

第1章 導入編

第1節 開発環境の構築

開発環境の構築

Android アプリケーションを開発するためには、パソコン内の開発環境を整える必要があります。

※本書で説明している開発環境は 2014 年 12 月現在の情報です。

<ダウンロードが必要なもの>

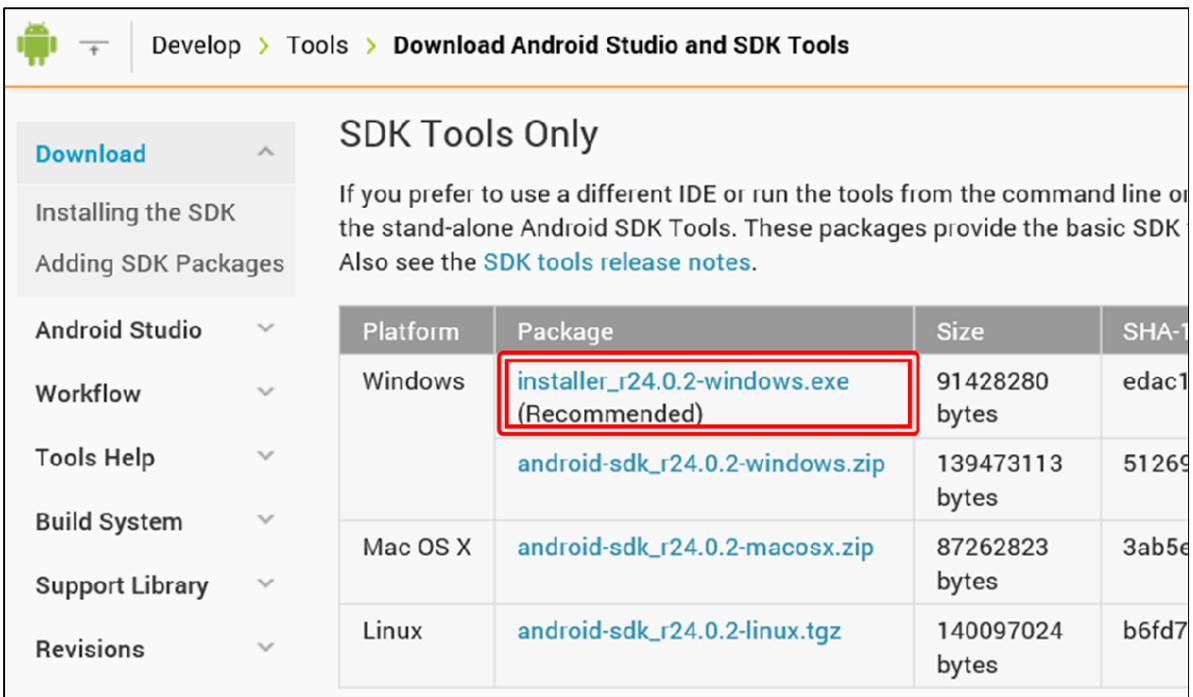
※OS は Windows の 64bit 版を例に挙げています。

1. installer_r24.0.2-windows.exe

「<http://developer.android.com/sdk/index.html>」のサイトから Android SDK をダウンロードします。

①ダウンロードするページへアクセス

画面中の「installer_r24.0.2-windows.exe (Recommended)」を選択するとダウンロード画面に進みます。



The screenshot shows the 'Download Android Studio and SDK Tools' page. Under the 'SDK Tools Only' section, there is a table of available packages. The 'installer_r24.0.2-windows.exe (Recommended)' package is highlighted with a red box.

Platform	Package	Size	SHA-1
Windows	installer_r24.0.2-windows.exe (Recommended)	91428280 bytes	edac1
	android-sdk_r24.0.2-windows.zip	139473113 bytes	51269
Mac OS X	android-sdk_r24.0.2-macosx.zip	87262823 bytes	3ab5e
Linux	android-sdk_r24.0.2-linux.tgz	140097024 bytes	b6fd7

②利用規約に同意してダウンロード

「I have read and agree with the above terms and conditions」を選択することにより利用規約に同意し、ダウンロードできるようになります。「Download installer_r24.0.2-windows.exe」を選択するとダウンロードが始まります。

Download

Before installing Android Studio or the standalone SDK tools, you must agree to the following terms and conditions.

Terms and Conditions

This is the Android Software Development Kit License Agreement

1. Introduction

1.1 The Android Software Development Kit (referred to in this License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of this License Agreement. This License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: <http://source.android.com/>, as updated from time to time.

1.3 "Google" means Google Inc., a Delaware corporation with principal place of business at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States.

利用規約に同意

I have read and agree with the above terms and conditions

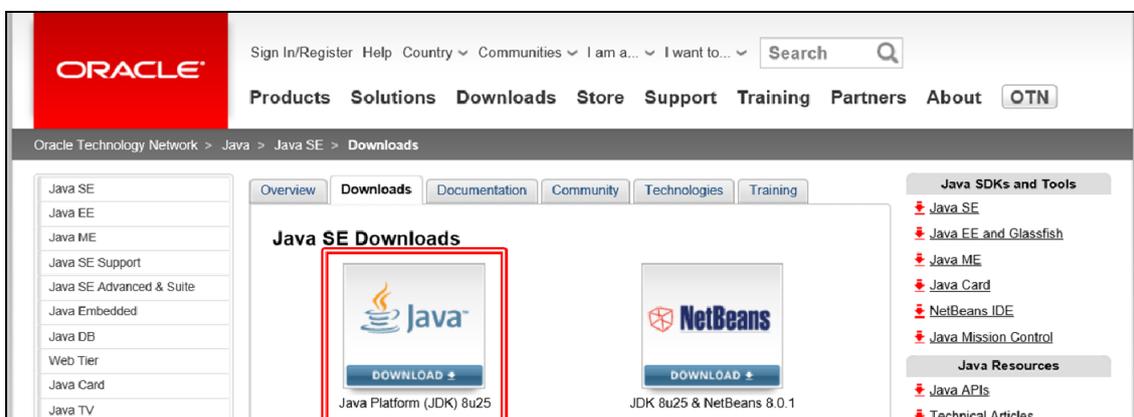
Download installer_r24.0.2-windows.exe

2. jdk-8u25-windows-x64.exe

「<http://www.oracle.com/technetwork/java/javase/downloads/index.html>」のサイトからJDKをダウンロードします。

①ダウンロードするページへ

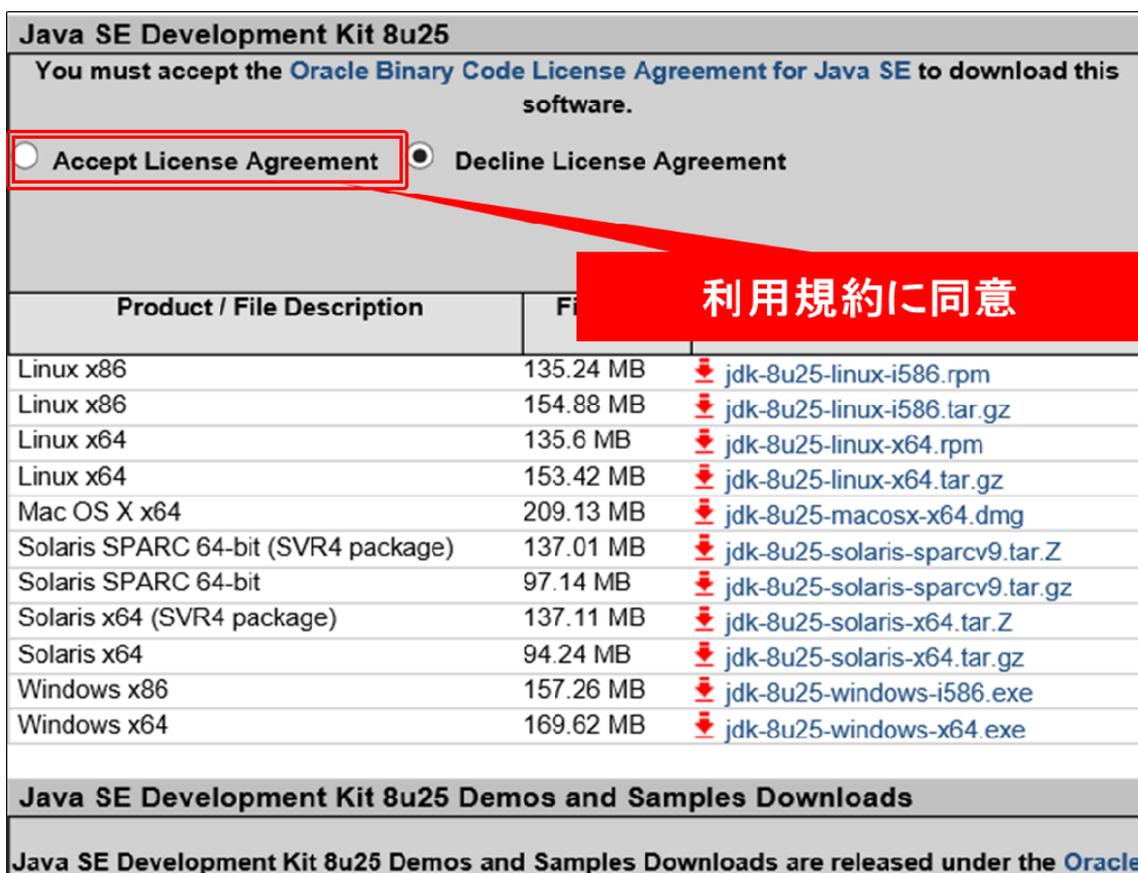
「Java Platform (JDK) 8u25」の「DOWNLOAD」を選択するとダウンロード画面に進みます。



② 利用規約に同意してダウンロード

「Accept License Agreement」を選択することで利用規約に同意し、ダウンロードできるようになります。OSによって選択する項目が異なるため、自分のパソコンの設定と対応する項目を選択してください。「Download」項目の「jdk-8u25…」を選択するとダウンロードが始まります。

- Windows x86 (32bit) → jdk-8u25-windows-i586.exe
- Windows x64 (64bit) → jdk-8u25-windows-x64.exe



Java SE Development Kit 8u25
You must accept the **Oracle Binary Code License Agreement for Java SE** to download this software.

