2 Android プログラミング入門
Android の概要からプログラミングまで

木島貴志　　石丸宗平
(株) ナノコネクト

Android の可能性とその動向

■ Android とは何か

● Android とは

Android とは，米 Google 社が中心となって，開発を進めている Linux ベースの携帯端末用プラットフォームです．主にスマートフォンのプラットフォームとして利用されることを想定されており，世界中で続々と Android を搭載した端末がリリースされています．

Android の特徴としては，ソフトウェアのライセンス料などが一切課されず，プラットフォームが無償で提供されており，ソースコードがオープンソースとして公開されているということです．

また，Android で動作するアプリケーションを，世界中の Android 端末ユーザに公開，販売できる Android Market があり，個人のアプリケーション開発者でも簡単にアプリケーションを公開，販売することができます．

● いろいろなデバイスに移植されている

Android は，主要なモジュールの多くは，Apache License2.0 を採用しており，Apache License を使用していることを明示すれば，開発者が独自に改変して使用することができます．そして，企業が持つソフトウェア技術や資産を Android に組み込んでも，公開する義務は発生しません．したがって，多くの企業にとって使用しやすいライセンス条項となっているため，カーナビゲーションシステム，放送信号を受信する装置であるセットトップボックス，フォトフレーム，プリンタ，デジタルテレビなど，さまざまな家電製品への移植が行われています．

● Android が生み出す可能性

Android を搭載したデバイスは，クラウドコンピューティングサービスの恩恵を受けるための共通のプラットフォームになり，またアプリケーション開発エンジニアからすると，アプリケーションを提供するための共通の市場になります(図 -1)．

そして携帯端末はもちろん，その他の家電にも搭載され，組込み OS としてもデファクトスタンダードになる可能性を十分に秘めています．

最近では，Android を搭載したパソコンもリリースされ始めているおり，今後サポートされるアプリケーションや，Web アプリケーションの普及によっては，パソコン用の OS の市場も Android のシェアが続伸してい

図 -1　Android とクラウドサービス

図 -2　Android のアーキテクチャ階層図

く可能性は十分にあると考えられます．

Android のアーキテクチャ

Android プラットフォームのアーキテクチャは，Linux カーネルをベースに，その上位にいろいろな機能を実装したネイティブライブラリ群や，Dalvik と呼ばれる仮想マシンが実装され，仮想マシン上で Android のアプリケーションが実行されるというアーキテクチャになっています．

アプリケーションからデバイスを制御するシーケンスは，アプリケーションからフレームワークの API をコールし，そこからライブラリ層のミドルウェアがコールされ，ミドルウェアから Linux カーネルを通して，デバイスドライバに処理が移り，デバイスドライバからデバイスが制御されます（図 -2）．

● アプリケーション

Android のアプリケーションには，Android プラットフォームに標準で含まれているメールクライアント，ブラウザ，電話帳などの基本アプリケーション，携帯電話メーカがあらかじめインストールするバンドルアプリケーション，ユーザがダウンロードしてインストールするダウンロードアプリケーションといった種類があります．

Android では iOS とは違い，「すべてのアプリケーションが平等である」という思想がありますので，電話帳や文字の入力などの基本アプリケーションであろうが，ダウンロードアプリケーションで置換することができます．この思想により，システムの根本からカスタマイズすることができるということが Android の魅力でもあります．

● アプリケーションフレームワーク

アプリケーションを起動したり，ウィンドウを表示したりするためのライブラリの総称です．

Android アプリケーションを容易に作成するための API が用意され，Android アプリケーションはこのアプリケーションフレームワークの API から，ライブラリ層以下のミドルウェア群の機能を使用することができます．

● Android ランタイム

Android は，Java SE を元に作成されているため，Java SE で利用できるコア・ライブラリの大半は含まれていますが，GUI に関する java.awt パッケージ等は削除されています．そして Android 固有の GUI を補う Android のパッケージや，センサやカメラなどのハードウェアを制御するためのパッケージが追加されています．

Dalvik 仮想マシンは Java の仮想マシンと同様の機能を提供していますが，バイトコードそのものには互換性がありません．Android のアプリケーションは Java で記述されますが，コンパイルされて Java バイトコードになった後，dx と呼ばれるツールで，Dalvik Executable(.dex) 形式へ変換されます．Dalvik 仮想マシンは，変換された Dalvik Executable 形式のファイルを，プロセス単位で実行します．

● ライブラリ

C や C++ 言語で作成されたライブラリ

で構成されており，Linux の libc や SQLite，Webkit などのオープンソースのライブラリを Android 仕様にカスタマイズし，各機能を実現するための汎用的なミドルウェア群として構成されています．

● Linux カーネル

Android は Linux ベースのプラットフォームであるため，Linux カーネル層があります．

Linux カーネルバージョンは 2.6 が採用され，セキュリティ，メモリ管理，プロセス管理，ネットワークスタック，デバイスドライバなどのコア・システムサービスを提供しています．

このカーネルは，ハードウェアと上位階層との抽象化レイヤとしても機能しています．

● ハードウェア

物理的なハードウェアの層であり，CPU，メモリ，ディスプレイデバイス，Bluetooth デバイス，GPS デバイス，オーディオデバイス，バッテリー等がこれに該当します．

