謝辞

2011年10月、スティーブ・ジョブス氏が亡くなりました。氏は、1996年アップル社に復帰するや、パソコンiMac、デジタル音楽プレーヤiPodを矢継ぎ早に開発し、倒産寸前のアップル社を見事に再建しました。そして、2007年に世紀の商品iPhoneを発表し、コンピュータの歴史にその名を刻みました。

スティーブ・ジョブス氏は毀誉褒貶なかばする人物でした。それは氏のヒューマンスキルの軽視によるもので、氏の意識は常にイノベーションの重視と、技術と芸術への複眼的視座にありました。いつの日かイノベーションを忘れ、ヒューマンスキルだけが目立つようになった私たち日本人は、今こそ氏の生きざまに刮目すべきなのかもしれません。

優れた技術は一社によって統制されるべきとする氏の遺志を受け継いだアップル社に対し、オープンソースの御旗を掲げて戦いを挑む「Android」陣営は、アップル社シェア40%に対し、「Android」陣営47%（スマートフォン2011/3）と善戦しています。しかし、アップル社の優位には変わりがなく、「Android」傘下の企業には息のぬけない日々が続いています。私も、本書の著者として、この戦いの今後の展開と帰趨を注意深く見守っていきたいと思っています。

最後にAndroidアプリ開発についての貴重なノウハウを伝授してくださった㈱ネットワーク応用技術研究所高木宏社長、同社研究員緒方卓実さま、本書執筆の機会を与えてくださった技術評論社さま、数々の助言や励ましをくださった池本さん、雑駁な原稿に怯まずりっぱな書籍にしあげてくださった編集部に厚く御礼申し上げます。

2012年7月　著者しるす

●免責

本書に記載された内容は、情報の提供だけを目的としています。したがって、本書を用いた運用は、必ずお客様自身の責任と判断によって行ってください。これらの情報の運用の結果について、技術評論社および著者はいかなる責任も負いません。

本書記載の情報は、2012年6月現在のものを掲載していますので、ご利用時には、変更されている場合もあります。

また、ソフトウェアに関する記述は、特に断わりのないかぎり、2012年7月現在でのバージョンをもとにしています。ソフトウェアはバージョンアップされる場合があり、本書での説明とは機能内容や画面図などが異なってしまうこともあり得ます。本書ご購入の前に、必ずバージョン番号をご確認ください。

以上の注意事項をご承諾いただいた上で、本書をご利用願います。これらの注意事項をお読みいただかずに、お問い合わせいただいても、技術評論社および著者は対処しかねます。あらかじめ、ご承知おきください。

●商標、登録商標について

・本書に登場する製品名などは、一般に各社の登録商標または商標です。なお、本文中に™、®などのマークは特に記載しておりません。

はじめに

コンピュタリゼーションは、メインフレーム、ネットワーク、端末、の3つの柱によって支えられ、これらの技術のイノベーションによって発展してきました。

メインフレームの中核は回路の集積化技術です。ICに始まりLSI、VLSI、ULSIと高度化してきた技術は、フロアを占拠したメインフレームを本箱サイズに縮小し、1京FLOPSの計算をこなすスーパーコンピュータを出現させました。

ネットワークのエポックは、光通信技術による高速広帯域通信と通信プロトコルの統一です。光による高速通信は、私たちにストレスを感じさせることなく音声や映像を届けてくれるようになりました。通信プロトコルTCP/IPへの統一は、ネットワークの構築を容易なものとし、ネットワークどうしが接続されたネットワーク、すなわちインターネットの幾何級数的拡大を実現しました。ネットワークの拡大による利用者数の増加は、「通信網の価格は利用者数に比例し、通信網の価値は利用者数の二乗に比例する」というメトカーフの法則（Metcalfe's law）どおり、ネットワークの価値を飛躍的に高めました。