Accept License Agreement **Decline License Agreement**

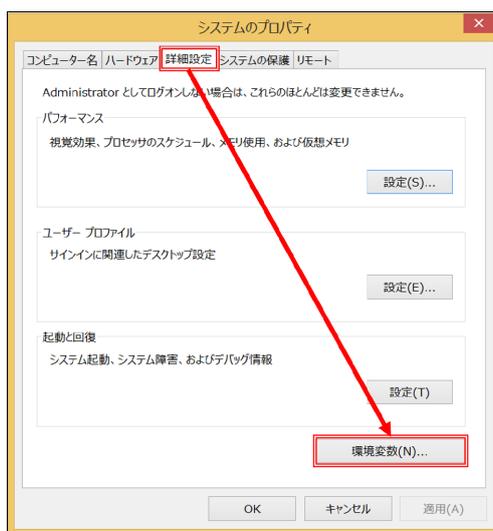
利用規約に同意

Product / File Description	File Size	Download Link
Linux x86	135.24 MB	jdk-8u25-linux-i586.rpm
Linux x86	154.88 MB	jdk-8u25-linux-i586.tar.gz
Linux x64	135.6 MB	jdk-8u25-linux-x64.rpm
Linux x64	153.42 MB	jdk-8u25-linux-x64.tar.gz
Mac OS X x64	209.13 MB	jdk-8u25-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	137.01 MB	jdk-8u25-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	97.14 MB	jdk-8u25-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	137.11 MB	jdk-8u25-solaris-x64.tar.Z
Solaris x64	94.24 MB	jdk-8u25-solaris-x64.tar.gz
Windows x86	157.26 MB	jdk-8u25-windows-i586.exe
Windows x64	169.62 MB	jdk-8u25-windows-x64.exe

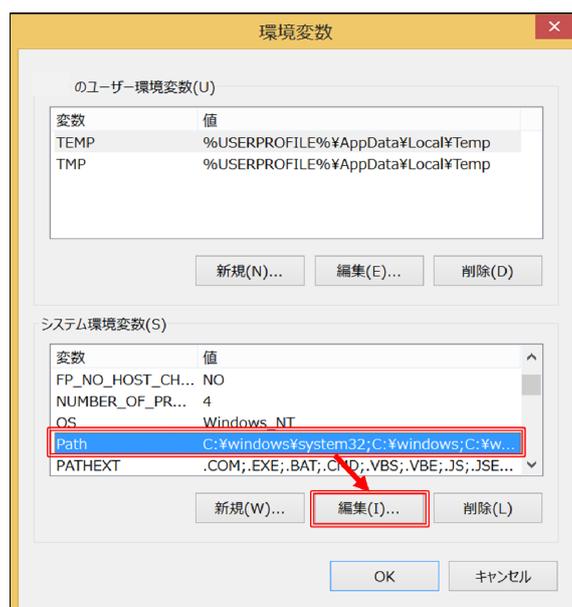
Java SE Development Kit 8u25 Demos and Samples Downloads
Java SE Development Kit 8u25 Demos and Samples Downloads are released under the **Oracle**

③環境変数の設定

Eclipse を起動するには環境変数の設定が必要です。「コントロールパネル」を起動し、「システムとセキュリティ」から「システム」を選択します。左サイドメニューの「システムの詳細設定」を選択すると、「システムのプロパティ」画面が表示されます。

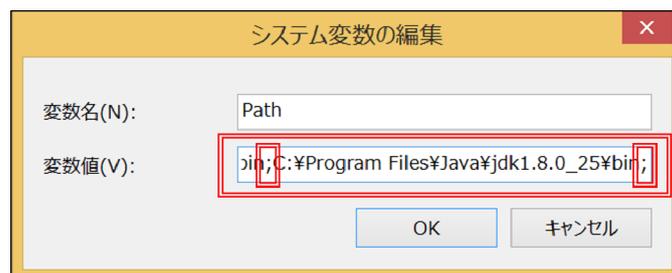


「システムのプロパティ」画面の「詳細設定」タブを選択し、「環境変数」から「環境変数」画面が表示されます。



システム環境変数の「Path」を選択し、「編集」を選択します。
※変数「Path」が無い場合、「新規」を選択し、「新しいシステム変数」画面から変数「Path」を作成してください。

「システム変数の編集」画面の変数値にJDKをインストールしたフォルダのパスを追加します。環境変数の設定は、すでに入力されている変数値を消さずに、セミコロン (;) 区切りで変数値の末尾に追加してください。



※この例では、変数値に「C:¥Program Files¥Java¥jdk1.8.0_25¥bin;」を設定しています。

設定完了後、パソコンを再起動してください。

3. pleiades-e4.2-java_20130303.zip

日本語化された開発環境 (Eclipse) を「<http://mergedoc.sourceforge.jp/>」のサイトからダウンロードします。Eclipse のバージョンを選択してダウンロード画面に進みます。本書では「Eclipse4.2 Juno」を使用しています。



ダウンロード画面で Java の 64bit、Standard Edition を選択して「pleiades-e4.2-java_20130303.zip」をダウンロードしてください。

Pleiades All in One 日本語ディストリビューション (zip) ダウンロード

Pleiades All in One 4.2.2a.v20130303

Eclipse 4.2.2a Juno SR2a for Windows ベース

- Juno SR1 で判明していた致命的なパフォーマンス問題が SR2 で改善されました。パフォーマンス関連 15
- 4.2.2.v201302xx には EGit の致命的なデータ破壊バグがあったため、4.2.2a.v201303xx 以降を使用し

- 開発対象となる言語に合わせてパッケージをダウンロードしてください。
- Full Edition には Eclipse 実行用の JRE や各言語の処理系が含まれています。よく分からない場合は Full Edition を選んでください。

 plugins、features ディレクトリーに格納されたプラグイン

 dropins ディレクトリーに格納されたプラグイン

 Eclipse 実行用の JRE や各言語のコンパイラー、ランタイムなどの処理系

		Platform	Ultimate	Java	C/C++	PHP	Python
32bit	Full Edition						
	Standard Edition						
64bit	Full Edition						
	Standard Edition						

<開発環境構築手順>

① フォルダの解凍

「pleiades-e4.2-java_20130303.zip」を解凍します。

※開発環境構築後、解凍したフォルダ「pleiades-e4.2-java_20130303.zip」を構築時と異なる場所へ移動させると、開発環境に不具合が生じる場合があるため、構築後にフォルダの場所を変更しないようにしてください。