■ Android デバイスとアプリケーションの特徴

● 最近のおもしろいデバイス

最近では，多種多様のデバイスに Android の移植が進んでおり，異彩を放つデバイスが世の中に出てきています．たとえば，Android が搭載されたテレビのリモコンや，自転車のサイクルコンピュータなどがリリースされて注目を浴びています．

ナノコネクト社が開発した Android 搭載の「すーぱーどろいど君」も例外ではないかもしれません．「すーぱーどろいど君」は，タッチパネル LCD，Bluetooth，WiFi，GPS，カメラ等のデバイスが搭載されており，遠隔地から，カメラに映った映像を見ながら，ロボットを制御することができます．

たとえば，「すーぱーどろいど君」を自宅に設置し，外出先からネットワークを介して自宅の状態を監視したり，他のシステムと連動して家電をコントロールしたりできるようになります（図 -3）．

図 -3　すーぱーどろいど君 外面写真

● 新規にデバイスを追加するには

標準的に Android がサポートしていないデバイスを追加する場合，デバイスを物理的に接続し，デバイスドライバを追加する必要があります．さらにそれを Linux カーネルに登録し，必要であればミドルウェア，Java のアプリケーションから利用しやすい形にするためのフレームワークを作成します．そのため新規にデバイスを追加するにはカーネルのコンパイルが必要になります．

■ Android アプリケーションを販売

● Android Market とは

Android Market とは，アプリケーションの利用者と，開発者との仲介を目的とした Google 社のサービスです．

開発者は，アプリケーションやアプリケーションの情報，プロモーション用の宣材を登録したり，ユーザからの評価やエラー報告を受けたりすることができます．

逆に利用者は，必要なアプリケーションをカテゴリやキーワードで検索し，アプリケーションの購入やインストールを手軽に行うことができます．

● 開発者アカウントの取得

Android Market でアプリケーションを公

開するためには，開発者アカウントの登録が必要になります．開発者アカウントを取得するには，Google アカウントとクレジットカードの登録と，初回の登録料として，25 米ドルが必要になります[☆1]．

● アプリケーションの登録

作成したアプリケーションは，アプリケーションを配布する形式の APK 形式で出力し，それにディジタル署名にて署名を行います．そしてデベロッパーコンソールと呼ばれる Web サイトから公開するアプリケーションの APK ファイルやスクリーンショット，説明文章（多言語にも対応），カテゴリ，公開地域などを指定します．設定項目を入力した後に「公開」をクリックすると Android Market にてアプリケーションがダウンロードできる状態になります．

そのほかにもデベロッパーコンソールでは，アプリケーションのダウンロード数やアップデート，ユーザの評価，エラー情報などの統計情報も閲覧することができます．

● Android アプリケーションを販売して億万長者に

海外では，Android アプリケーションを販売して，毎月数百万円の収入を得ている開発者もいます．

この Android アプリケーションを企業でも開発し，販売すれば大きな収入になるのではと考えている企業も多いと思います．しかし現状は厳しく，大ヒットする Android アプリケーションはわずかで，アプリケーションの販売だけでは企業の収益を支えていくのは困難です．

そこで Android アプリケーションの開発者は広告を使用した収益モデルへ移行したり，クラウドサービスを立ち上げて，サービスの利用料を徴収したりするモデルへ移行しており，今後このようなビジネスモデルはさらに増えていくと思われます．

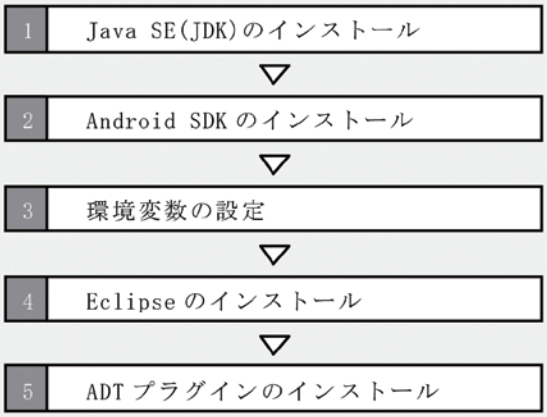

図 -4　開発／実行環境の構築手順

Android の開発／実行環境

■ 開発に必要なものを準備する

Android の概要に次いでアプリケーションの作成方法について記します．

はじめに Android アプリケーションを開発／実行するにあたって，必要なソフトウェアをインストールし，Android アプリケーションの開発を行う環境の構築を行います（図 -4）．

Android アプリケーションの開発に要求されるパソコンのスペックは，Android Developers の Web サイト[☆2] に記載されており，この要件を満たすパソコンを準備する必要があります．Windows，Linux，Mac OS X などのさまざまな OS に対応していますが，ここでは，Windows 7（64 ビット）上で，2011 年 1 月現在の最新版である以下のソフトウェアを用いた手順で説明します．

- Android SDK (r8)
- Eclipse IDE for Java Developers 3.6 (Helios)
- ADT Plugin 8.0.1

☆1 登録サイトの URL：http://market.android.com/publish/signup/
☆2 http://developer.android.com/sdk/requirements.html

- Java SE 6 Update 23 (JDK 6)