端末のエポックは、CRTの採用とPC（パーソナルコンピュータ）の出現です。CRTは、それまでのタイプライタに比べ高速で面的表示の特性から、CUIからGUIへとマンマシンインターフェースを改善しました。PCは、最初こそCRT端末のエミュレータとして導入されましたが、しだいにその特性が活かされ、クライアントの位置づけになりました。サーバとの機能分担によって、ブラウザと呼ばれるプレゼンテーション機能だけを持たせたシンクライアント、ビジネスロジックまで拡大しデータベースだけをサーバに残したファットクライアント、両クライアントの中間的存在のリッチクライアント、に使い分けられるようになりました。

そして現在、端末の分野にニューフェースが現れました。スマートフォンです。スマートフォンは自身でアプリ開発こそできないものの、多種多様なアプリをダウンロードすることができます。もちろん、シンクライアントにもファットクライアントにもなります。本来の電話機能に加え、GPS、加速度センサー、ジェスチャ、音声認識機能などを備え、なにより手のひらサイズの優れたポータビリティから、PCでは考えられなかった分野への適用が期待されています。

現在、揺籃期にありがちなこととして、「Android」「iOS」「Windows Phone」の陣営に分かれて、技術とマーケティングでしのぎを削っています。それぞれに主義主張があり、一長一短があり、一概に優劣の判断はできませんが、オブジェクト指向が3者に共通する技術となっています。

本書は、以上の現状を踏まえて、3者のうちの「Android」をプラットフォームにして、スマートフォンや同類のタブレットに共通する、オブジェクト指向による設計と実装について、開発経験をベースに実践的側面に重きを置いて著作したものです。本書が、読者のみなさまの、「Android」「iOS」「Windows Phone」スマートフォンアプリ開発の参考になれば幸甚です。

2012年7月　三苫健太

本書の読み方

本書は、「Android」「iOS」「Windows Phone」各プラットフォームのアプリ開発に共通する技術であるオブジェクト指向による設計と実装スキルの修得を第一の目的とし、「Android」をプラットフォームにして実践面に重きを置いて著作したものです。しかしながら、本書が前提とする読者の知識は「簡単なJavaプログラムを組める」程度とし、文中で使用する用語や事例は、平易で身近なものにしています。ビギナーの方は「第1部　基礎編」から、開発経験のある方は「第2部　拡張編」から熟読していただければ、本書の目的は達成されるものと考えます。

オブジェクト指向の特長はモデル化とパターン化にあります。モデル化によってシステムを視覚的に把握できれば、ドキュメントを含むシステムの品質を向上させることができます。パターン化すれば、システムの病根ともなる属人性を排除できます。また流用や援用が促進され、開発コストが抑えられます。もちろん、品質向上にも効果的です。しかし、オブジェクト指向にも短所があります。それは「冗長になりやすい」ことです。これは、省資源高効率が求められるスマートフォンでは、看過されないところです。問題をわかりやすくするために、良いアプリの条件を整理してみます。

- **信頼性**——正しい結果、求められる精度の結果が得られる。
- **一般性**——想定されるすべての事態に対応している。
- **高効率**——無駄な負荷や資源の浪費がない。スマートフォンでは省バッテリーも重要。
- **拡張性**——仕様変更や追加が容易である。

オブジェクト指向は「高効率」の点で問題を残すということですが、他の要件については群を抜いて優れています。したがって、まずはオブジェクト指向で開発し、信頼性や一般性が確認できたら、効率に対してクリティカルな部分をチューニングすることが大切です。オーソドックスなオブジェクト指向開発の手順とスキルを学ぶ、これが本書の第二の目的です。

本書の各章の構成は、技術事項の説明→使用例→事例研究→演習、を基本としています。使用例で記述方法を学び、事例研究で実例から適用方法を学び、演習で知識とスキルの定着を図ろうとするわけです。使用例、事例研究、演習で対象とするアプリも、章を通してそれぞれ同じものを採用しています。これは、章の進行とともにアプリが完成されていく過程を見てもらうためであり、本書の第三の目的でもあります。したがって、時間の都合で演習を取り組めない章があったとしても、解答例をコピーするなどして章ごとに解答を完成されることをお勧めします。