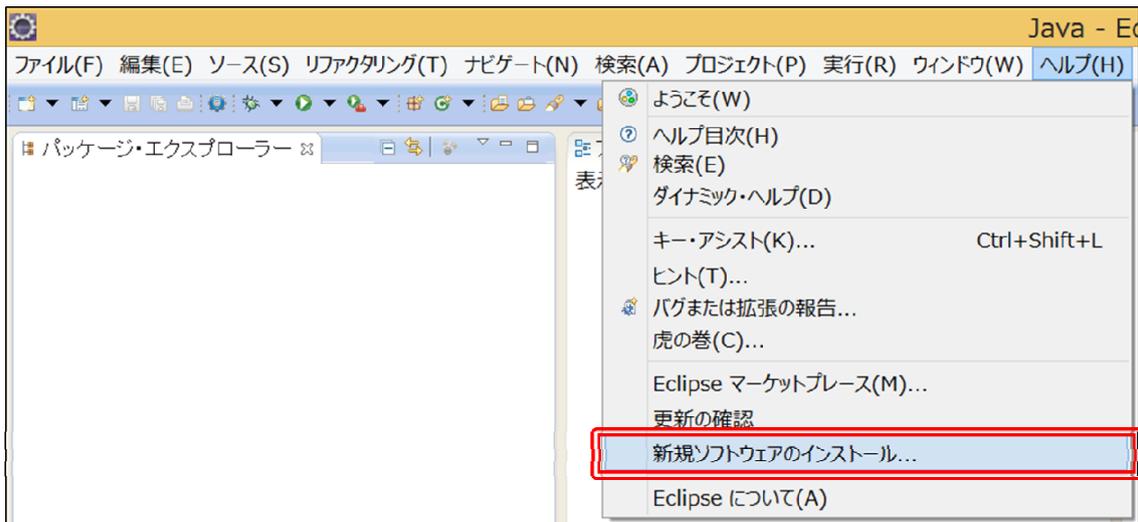
② Android SDK のインストール

「installer_r24.0.2-windows.exe」を実行して Android SDK をインストールします。

③ Eclipse の設定

1. ADT Plugin のインストール

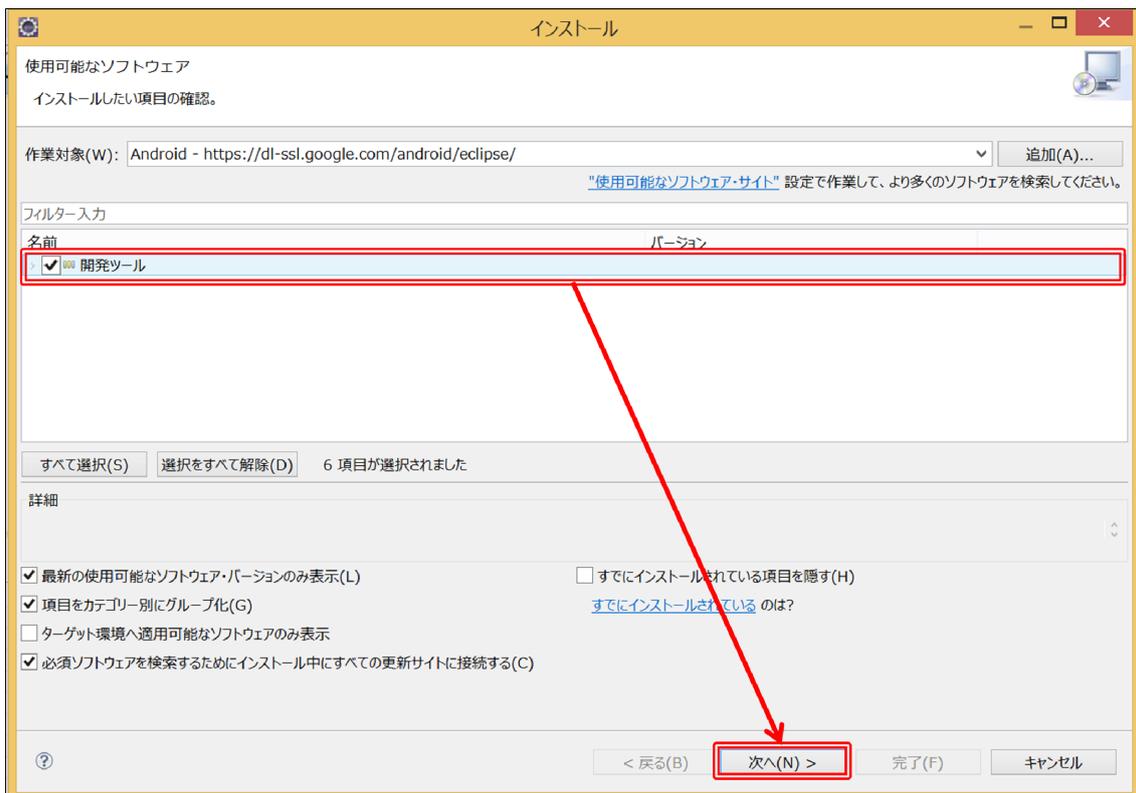
Android アプリケーション開発に必要なソフトウェアをインストールします。Eclipse を起動し、「ヘルプ」から「新規ソフトウェアのインストール」を選択します。



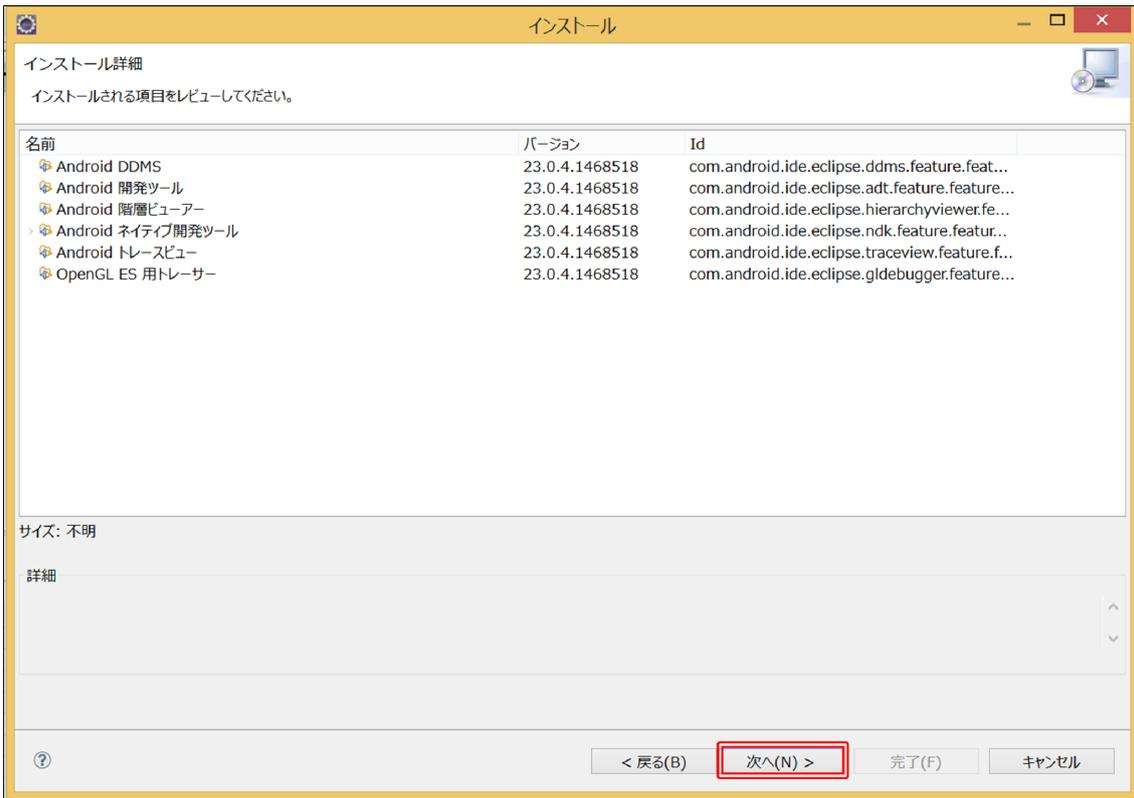
インストール画面で「追加」を選択します。名前は任意、ロケーションには「<https://dl-ssl.google.com/android/eclipse/>」を入力し、「OK」を選択します。