また Windows のユーザアカウントに 2 バイト文字が使用されていると，開発／実行環境が正常に動作しません．半角英数で表すことができる Windows のユーザアカウントを用意しておく必要があります．

● Java SE（JDK）のインストール

まず Java アプリケーションを開発／実行するために必要な Java SE のインストールを行います．パソコンから Sun Developers Network（SDN）のサイト☆3 へアクセスし，任意の場所へ「jdk-6u23-windows-i586.exe」をダウンロードします．ダウンロードした「jdk-6u23-windows-i586.exe」を実行し，インストールウィザードを進めていきます．ここでは特別な設定をする必要はなく，必要に応じてインストールする機能を設定し，インストールウィザードに従って進めていけばインストールは完了となります．

● Android SDK のインストール

次に Android アプリケーションの開発ツール群が含まれる Android SDK のインストールを行います．パソコンから Android Developers のサイト☆4 へアクセスし，「installer_r08-windows.exe」をダウンロードします．

「installer_r08-windows.exe」を実行し，インストールウィザードに従って進めていけばインストールは完了となります．

その後，スタートメニューに登録された「SDK Manager」を「管理者として実行する」で実行する☆5 と，リポジトリからパッケージをダウンロードするウィザードが始まります．今回は，「Android SDK Platform-tools, revision 1」と「SDK Platform Android 2.3, API 9, revision 1」のパッケージをダウンロードします．なお，HTTP プロキシサーバを介在したネットワーク環境下では，「Android SDK and AVD Manager」-「Settings」の「Proxy Settings」項にて，「HTTP Proxy Server」と「HTTP Proxy Port」を設定しなければ通信を行うことができません．

● 環境変数の設定

ここでは Android SDK と，Java SE（JDK）のツールにパスを通します．Android SDK や Java SE（JDK）には，コマンドラインで操作することができるツールが多数含まれているのでパスを通しておくことを推奨します．

環境変数を設定するには，コンピュータ☆6 を右クリック -「プロパティ」-「システムの詳細設定」-「環境変数」を開く．Android SDK の tools および platform-tools ディレクトリと，Java SE（JDK）の bin ディレクトリのパスをセミコロン区切りで設定します．その際，ユーザまたはシステム環境変数に PATH がすでにある場合は，セミコロンの後ろに以下の値を追加し，PATH がない場合は新規に変数 PATH を追加します．

パスを通すディレクトリ
- C:\Program Files (x86)\Android\android-sdk-windows\tools
- C:\Program Files (x86)\Android\android-sdk-windows\platform-tools
- C:\Program Files (x86)\Java\jdk1.6.0_23\bin

● Eclipse のインストール

Eclipse は，Java などの言語に対応したオープンソースの統合開発環境で，Android アプリケーションの開発において使用が推奨されています．

パソコンから，Eclipse のサイト☆7 へアク

☆3 http://java.sun.com/javase/ja/6/download.html
☆4 http://developer.android.com/sdk/
☆5 Windows XP 環境下では，「SDK Manager.exe」をダブルクリックしての起動でよい．
☆6 Windows XP では，マイコンピュータを指す．
☆7 http://www.eclipse.org/downloads/

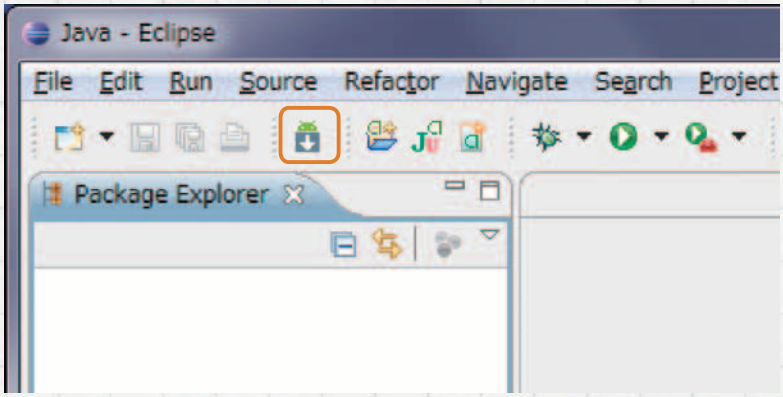

図 -5　Android SDK and AVD Manager のアイコン

セスし，「Eclipse IDE for Java Developers」にある「eclipse-java-helios-SR1-win32.zip」をダウンロードし，ダウンロードしたアーカイブを，任意の場所に展開することで Eclipse のインストールは完了となります．

● ADT プラグインのインストール

単体の Eclipse では，Android アプリケーションの開発には対応していません．Android SDK に含まれる開発／実行環境を，Eclipse 上で使用できる ADT プラグインを Eclipse にインストールします．

ADT プラグインのインストールは，2 種類方法があり，URL を指定してネットワーク上からインストールする方法と，ADT プラグインのアーカイブをあらかじめダウンロードしておき，それをインストールする方法があります．今回は，前者の方法を用いて以下の手順でインストールを行います．ただし，HTTP プロキシサーバを介在したネットワーク環境下では，Eclipse の「メニューバー」-「Window」-「Preferences」を開き，「General」-「Network Connections」に HTTP プロキシの設定を行う必要があります．