なお、本文に示したソースリストは、テーマに直接関係のない部分は省略しています。これは、コードを複雑にして読者を無用に混乱させることがないようにとの配慮によるものです。したがって、完成されたコードは本書のサイト（http://gihyo.jp/book/2012/978-4-7741-5189-2）からダウンロードしてご確認ください。

各章のポイント

第１章　Androidアプリ開発の基本

Androidアプリを構成するActivity、Service、Receiverの各コンポーネントの役割と相互の関係を解説します。続いて、EclipseによるAndroidアプリ開発の方法について、手順を追って解説します。

MVCアーキテクチャを概説し、実際に簡単なAndroidアプリをMVCアーキテクチャで作ってみます。

第２章　UIクラスと画面設計

Android画面を構成するUIクラスについて、まず相互の継承関係を明確にします。続いて、すべてのUIクラスの基底になるViewとViewGroupの属性とメソッドを解説します。その後、すべてのUIクラスをレイアウトクラス、ウィジェットクラスに分けて解説します。

画面設計のドキュメント、画面に関係するリソース、設計から実装にいたる手順を解説します（事例研究と演習あり）。

第３章　ビューの設計と実装

ビュークラスの標準メンバー（プロパティとメソッド）、ビューが主目的とする画面制御の方法を解説します。また、画面入力の補助手段として使うダイアログを解説します。ビュー設計における静的モデルと動的モデルの意義、単票型画面とリスト型画面の設計と実装の相違点を明確にします（事例研究と演習あり）。

第４章　コントロールの設計と実装

コントロールクラスの標準メンバーを、メニュー機能、状態遷移（ライフサイクル）、画面遷移などの切り口から解説します。メニュー機能については、オプションメニューとコンテキストメニューの生成のしかたも含めて解説します。画面遷移では、インテントの具体的使用方法を詳述します。プリファレンス画面の定義と対応するコントロールクラスの標準メンバーも解説します（事例研究と演習あり）。

第５章　モデルの設計と実装

リソース型モデルとイベント型モデルの設計パターン、パターンを構成するビーンズ、DTO、DAOの標準メンバーを解説します。DAOは、データベースDAO、ファイルDAO、リソースDAO、に分けて解説します（事例研究と演習あり）。

第６章　インテントの設計と実装

インテントとその構成要素、インテント受信のためのインテントフィルタについて、設計方法を含めて解説します。明示的インテントと暗黙的インテントの相違点と使い分けを解説します（事例研究と演習あり）。

第７章　サービスコンポーネントの設計と実装

サービスコンポーネントの設計パターンと、バックグラウンドタスク、バインドサービスを解説します。バックグラウンドタスクではサイクリック型とキュー型の標準クラスを、バインドサービスではプロセス間通信を中心に解説します（事例研究と演習あり）。

第８章　レシーバコンポーネントの設計と実装

ブロードキャストレシーバクラスの標準メンバー、ブロードキャストインテントとインテントフィルタを解説します。システムが発行するブロードキャストインテントも紹介します（事例研究と演習あり）。

第９章　コンテンツプロバイダの設計と実装

コンテンツプロバイダクラスの標準メンバーを、データベースプロバイダを例に解説します。URIの活用方法についても解説します（事例研究と演習あり）。

第10章　ジェスチャと音声認識／合成の適用

ジェスチャを適用したコントロールクラスの標準メンバーとジェスチャ解析クラスを解説します。

音声認識を適用したコントロールクラスの標準メンバー、音声合成を適用したコントロールクラスの標準メンバーと合成エンジンを解説します（事例研究と演習あり）。

第11章　位置と地図情報の適用

位置情報アプリの構成、コントロールクラスの標準メンバーとロケーションプロバイダを解説します。ロケーションプロバイダでは、GPSプロバイダとネットワークプロバイダについて相違点を中心に解説します。地図情報アプリのコントロールクラスとビュークラスの標準メンバーを解説し、最後に、地図への位置情報のオーバーレイについて解説します。

第12章　Web通信の適用

Web通信をDAOパターンで構造化したときの、各クラスの標準メンバー、別スレッドでの通信、同一スレッドでの通信を解説します。スレッド間の連携を行うハンドラも解説します。