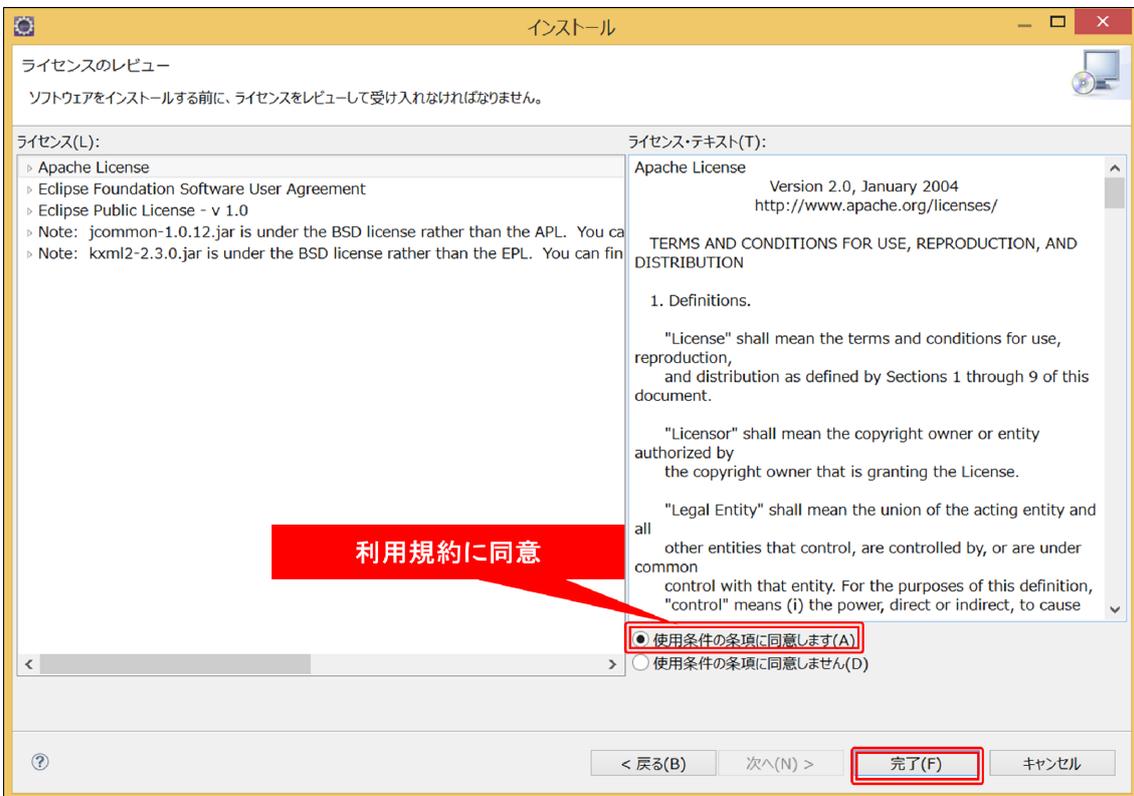
インストール項目の「開発ツール」にチェックを入れて「次へ」を選択します。



インストールされる項目の詳細が表示されるので「次へ」を選択します。



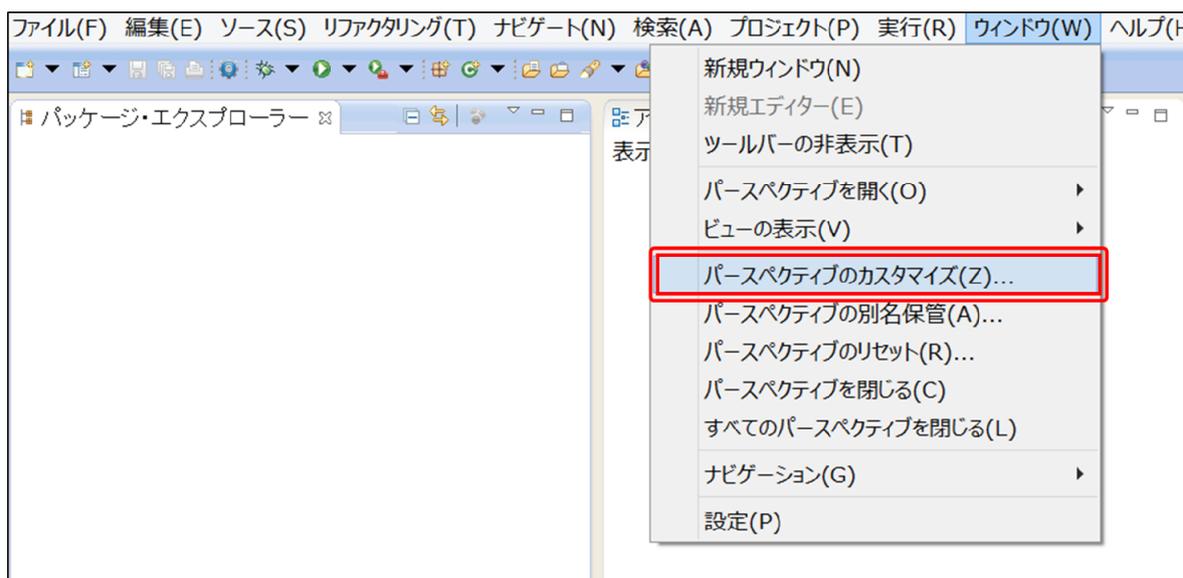
利用規約に同意して「完了」を選択すると、インストールが開始されます。



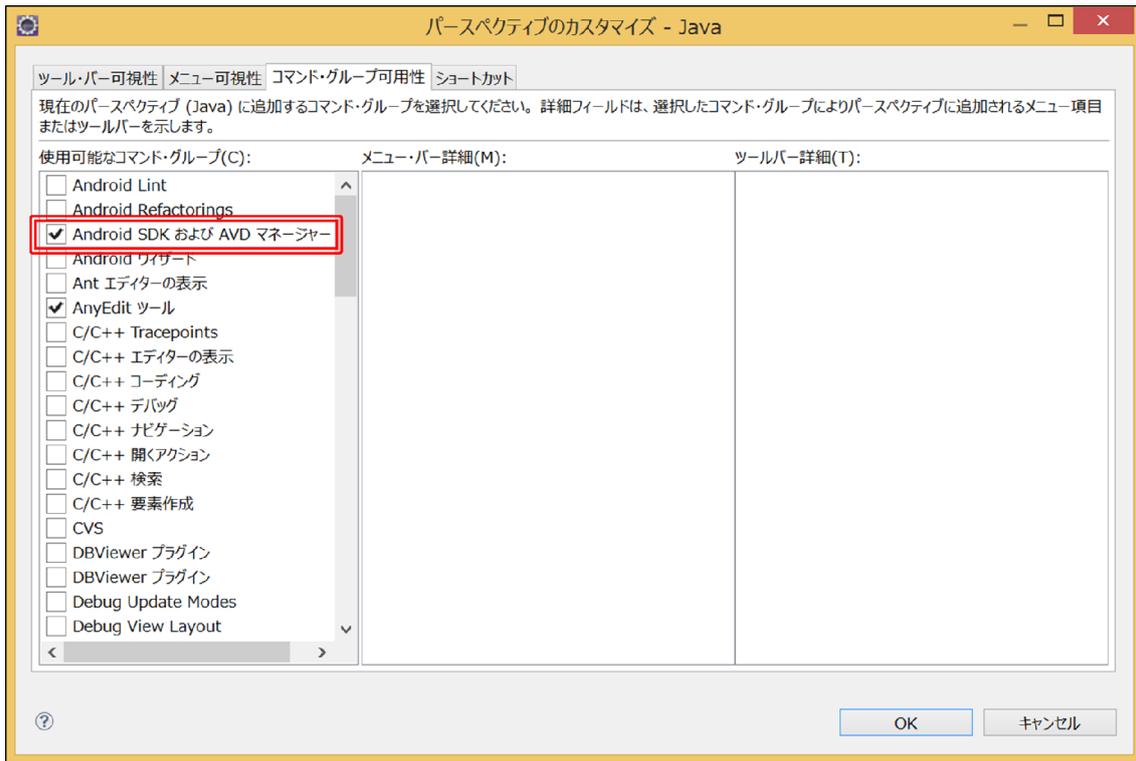
インストール完了後、Eclipse を再起動してください。

2. Eclipse の設定

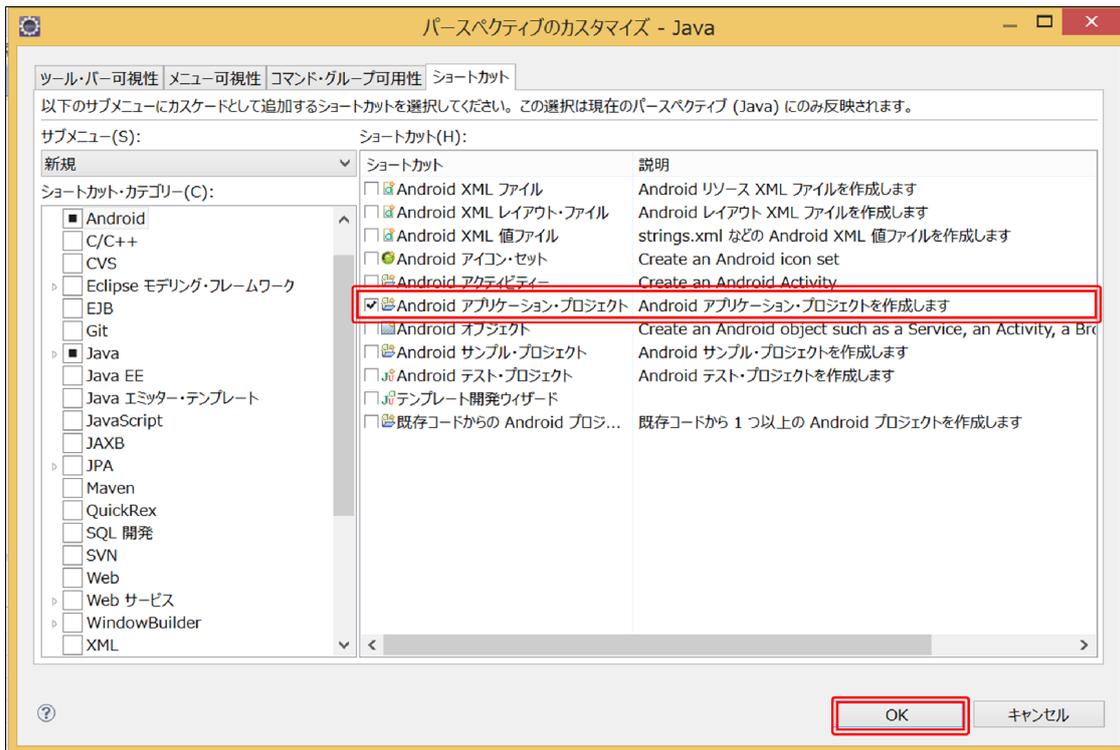
Android アプリケーションの開発をしやすいようにするため、本書では次の設定を行うことを前提としています。「ウィンドウ」から「パースペクティブのカスタマイズ」を選択します。



まず、Eclipse のツールバーに「Android SDK マネージャー」と「Android 仮想デバイス・マネージャー」のアイコンを表示する設定として、「コマンド・グループ可用性」タブを選択し、使用可能なコマンド・グループ内の「Android SDK および ADV マネージャー」にチェックを入れます。



次に、「新規」のショートカットに「Android アプリケーション・プロジェクト」を追加する設定として、「ショートカット」タブを選択し、ショートカット・カテゴリーの「Android」内にある「Android アプリケーション・プロジェクト」にチェックを入れます。



設定完了後、「OK」を選択して設定を反映します。

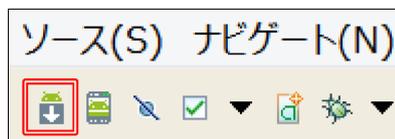
ここまです、Android アプリケーション開発環境を整えるために必要な準備です。