① Eclipse を起動します．初回起動時には，どのワークスペースで作業を行うかを指定するダイアログが表示されるので，任意のディレクトリを指定します．

②「メニューバー」-「Help」-「Install New Software. . .」をクリックします．

③ ダイアログの「Add」ボタンをクリックし，「Location」に以下の URL を入力します．
https://dl-ssl.google.com/android/eclipse/

④「Developer Tools」にチェックを入れ，ウィザードに従ってインストールを進めていきます．

⑤ Eclipse の再起動が求められ，再起動を行います．Eclipse のツールバーに，「Android SDK and AVD Manager」を起動するアイコン（図 -5）が表示されていれば，正常に ADT プラグインのインストールが完了した状態となります．

⑥「Eclipse のメニューバー」-「Window」-「Preference」-「Android」の「SDK Location」に，「android-sdk-windows」ディレクトリのパスを指定します．画面上部のエラー表示が消えると，ADT プラグインの設定は完了となります．

■ 仮想デバイス (AVD) の作成と起動

Android アプリケーションの開発を，パソコンのみで行う場合は，パソコン上でアプリケーションをエミュレート実行できる仮想の Android デバイスである AVD を作成します．この仮想デバイスでは，センサや Bluetooth などのハードウェア部分がエミュレートされていないため，これらのハードウェアを利用するアプリケーションを作成する際には実機が必要となります．

AVD を作成する手順は，「Android SDK and AVD Manager」から，「Virtual devices」-「New」をクリックし，「Name」に任意の名称，「Target」に「Android 2.3 - API Level 9」を選択して「Create AVD」をクリックすると，「Virtual devices」に AVD が追加されます．

追加した AVD を選択して，「Start」を押すと，起動オプションが表示され，環境の解像

度に応じて Scale 値を設定し，「Launch」を押すとエミュレータの起動が始まります．

ソースコードの配布について

本項で使用しているソースコードおよびリソースファイルなどのプロジェクトに関するファイルは，次の Web サイトで配布を行っています．これを元にして次項より解説を行っていきます．

http://www.nanoconnect.co.jp/samplecode/ipsj/

簡易計算機の作成

本稿では，簡易計算機を作成し，Android のプロジェクトの作成方法，基本的なコンポーネントの使い方，イベントの取得方法について解説を行い，Android アプリケーションの作成についての流れを取得することを目的とします．

作成するアプリケーションは，値を入力できるフィールドを 2 つ持ち，スピナーと呼ばれるリストで四則演算を行う記号を選択します．一番下の「=」ボタンを押すと，トーストと呼ばれる画面中央から下寄りに表示される灰色の角丸矩形で計算結果を表示するとともに，「=」ボタンが伸縮するアニメーションを実行します（図 -6）．

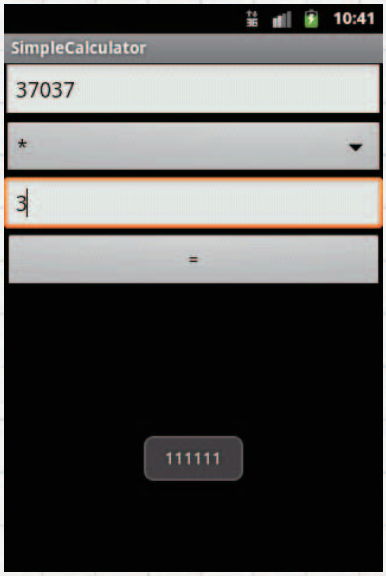

図 -6　簡易計算機の画面

■ プロジェクトの新規作成

● プロジェクトを新規に作成する

Eclipse は，プロジェクトと呼ばれる単位で，作成するアプリケーションを管理します．Android も同様に，「Android Project」を新規に作成してアプリケーションを作成していきます．「Android Project」を作成するには，以下の手順でプロジェクトを作成します．

①「Eclipse のメニューバー」-「File」-「New」-「Android Project」をクリックします．

② それぞれの項目に対して，表 -1 の設定値を入力します．

③「Finish」をクリックします．以上の手順でプロジェクトのテンプレートが自動生成されます．

● プロジェクトの構成

生成されたプロジェクトは，テンプレートで定義された構成になっており，必要に応じてディレクトリの追加，削除を行います．それぞれのディレクトリ，ファイルの役割を表 -2 に記します．

■ 画面の作成

Android アプリケーションでは，XML 形式で，画面のレイアウトを作成することができます．これによりプログラマとデザイナが同時に開発を進めることができ，また画面サイズや縦横の方向が異なるなどの動作環境によって容易にレイアウトを切り替えることができるようになります．

● レイアウトファイルを新規作成

作成した SimpleCalculator プロジェクトの，「res/layout」にレイアウトを定義した XML ファイルを新規に作成します．

項目名	概要	設定値
Project name	Eclipseで管理するための識別名	SimpleCalculator
Build Target	ターゲットのバージョン	Android 2.3
Application name	Android端末上で表示されるアプリケーション名	Simple Calculator
Package name	作成者のドメインを逆順にしたもの．アプリケーションごとにユニークな名称が必要	jp.co.nanoconnect.simplecalculator
Create Activity	自動生成するActivityクラスの名称	CalculatorActivity
Min SDK Version	アプリケーションが動作する最小のバージョン	設定なし

