2 繰り返しをカウントする変数値の利用	27
3.1.3 繰り返しのネスト	28
3.2 while 文による繰り返し	30
3.3 論理演算子	30
3.4 if 文	32
3.5 break 文 と continue 文	33
3.6 switch 文	35
第4章 クラス変数とクラスメソッド	36
4.1 クラス	36

4.2	クラス変数	37
4.3	クラスメソッド	37
4.4	Java API に現れるクラス変数, クラスメソッド	39
4.4.1	Java API の利用	39
4.4.2	javafx.scene.paint.Color クラス	40
4.4.3	Math クラス	40
4.4.4	System クラス	42
4.4.5	String クラス	43
第5章	クラスの作成	44
5.1	メソッド	44
5.1.1	メソッドの追加	44
5.1.2	メソッドと段階的なプログラムの開発	48
5.1.3	サブクラスと継承	49
5.1.4	メソッドのオーバーロード (多重定義)	49
5.2	インスタンス変数	50
5.3	クラス変数とクラスメソッド	53
5.4	パッケージとアクセス修飾子	54
5.4.1	パッケージ	54
5.4.2	アクセス修飾子	55
5.4.3	カプセル化	56
5.5	コンストラクタ	57
5.6	内部クラス	59
5.7	まとめの問題: グラフの描画	60
第6章	配 列	64
6.1	配 列	64
6.2	初期値をもった配列	67
6.3	可変長引数	69
6.4	拡張された for 文	69
6.5	配列の配列	70
6.5.1	多次元配列	70
6.5.2	配列の配列	71
6.6	Arrays クラス	72
6.7	main の引数	73
6.8	まとめの問題: 折れ線の描画	74

第7章 プリミティブ型とラッパークラス	76
7.1 プリミティブ型	76
7.2 演算子	77
7.2.1 数の演算とキャスト	77
7.2.2 変数への代入とキャスト	78
7.2.3 式と文	79
7.2.4 変数の値を変化させる演算子	80
7.2.5 演算子の優先順位	81
7.3 ラッパークラス	81
7.4 列挙型	83
7.5 ガーベッジコレクション	83
7.6 クラスパス	84
第8章 再帰呼び出しと例外処理	85
8.1 メソッド呼び出しとスタック	85
8.2 再帰呼び出し	86
8.2.1 再帰的なメソッド定義	86
8.2.2 木のフラクタルの描画	88
8.2.3 自己相似図形の描画	89
8.3 例外処理	93
8.3.1 例外	93
8.3.2 try-catch 文	94
第9章 メソッドのオーバーライドとインターフェース型	97
9.1 メソッド呼び出しと型検査	97
9.1.1 型検査	97
9.1.2 クラスと型の関係	98
9.1.3 メソッドのオーバーロードの解決	99
9.1.4 クラス型のキャスト	99
9.2 メソッドのオーバーライド（再定義）	101
9.2.1 メソッドのオーバーライドとメソッド探索	101
9.2.2 例：Tensen クラスでの fd の再定義	101
9.2.3 super メソッド呼び出し	102
9.2.4 オーバーライドされたメソッドの呼び出し	103
9.2.5 動的結合	103
9.2.6 スーパークラスで定義されたメソッドからの呼び出し	105
9.2.7 Object クラスのメソッドのオーバーライド	107
9.3 フィールドの隠蔽	108

9.4	抽象メソッドと抽象クラス	108
9.5	インターフェース	109
9.5.1	インターフェース型	109
9.5.2	インターフェースを実装したクラス	109
9.5.3	インターフェース型の変数を用いたプログラム	110
9.5.4	インターフェースのその他のメンバー	112
第10章 ラムダ式と関数型インターフェース		113
10.1	ラムダ式と関数	113
10.2	ラムダ式を引数や返値にするメソッド	116
10.3	ラムダ式によるローカル変数のキャプチャ	117
10.4	メソッド参照	119
10.5	java.util.function パッケージの関数型インターフェース	119
第11章 コレクションフレームワーク		121
11.1	コレクションフレームワーク	121
11.2	List インターフェースと ArrayList クラス	122
11.3	LinkedList クラス	126
11.4	イテレータ	127
11.5	Set インターフェースと SortedSet インターフェース	128
11.6	Map インターフェースと HashMap クラス	129
11.7	Comparator とソート	132
11.8	コレクションと配列の変換	133
11.9	ワイルドカード	134
11.10	まとめの問題: 線画編集プログラム	136
第12章 マルチスレッドと並行処理		137
12.1	スレッド	137
12.1.1	Thread クラスと Runnable インターフェース	137
12.1.2	ラムダ式とスレッド	139
12.2	sleep と割り込み	141
12.2.1	join による同期	141
12.2.2	sleep メソッドと割り込み	141
12.2.3	割り込みとスレッドの強制終了	142
12.3	タスクとエグゼキューターサービス	143
12.3.1	エグゼキューターサービス	144
12.3.2	Callable インターフェースと Future オブジェクト	145
12.3.3	invokeAny と invokeAll	146