<SDK マネージャー>

SDK マネージャーでは、アプリケーションの動作に必要なファイルなどをインストールすることができます。

①SDK マネージャーの起動

Eclipse を起動し、ツールバーの「Android SDK マネージャー」アイコンを選択すると SDK マネージャーが起動します。



②必要なファイルの選択

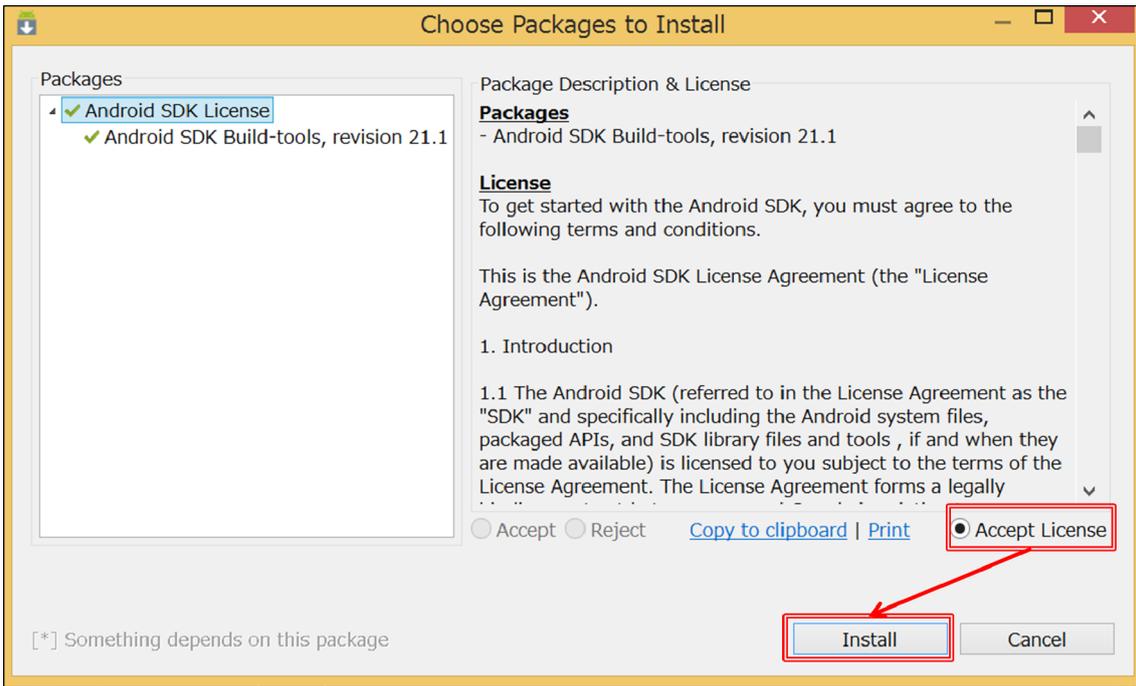
インストールできるファイルの一覧が表示されるため、必要なものにチェックを入れ、「Install ○○ packages」を選択するとインストール画面に進みます。

「Tools」、「Extras」、「Android ○.○(API ○)」をインストールします。全てにチェックを入れても構いませんが、インストールに時間がかかります。

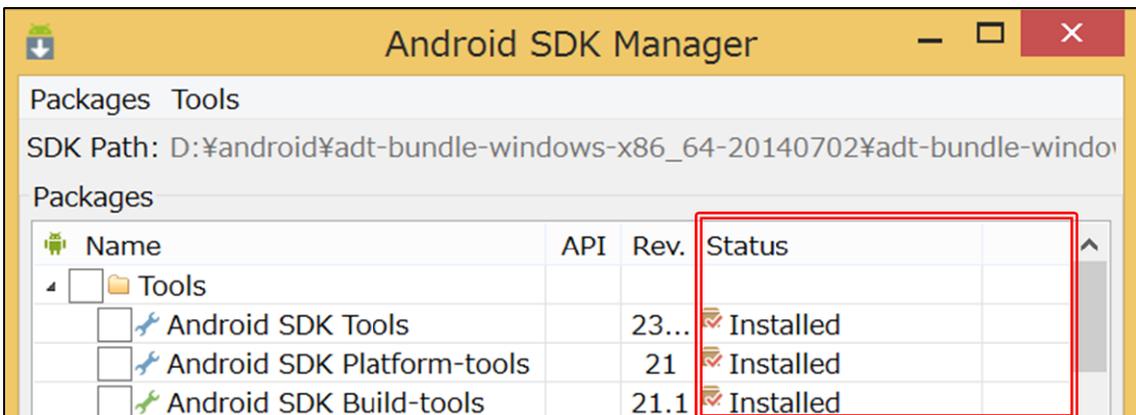
Packages	
	Name
<input checked="" type="checkbox"/>	Tools
<input checked="" type="checkbox"/>	Android 5.0 (API 21)
<input checked="" type="checkbox"/>	Android 4.4W.2 (API 20)
<input checked="" type="checkbox"/>	Android 4.4.2 (API 19)
<input checked="" type="checkbox"/>	Android 4.3.1 (API 18)
<input checked="" type="checkbox"/>	Android 4.2.2 (API 17)
<input checked="" type="checkbox"/>	Android 4.1.2 (API 16)
<input checked="" type="checkbox"/>	Android 4.0.3 (API 15)
<input checked="" type="checkbox"/>	Android 4.0 (API 14)
<input checked="" type="checkbox"/>	Android 3.2 (API 13)
<input checked="" type="checkbox"/>	Android 3.1 (API 12)
<input checked="" type="checkbox"/>	Android 3.0 (API 11)
<input checked="" type="checkbox"/>	Android 2.3.3 (API 10)
<input checked="" type="checkbox"/>	Android 2.2 (API 8)
<input checked="" type="checkbox"/>	Android 2.1 (API 7)
<input checked="" type="checkbox"/>	Android 1.6 (API 4)
<input checked="" type="checkbox"/>	Android 1.5 (API 3)
<input checked="" type="checkbox"/>	Extras

③ 利用規約の承認とインストール

確認画面が表示されるため、「Accept License」を選択することでインストールできるようになります。「Install」を選択するとインストールが始まります。



インストール後、SDK マネージャーの Status の項目が「Installed」になります。

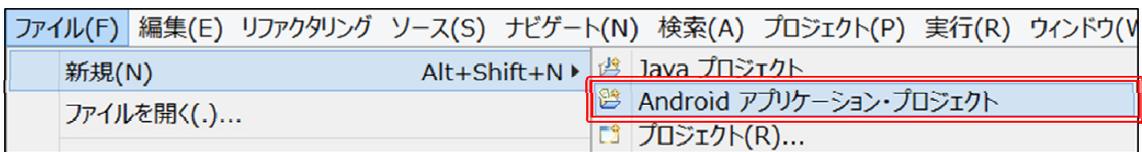


新規プロジェクトの作成

プロジェクトとは、1つのアプリケーションの動作に必要な画像ファイル、プログラムや設定ファイルなどをひとまとめにした単位のことです。

<新規プロジェクトの作成>

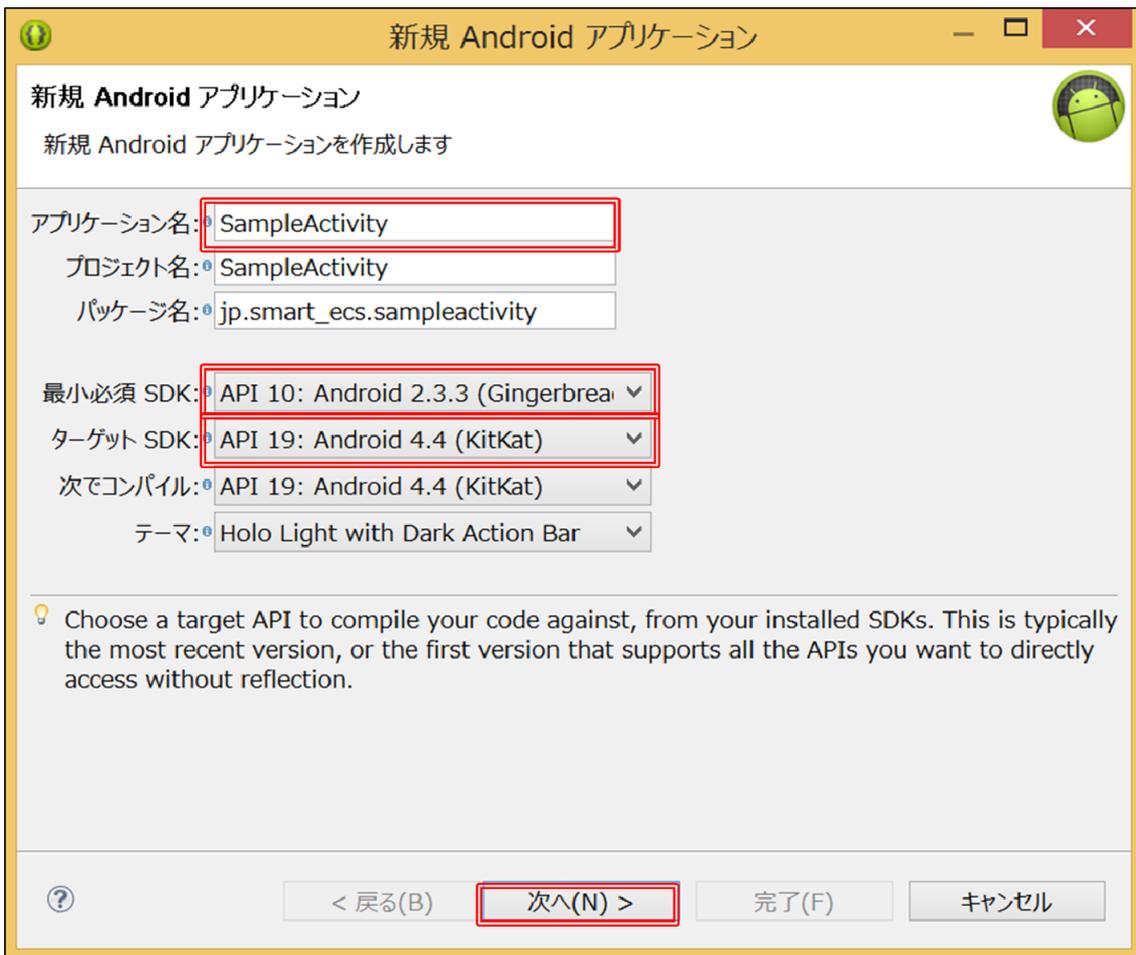
「ファイル」から「新規」を選択し、「Android アプリケーション・プロジェクト」を選択します。