表 -1　プロジェクトの設定

名称	概要
src/	ソースコードを格納するディレクトリ
gen/	リソースファイルに割り振られたIDを宣言したRクラスが格納されているディレクトリ
assets/	res/に格納できないHTMLファイルやフォントファイル，その他ファイルを格納するディレクトリ
res/	アプリケーションで使用するリソースファイルを格納するディレクトリ．定められたディレクトリ名のみ格納することができます
res/drawable/	主に画像ファイルを格納するディレクトリ
res/layout/	XMLで定義したレイアウトを格納するディレクトリ
res/values/	XMLで定義した文字列，色，テーマ，サイズなどを定義するディレクトリ
AndroidManifest.xml	Androidアプリケーションに関する情報を設定するファイル

表 -2　プロジェクトディレクトリの構成

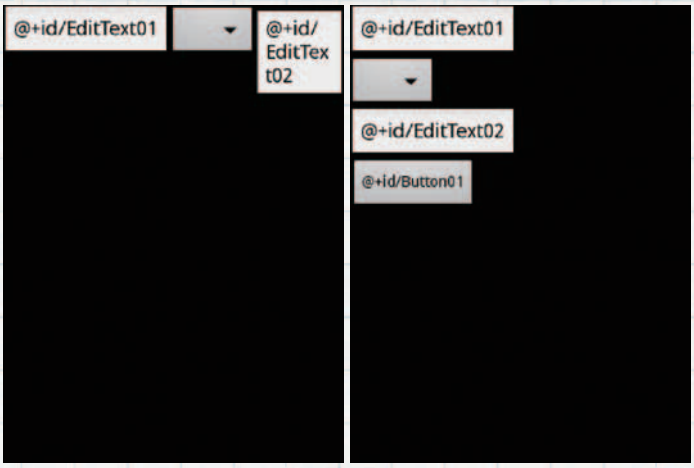

図 -7　縦方向へ整列後の画面

①「Package Explorer」の「SimpleCalculator」-「res」-「layout」にフォーカスを当て，「右クリック」-「New」-「Android XML File」を選択します．

②「File」に，「calculator.xml」と入力し，「Finish」をクリックする．これにより，「calculator.xml」の雛形が生成されます．

● 画面レイアウトを作成

ADT プラグインには，レイアウト画面を GUI 操作または XML の直接編集で作成することができる機能を備えており，画面中央にある「Graphical Layout」タブと「calculator.xml」タブで，編集モードを切り替えることができます．GUI モードで編集する場合は，「Palette」に用意された GUI 部品の一覧から，画面上にドラッグアンドドロップすることで，View と呼ばれる GUI 部品を画面上に配置することができます．また，複数の View を内包して並べることができるコンポーネントを ViewGroup と呼びます．

● LinearLayout の配置方向を指定

今回作成するアプリケーションでは，上から順に「EditText」，「Spinner」，「EditText」，「Button」という順に View を並べていきますが，意図に反して View は横方向に並んでしまいます．これは，レイアウトファイルの作成時に自動挿入された「LinearLayout」と呼ばれる View の性質で，縦方向か横方向のどちらかにしか View を並べることができず，初期の設定では横方向へ配置されます．並べる方向を切り替えるには，画面上の背景部分（黒色）を右クリックし，「Orientation」-「vertical」を選択すると，View を縦方向に配置することができます（図 -7）．

● View の幅を変更

View の高さや幅は，具体的なピクセルなどの数値で指定することもできますが，「wrap_content」と「match_parent」という相対的な値で指定することができます．

「wrap_content」は，中の要素に合わせたサイズ，「match_parent」は，その View の 1 階層上の ViewGroup に合わせたサイズを取ります．幅や高さの指定方法は，変更したい View を右クリックし，「Layout width」または「Layout height」の属性を変更します．

今回は，すべての View の「Layout width」を，「match_parent」に変更します．

● ID を変更

ソースコード上から，XML 形式で定義した View や ViewGroup を識別するために，ID を付加することができます．GUI モードでドラッグアンドドロップした View は，「EditText01」などの ID があらかじめ割り振られますが，View の数が増えると，識別しにくくなるため，任意の ID に変更を行います．ID を変更するには，変更する View を右クリックし，「Show In」-「Properties」を選択すると，画面下段に「Properties」と呼ばれるウィンドウが開きますので，その「Properties」ウィンドウの「Id」に，新たな ID を入力します．今回は，上から順に，「@+id/operand1」，「@+id/operator」，「@+id/operand2」，「@+id/equal」に変更を行います．

● ボタンのラベル

アプリケーションで使用する文字列は，res/values ディレクトリ以下の XML ファイルに定義することが一般的であり，文字列を XML ファイルに局在化させることで，多言語への対応が容易となります．新たに文字列を追加するには，「res/values/strings.xml」ファイルをダブルクリックし，Android Resources を開きます．そして「Add」-「String」-「OK」を選択し，文字列の名前と値を追加します．今回は，「Name」に「btn_text_equal」，「Value」に「=」を定義します．