12.4	スレッド間の同期	148
12.4.1	synchronized による排他的実行	148
12.4.2	変数の共有とデッドロック	149
12.4.3	アトミック変数	150
12.5	スレッドの協調動作	151
12.5.1	CountDownLatch	151
12.5.2	CyclicBarrier	153
12.5.3	Semaphore (セマフォ)	154
12.5.4	wait と notify	155
12.6	コンカレントコレクション	157
12.6.1	同期化コレクション	157
12.6.2	コンカレントコレクション	157
12.6.3	ConcurrentHashMap	158
12.6.4	CopyOnWriteArrayList	158
12.6.5	BlockingQueue	159
12.7	ストリーム	160
12.7.1	ストリームによる宣言的なデータ変換の記述	160
12.7.2	並列ストリーム	163

第13章 GUI クラス 165

13.1	GUI クラス	165
13.1.1	GUI クラスの役割	165
13.1.2	GUI ライブラリの種類	166
13.1.3	JavaFX のクラス	167
13.1.4	UI コンポーネントクラスのプロパティ	170
13.2	JavaFX アプリケーションの基本	171
13.2.1	枠組み	171
13.2.2	画像の表示	173
13.3	UI コンポーネントの配置とスタイル	175
13.3.1	概要	175
13.3.2	Region クラスのプロパティ	176
13.3.3	レイアウトコンテナ内のノードの大きさと位置合わせ	177
13.3.4	FlowPane クラス	179
13.3.5	BorderPane クラス	180
13.3.6	VBox クラス	181
13.3.7	HBox クラス	183
13.3.8	GridPane クラス	184
13.4	CSS によるスタイルの指定	186

13.4.1	JavaFX CSS の概要	186
13.4.2	スタイルの指定場所	186
13.4.3	スタイル属性の値	188
13.4.4	CSS ファイルとその指定	191
13.5	UI コンポーネントの API	192
13.5.1	ラベル : Label	192
13.5.2	ボタン基本抽象クラス : ButtonBase	193
13.5.3	Button	194
13.5.4	CheckBox	195
13.5.5	RadioButton	196
13.5.6	ComboBox	197
13.5.7	TextField	198
13.5.8	メニュー	199
13.5.9	Chart	200

第 14 章 イベント処理 203

14.1	イベント処理の仕組み	203
14.1.1	概要	203
14.1.2	イベント処理の流れ	204
14.1.3	EventHandler の設定	206
14.1.4	イベントの情報を知るメソッド	208
14.2	アクションイベント処理	209
14.2.1	Button 上のアクションイベント	209
14.2.2	CheckBox 上のアクションイベント	210
14.2.3	TextField 上のアクションイベント	212
14.2.4	MenuItem 上のアクションイベント	214
14.3	マウスイベント処理	215
14.4	キーイベント処理	217
14.5	プロパティ値の変化に応じたイベント処理	218
14.6	プロパティ値へのバインディング	221
コラム	JAR ファイルとアプリケーションの実行	223

第 15 章 グラフィックス 224

15.1	描画の概要	224
15.1.1	Shape クラスと Canvas クラス	224
15.1.2	座標	225
15.1.3	座標変換	225
15.2	Shape クラス	226

15.2.1	概要	226
15.2.2	Shape オブジェクトへのイベントの設定	229
15.3	Canvas クラス	235
15.3.1	図形描画メソッド	235
15.3.2	Canvas オブジェクトへのイベントの設定	239
15.4	アニメーション	241
15.4.1	概要	241
15.4.2	Transition のサブクラスによるアニメーション	241
15.4.3	Timeline によるアニメーション	243
15.4.4	AnimationTimer によるアニメーション	246
15.4.5	音声の再生	250
コラム	アプリケーションの Web ブラウザからの実行	252

第16章 データ入出力 253

16.1	データ入出力の概要	253
16.1.1	io パッケージ	253
16.1.2	ストリームクラス	253
16.1.3	データ入出力の基本	256
16.2	文字ストリームの使用例	257
16.2.1	ファイルを1文字ずつ読む	257
16.2.2	ファイルを1行ずつ読む	259
16.2.3	キーボード入力から文字データを読む	260
16.3	書式付き入力	262
16.3.1	Scanner クラス	262
16.3.2	区切り文字による入力の解析	263
16.3.3	パターンを使った入力の解析	265
16.4	書式付き出力	266
16.4.1	書式付き出力メソッド	266
16.4.2	書式指示子	267
16.4.3	標準出力への書式付き出力	268
コラム	リソースファイルの指定	269

第17章 ネットワークインターフェース 270

17.1	ネットワークプログラミングの基本知識	270
17.1.1	サーバとクライアント	270
17.1.2	IP アドレスとホスト名	271
17.1.3	ポート番号	271
17.1.4	ソケット	271

17.2	クライアントプログラム	272
17.3	サーバプログラム	275
17.3.1	サーバ側のソケット	275
17.3.2	カウンセラーサーバ	276
17.3.3	カウンセラークライアント	279
17.4	マルチクライアント	281
17.5	URLを使ったデータの取得	285
17.5.1	URL クラス	285
17.5.2	URL からのデータ読み込み	285

索 引	289
------------	------------