<アプリケーション名の設定>

アプリケーション名を入力し、SDKを設定します。最小必須 SDK は API 10、ターゲット SDK は API 19 を選択し、「次へ」を選択します。

※本書ではパッケージ名を「jp.smart_ecs.〇〇」としています。



項目名	意味
最小必須 SDK	動作するために必要最低限の Android SDK バージョン
ターゲット SDK	対象とする Android SDK バージョン
次でコンパイル	コンパイルする Android SDK バージョン
テーマ	画面の ActionBar の設定

※Android SDK とは、Android アプリケーションを開発するための開発環境のことです。

※コンパイルとは、プログラミング言語を用いて作成したソースコードをコンピュータ上で実行可能な形式に変換することです。

※ActionBar とは、画面上にあるアクティビティ用のウィジェットのことで、デフォルトで Android3.0 以上に搭載されています。

ActionBar の設定	デザイン
None	背景：白、ActionBar：なし
Holo Dark	背景：黒、ActionBar：黒
Holo Light	背景：白、ActionBar：白

Holo Light with Dark Action Bar	背景：白、ActionBar：黒
---------------------------------	------------------

<プロジェクトの構成設定>

必要項目を選択し、「次へ」を選択します。



項目	意味
カスタム・ランチャ・アイコンを作成する	アイコンの設定
アクティビティの作成	アクティビティの設定
このプロジェクトをライブラリとしてマークする	プロジェクトをライブラリに設定
ワークスペース内にプロジェクトを作成	プロジェクトの保存先を「ワークスペース」フォルダに設定
ワーキング・セット	複数のプロジェクトをグループに分けて管理

<アイコンの設定>

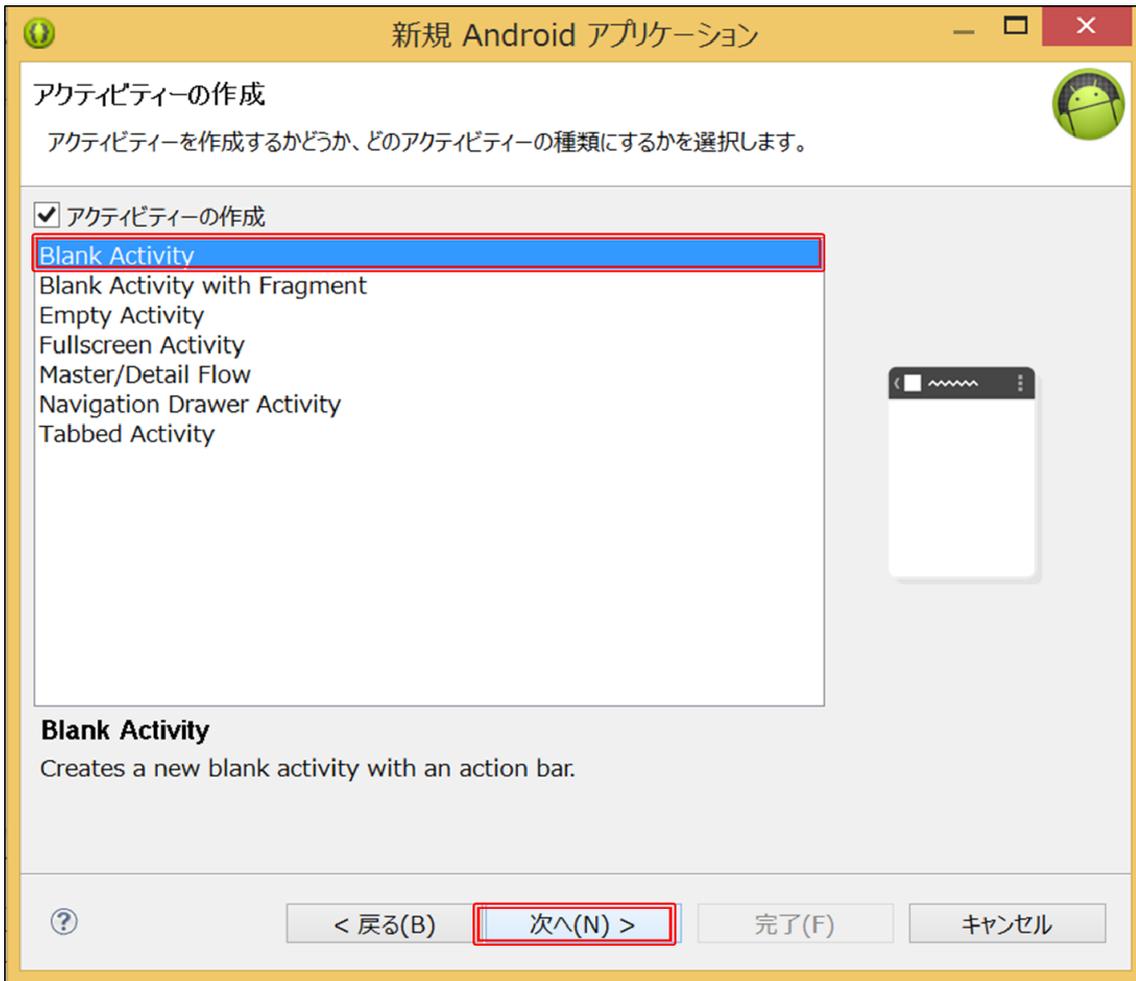
アイコンを設定し、「次へ」を選択します。



- ・「イメージ」を選択した場合、画像をアイコンに設定します。
- ・「クリップアート」を選択した場合、クリップアートをアイコンに設定します。
- ・「テキスト」を選択した場合、文字をアイコンに設定します。

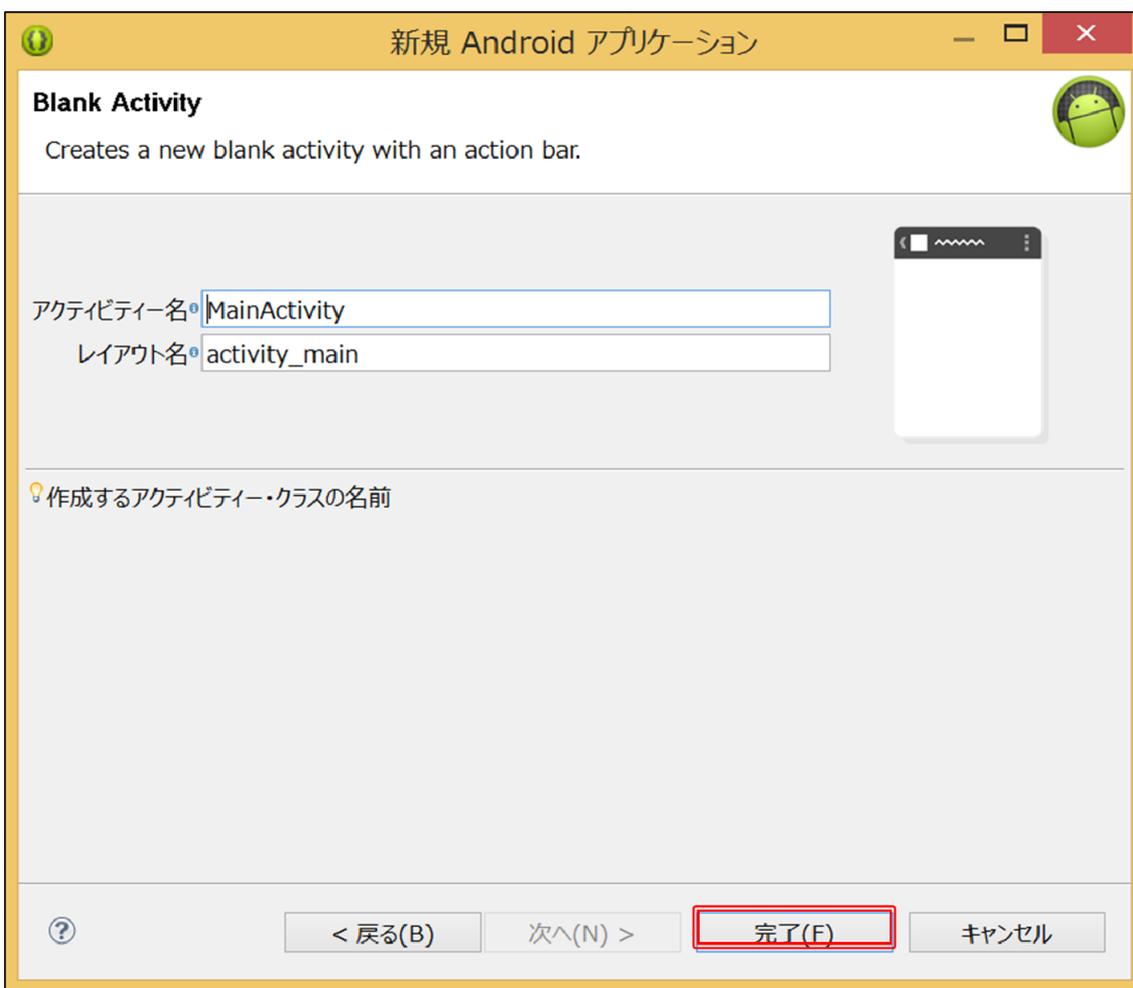
<アクティビティの設定>

アクティビティは、Android アプリケーションの画面に相当します。「BlankActivity」を選択し、「次へ」を選択します。BlankActivity とは、画面に TextView が配置されている雛形のことです。