定義した文字列を参照するには，「Properties」ウィンドウの「Text」項目の「Value」列にフォーカスを当てると，画面右端に「...」ボタンが表示されます．そのボタンを選択すると，文字列として定義されているリソースが一覧表示されるので，あらかじめ定義しておいた「btn_text_equal」を選択します．

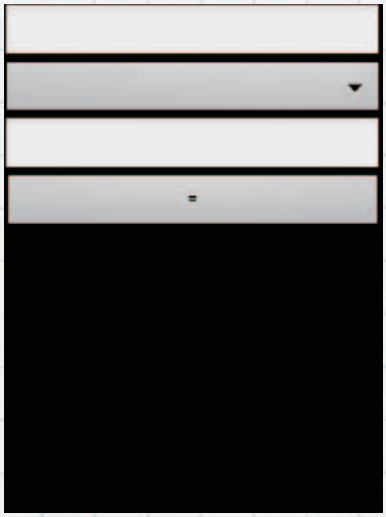

図 -8　Text を指定後の画面

また，「EditText」の「Text」項目にあらかじめ入力された「@+id/EditText01」などの不要な文字列は，削除しておきます（図 -8）．

● その他の属性を指定

EditText には，入力可能な文字の種類を指定することができる「Input type」と呼ばれるプロパティが用意されています．今回は，符号付きの整数値のみ入力を許可するため，「numberSigned」を設定します．そのほかにも，日付や URL などの種類を設定することができます．

以上の手順でレイアウトファイルは完成となります．

● XML でアニメーションを定義

Android では，簡易なアニメーションを XML でパラメータとして設定することができます．パラメータで設定できるアニメーションは，水平移動させる「translate」，透明度を変化させる「alpha」，回転させる「rotate」，伸縮させる「scale」が用意されており，複数の組合せも可能です．

```
<?xml version="1.0" encoding="UTF-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/bounce_interpolator">
    <scale android:fromXScale="0.6"
           android:toXScale="1.0" android:fromYScale="0.6"
           android:toYScale="1.0" android:pivotX="50%"
           android:pivotY="50%" android:duration="500" />
    <alpha android:fromAlpha="0.0" android:toAlpha="1.0"
           android:duration="500" />
</set>
```

図 -9　アニメーションを定義した bound.xml ファイル

```
<string-array name="operands">
    <item>+</item>
    <item>-</item>
    <item>*</item>
    <item>/</item>
</string-array>
```

図 -10　演算子の文字列配列を定義した strings.xml ファイル

今回のアプリケーションでは，ユーザの操作に対するアクセントとして anim/bound.xml に，対象の View の 0.6 倍から 1.0 倍へ拡大し，透明から非透明へ変化するアニメーションを実装しています．

また，「scale」や「rotate」では変化の中心座標を，「android:pivotX」や「android:pivotY」で定義します．今回のアプリケーションでは，アニメーションさせる対象のビューの縦と横の 50% を変化の中心座標として定義し，アニメーションを実行する時間は，「android:duration」を用いて 500 ミリ秒としています．

開始時と終了時のパラメータに加えて，「android:interpolator」を用いると，アニメーション実行中の動き方を指定することができます．あらかじめ Android プラットフォームで用意された「anim/bounce_interpolator」を指定することで，弾むような動きになります（図 -9）．

● 文字列配列を作成

スピナーで表示する演算子は，文字列配列をリソースファイルに定義して使用しています．/res/values/strings.xml に，<string-array> エレメントを定義し，その中に <item> として演算子の文字列が定義されています（図 -10）．

■ ソースコードの作成

● Activity とは

Activity は，画面を持つ Android アプリケーションの作成において，なくてはならないもので，画面を持った機能の単位を指します．そして Activity は，独立したライフサイクルを持ち，特定の状態になるときに，それに対応したメソッドを実行するので，Activity を継承したサブクラスで，オーバーライドしたメソッドに処理を実装します．

プロジェクトを新規に作成した段階では，onCreate メソッドのみ自動生成された状態となっています．

onCreate メソッドは，Activity が生成される際に実行されるメソッドで，主に画面のレイアウトをセットする処理を実装します．電話の着信や，端末のホームボタンをユーザが押した場合などは，Activity の「onPause」-「onStop」メソッドと実行され，停止状態となります．ホームボタンを長押しするなどして，再びそのアプリケーションが前面に表示され，実行状態となる際には「onRestart」-「onStart」-「onResume」メソッドが実行された後に実行中へ復帰します．また，アプリケーションの実行中に戻るボタンが押されたときは，「onPause」-「onStop」-「onDestroy」メソッドと実行され，終了状態となります．

通常，1 つのアプリケーションは 1 つのプ

ロセスで動作しており，そのプロセス内で実行中のアクティビティがなくなるとプロセスを終了する候補になります．このリストに追加されていると，メモリ容量が少なくなった際に図 -11 の破線矢印の [プロセス終了] 経路を通り，アプリケーションは終了します．その際には onDestroy メソッドや onStop メソッドは呼ばれません．再びそのプロセスが起動する際には，「onCreate」-「onStart」-「onResume」メソッドと実行され，実行中状態になります．

したがってアプリケーションが停止するときには，必ず解放しなければならない処理や，永続化しなければならないデータは，画面が隠れる際に実行される onPause メソッドに実装しなければ，実行される保証はありません．

● レイアウトのセット (1)

画面上に XML 形式で定義したレイアウトファイルを表示するには，onCreate メソッドで引数として gen/[パッケージ名]/R.java に定義されたレイアウトファイルの ID を setContentView メソッドで渡します．今回は，calculator.xml の ID を指定して読み込みを行います (図 -12，図 -13)．

● ボタンイベントの通知先をセット (2)

XML で定義された View をソースコード上に取得します．ビューを表すリソース ID を，findViewById メソッドに渡すとそのビューインスタンスを取得することができ，必要に応じてビュークラスのサブクラスであるボタンやエディットテキストへキャストを行います．

続いてイベントの処理を行うためのビューがタッチされたときのイベントの通知先を設定します．OnClickListener インタフェースを実装したクラスを，取得してきたビューオブジェクトの setOnClickListener メソッドで渡すことでイベント発生時にコールバックさ

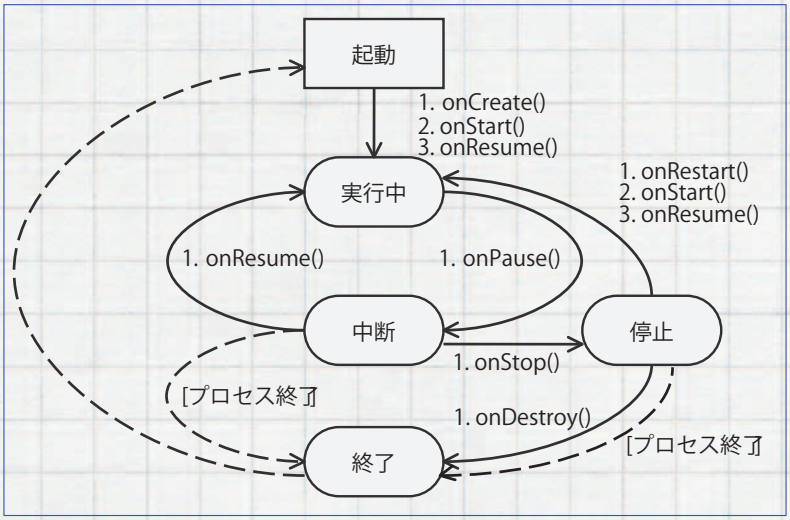

図 -11　アクティビティのライフサイクル

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    /* レイアウトのセット (1) */
    setContentView(R.layout.calculator);
}
```

図 -12　CalculatorActivity の onCreate メソッド

```
@Override
protected void onResume() {
    /* ボタンイベントの通知先をセット (2) */
    mEqualButton = (Button)
       findViewById(R.id.btn_equal);
    mEqualButton.setOnClickListener(this);
    /* アダプターを生成 (3) */
    ArrayAdapter<CharSequence> adapter =
      ArrayAdapter.createFromResource(
        this, R.array.operands,
        android.R.layout.simple_spinner_item);
    /* ドロップダウン時のレイアウトを指定 (4) */
    adapter.setDropDownViewResource(
      android.R.layout.simple_spinner_dropdown_item);
    /* スピナーの生成 (5) */
    mOperandSelector = (Spinner)
       findViewById(R.id.operand);
    mOperandSelector.setAdapter(adapter);
    /* アニメーションの読み込み (6) */
    mBoundAnimation = AnimationUtils
      .loadAnimation(this, R.anim.bound);
    super.onResume();
}
```

図 -13　CalculatorActivity の onResume メソッド

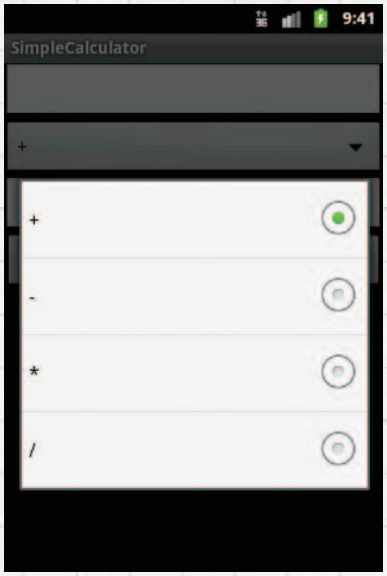

図 -14　スピナーのドロップダウン時

れるようになります．

●アダプターを生成（3）

動的に生成される可変のデータを，リスト状に表示するには，アダプターと呼ばれるものを使用します．アダプターは，表示するデータと，そのデータをどこに表示するかを関連付ける役割を果たします．

アダプターには，データの種類によってさまざまな種類のアダプターが用意されており，データベースを検索した結果のカーソルを用いる場合は CursorAdapter，文字列の配列を用いる場合は ArrayAdapter などがあります．

今回のアプリケーションでは，演算子を定義した文字列配列を，スピナーがタッチされた際に表示されるリストへ表示しますので，ArrayAdapter クラスを使用します．

ArrayAdapter を生成する際に，演算子を定義した文字列配列のリソース ID と，Android のシステムで定義された，1 つのリストに 1 行の文字列のみを表示するレイアウトの「simple_spinner_item」を指定します．

●ドロップダウン時のレイアウトを指定（4）

スピナーをタッチした際に表示されるリスト状のレイアウトを，ArrayAdapter クラスの setDropDownViewResource メソッドで定義することができます．今回はラジオボタンが付加された「simple_spinner_dropdown_item」を利用します（図 -14）．

●スピナーの生成（5）

XML で定義されたスピナーを findViewBuld メソッドの引数でスピナーのリソース ID「R.id.operand」を指定して取得します．取得したスピナーインスタンスに，ArrayAdapter インスタンスを setAdapter メソッドの引数として渡すことでドロップダウン時の選択肢が表示されるようになります．

●アニメーションの読み込み（6）

アニメーションを，特定のタイミングで実行するためには，まず XML で定義されたアニメーションをソースコード上に読み込む必要があります．アニメーションを読み込むには，コンテキストと，アニメーションを定義したリソース ID を，AnimationUtils クラスの loadAnimation メソッドに渡して，Animation インスタンスを取得します（図 -15）．

●ボタンのイベント処理（7）

ボタンが押されたときの処理を，OnClickListener インタフェースで定義された onClick メソッドに実装します．onClick メソッドの引数で，タッチされたビューインスタンスが入力され，そのビューインスタンスから getId メソッドで取得した ID と，R クラスに定義された ID を比較し，それに対応したイベント処理を行います．

●アニメーションの実行（8）

アニメーションを開始したいタイミングで，取得した Animation インスタンスを，View クラスの startAnimation メソッドに渡すと，アニメーションが実行されます．今回は，「=」ボタンがタッチされたときをイベントのトリガとしてアニメーションが実行されるように実装しています．

●スピナーの場所番号を取得(9)

スピナーがドロップダウン時にユーザによって選択されたアイテムの場所を示す番号を取得するには，getSelectedItemPosition メソッドを使用します．この番号により，スピナーの何番目のアイテムが選択されているかを判定し，それに応じた処理を行います．

●計算結果を表示(10)

今回のアプリケーションでは，トーストと呼ばれるユーザに簡単なメッセージを伝えるための GUI を使用します．Toast クラスの makeText メソッドで，コンテキスト，表示する文字列，表示する時間を設定した Toast を取得し，返り値の Toast インスタンスに対して show メソッドを実行することで表示することができます．