<アクティビティ名の設定>

アクティビティ名を入力し、「完了」を選択すると新規プロジェクトが作成されます。



項目	意味
アクティビティ名	初期画面のアクティビティ名
レイアウト名	初期画面の画面設定ファイル名

<最新版の開発環境について>

本書に掲載しているアプリケーションは、最新版の開発環境（2014年10月現在）で開発を行っているものではありません。最新版を使って新規プロジェクトの作成を行う場合、自動で生成されるプロジェクトの内容が一部異なります。本書に掲載しているソースコードが動作する環境を構築する手順を説明します。

<最新版の開発環境の特徴>

- ・最新版で新規プロジェクトを作成したとき、新規プロジェクトと「appcompat_v7」が作成されます。

※appcompat_v7とは、Android OS 2.Xの端末でもActionBarを使えるようにするための

プロジェクトのことです。

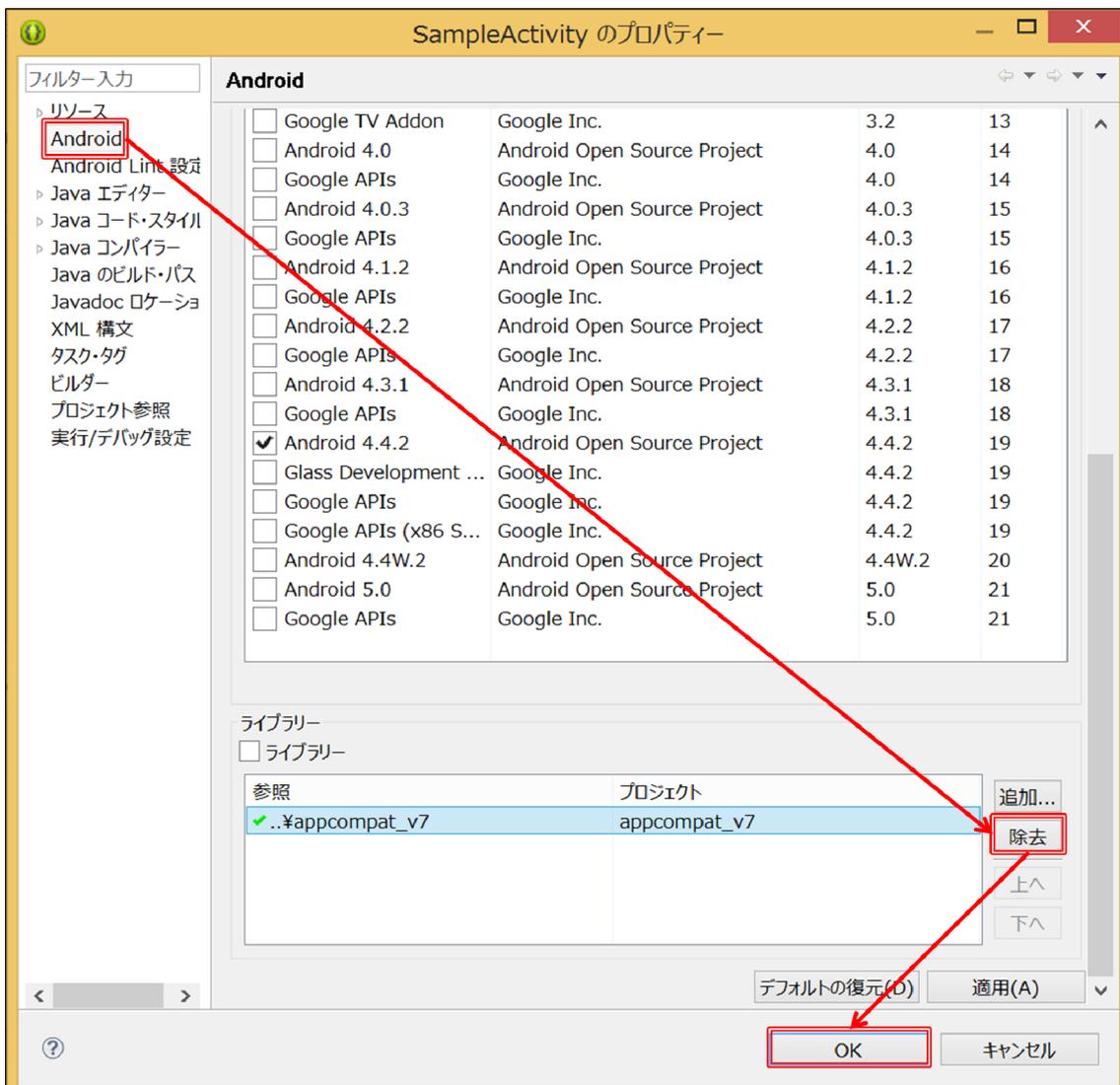
・最新版で作成したプロジェクトは「appcompat_v7」を参照しているため、参照を外す必要があります。

① appcompat_v7 の参照を外す

・プロジェクト名を右クリックし、「プロパティ」を選択します。



・「Android」を選択します。ライブラリとして「appcompat_v7」を参照しているので「appcompat_v7」を選択してから「除去」を選択し、「OK」を選択します。



② MainActivity を変更

- import 文に追加するクラスを、「android.support.v7.app.ActionBarActivity」から「android.app.Activity」に変更します。
- 継承を ActionBarActivity から Activity に変更します。
- onOptionsItemSelected メソッドから return 文以外を削除します。

◆ 変更前

MainActivity.java

```
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends ActionBarActivity {
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

◆変更後

MainActivity.java

```

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        return super.onOptionsItemSelected(item);
    }
}

```

③styles.xml から appcompat_v7 の参照を外す

- ・「res」フォルダ内の「values/styles.xml」、「values-v11/styles.xml」、「values-v14/styles.xml」を編集します。
- ・「styles.xml」にある「style」タグの「parent」の値を変更します。

1. 「Theme.AppCompat.Light」 → 「android:Theme.Light」

◆変更前

values/styles.xml

```
<resources>
    <style name="AppBaseTheme" parent="Theme.AppCompat.Light">
    </style>
    <style name="AppTheme" parent="AppBaseTheme">
    </style>
</resources>
```

◆変更後

values/styles.xml

```
<resources>
    <style name="AppBaseTheme" parent="android:Theme.Light">
    </style>
    <style name="AppTheme" parent="AppBaseTheme">
    </style>
</resources>
```

2. 「Theme.AppCompat.Light」 → 「android:Theme.Holo.Light」

◆変更前

values-v11/styles.xml

```
<resources>
    <style name="AppBaseTheme" parent="Theme.AppCompat.Light">
    </style>
</resources>
```

◆変更後

values-v11/styles.xml

```
<resources>
    <style name="AppBaseTheme" parent="android:Theme.Holo.Light">
    </style>
</resources>
```

```
</resources>
```

3. 「Theme.AppCompat.Light.DarkActionBar」 →

「android:Theme.Holo.Light.DarkActionBar」

◆変更前

values-v14/styles.xml

```
<resources>
    <style name="AppBaseTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    </style>
</resources>
```

◆変更後

values-v14/styles.xml

```
<resources>
    <style name="AppBaseTheme" parent="android:Theme.Holo.Light.DarkActionBar">
    </style>
</resources>
```

④オプションメニューの編集

・「main.xml」にある「menu」タグと「item」タグの一部を削除します。

◆変更前

main.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="jp.smart_ecs.sampleactivity.MainActivity" >
    <item
        android:id="@+id/action_settings"
        android:orderInCategory="100"
        android:title="@string/action_settings"
        app:showAsAction="never"/>
</menu>
```