```
public void onClick(View v) {
    if (v.getId() == R.id.btn_equal) {
        /* ボタンが押されたとき，ソフトウェアキーボードを隠す */
        InputMethodManager imm = (InputMethodManager)
           getSystemService(Context.INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(
           v.getWindowToken(), 0);

        /* アニメーションの実行(8) */
        mEqualButton.startAnimation(mBoundAnimation);

        /* スピナーの場所番号を取得(9) */
        int operandId = mOperandSelector
           .getSelectedItemPosition();
        String answer = "";
        long value1 = getValue(R.id.value1);
        long value2 = getValue(R.id.value2);
        switch (operandId) {
            case OPERAND_PLUS:
                answer = Long.toString(value1 + value2);
                break;
            case OPERAND_MINUS:
                answer = Long.toString(value1 - value2);
                break;
            case OPERAND_ASTERISK:
                answer = Long.toString(value1 * value2);
                break;
            case OPERAND_SLASH:
                if (value2 != 0) {
                    answer = Long.toString(
                       value1 / value2);
                } else {
                    answer = "NaN";
                }
                break;
        }
        /* 計算結果を表示(10) */
        Toast.makeText(this, answer,
                Toast.LENGTH_SHORT).show();
    }
}
```

図-15　CalculatorActivity の onClick メソッド

■ アプリケーションの実行

●エミュレータで実行

作成したアプリケーションをエミュレータで実行するには，Eclipse のパッケージエクスプローラより，「右クリック」-「Run As」-「Android Application」を選択します．仮想デバイスが作成されていれば自動的にエミュレータが起動し，アプリケーションが自動転送され実行されます．デバッグで実行するには，「右クリック」-「Debug as」-「Android Application」を選択すると同様に実行されます．

●Android 端末で実行

実機でアプリケーションを実行するには，実機が「ADB Device」としてパソコンに認識されている必要があります．Android 端末上で，「設定」-「アプリケーション」-「開発」-「USB デバッグ」にチェックを入れ，パソコンに USB ケーブルで接続し，Android SDK の「SDK Manager」からダウンロードすることができる「Google USB Driver package」のドライバをインストールする必要があります．

認識した状態でエミュレータと同様の手順で実行すると，Android 端末に転送され，アプリケーションが実行されます．

参考文献
1）Android Hacks (ISBN 487311456X)
2）Android Developers : http://developer.android.com/

（平成 23 年 2 月 9 日受付）

木島貴志　kishima@nanoconnect.co.jp
（株）ナノコネクト代表取締役．RTOS の技術者を経て 2006 年に当社を創業．教育サービス，クラウドサービスを展開．社会的に意義のある商品を開発提供して，ユビキタス社会の発展に貢献することを目指している．

石丸宗平　ishimaru@nanoconnect.co.jp
（株）ナノコネクト Android アプリ開発部 主任．Android アプリケーションの教育や Android アプリケーションの開発に従事．