◆変更後

main.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:title="@string/action_settings"/>
</menu>
```

</menu>

新規プロジェクト内の項目

プロジェクト内の項目の中で、特に重要な部分について説明します。

The image shows a screenshot of the Android Studio interface displaying the project structure for 'SampleApp'. The structure is as follows:

- SampleApp
 - src
 - jp.smart_ecs.sampleapp
 - ① MainActivity.java
 - gen [Generated Java Files]
 - Android 4.4.2
 - Android Private Libraries
 - assets
 - bin
 - libs
 - res
 - ② drawable-hdpi
 - drawable-ldpi
 - drawable-mdpi
 - drawable-xhdpi
 - drawable-xxhdpi
 - layout
 - ③ activity_main.xml
 - menu
 - ④ main.xml
 - values
 - values-v11
 - values-v14
 - values-w820dp
 - AndroidManifest.xml
 - ic_launcher-web.png
 - proguard-project.txt
 - project.properties

Callouts on the right side of the image explain the importance of these items:

- ① MainActivity.java: アクティビティの動作を設定できます
- ② drawable-hdpi: 画像を格納するフォルダです
- ③ activity_main.xml: オブジェクトの設定ができます
- ④ main.xml: オプションメニューの設定ができます
- ⑤ AndroidManifest.xml: アプリケーション全体の設定ができます

<各項目の説明>

①src/jp.smart_ecs.sampleapp/MainActivity.java

- ・ jp.smart_ecs : パッケージ名
- ・ MainActivity.java

アクティビティの動作（オブジェクトの動作など）を設定できます。

②res/drawable-hdpi

アイコンなどアプリケーション内で用いる画像を格納するフォルダです。

③res/layout/activity_main.xml

- ・「layout」フォルダ

画面レイアウトの XML ファイルを格納するフォルダです。

- ・ activity_main.xml

View の配置や大きさなどを設定することができます。この項目を選択したときに画面下部に「グラフィカル・レイアウト」と「activity_main.xml」のタブが表示されます。「グラフィカル・レイアウト」は視覚的に、「activity_main.xml」は言語的に画面レイアウトの作成を行うことができます。

※詳しくは、第 1 章第 2 節「画面レイアウト」で説明します。

④res/menu/main.xml

- ・「menu」フォルダ

オプションメニューの XML ファイルを格納するフォルダです。

- ・ main.xml

オプションメニューの配置やテキストなどを設定することができます。

※詳しくは、第 1 章第 3 節「オプションメニュー」で説明します。

⑤AndroidManifest.xml

アプリケーション全体の設定ができます。

※詳しくは、第 1 章第 2 節「AndroidManifest.xml への追加」で説明します。

仮想デバイスを用いたプロジェクトの実行

プロジェクトの実行は、仮想デバイスを用いた実行方法と実際の端末を用いた実行方法の2種類があります。

仮想デバイスを用いてプロジェクトを実行する場合、新規仮想デバイスを作成する必要があります。

<新規仮想デバイス作成方法>

①Android 仮想デバイス・マネージャーの起動

「Android 仮想デバイス・マネージャー」を選択します。

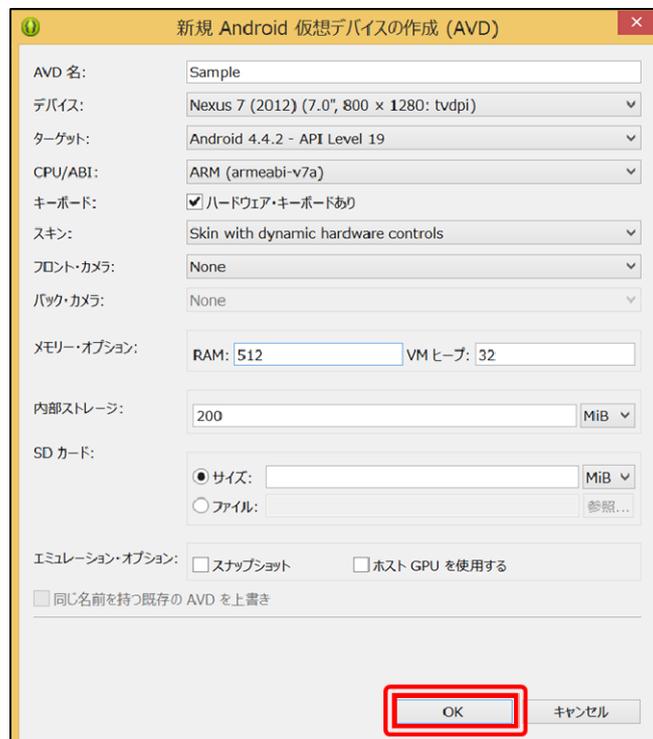


②Android 仮想デバイス・マネージャー

「作成」を選択します。

③新規 Android 仮想デバイスの作成

「AVD 名」、「デバイス」、「ターゲット」、「メモリ・オプション」内の「RAM」を設定します。



項目	意味
----	----

AVD 名	作成する仮想デバイスの名前
デバイス	仮想デバイスの機種選択
ターゲット	Android SDK バージョンの選択
メモリ・オプション	仮想デバイスのメモリ容量
RAM	メインメモリのサイズ

④作成した仮想デバイスの確認

「AVD 名」項目に作成した仮想デバイスが表示されます。



<仮想デバイスを用いた実行方法>

「実行ボタン」を選択し、プロジェクトを実行します。

端末を用いたプロジェクトの実行

端末を用いてプロジェクトを実行する場合、端末側と PC 側の設定が必要です。

< 端末側の設定 >

「提供元不明のアプリ」と「USB デバッグ」にチェックを入れます。

(例) 「GALAXY SIII α」の場合

「設定」→「セキュリティ」→「提供元不明のアプリ」

「設定」→「開発者向けオプション」→「デバッグ」→「USB デバッグ」



「提供元不明のアプリ」にチェック

「USB デバッグ」にチェック

< PC 側の設定 >

接続する端末のデバイスドライバをインストールする必要があります。端末によってデバイスドライバのインストール方法が異なるため、各メーカーのサイトから端末のドライバをダウンロードしてインストールしてください。

< 端末を用いた実行方法 >

「実行ボタン」を選択し、プロジェクトを実行します。端末を接続した状態で実行すると、「実行中の Android デバイスを選択する」項目に接続されている端末の名前が表示されます。端末名を選択し、「OK」を選択すると実行できます。

プロジェクトのインポート

ワークスペース外に既存のプロジェクトがある場合は、「ファイル」から「インポート」を選択し、既存のプロジェクトを **Eclipse** にインポートします。

本書では、プリクラアプリと名刺管理アプリでこの作業が必要です。

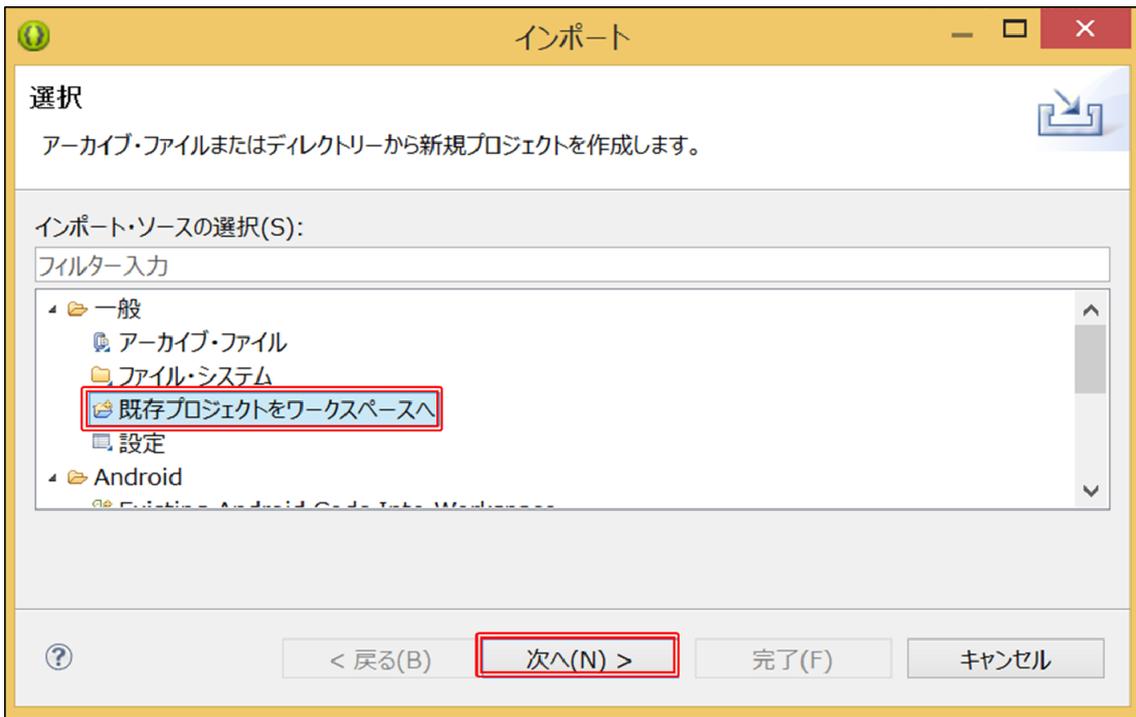
①インポートの選択

「ファイル」から「インポート」を選択します。



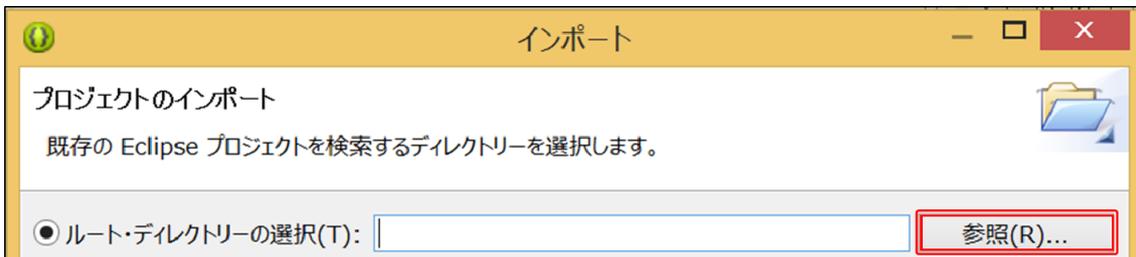
②インポート・ソースの選択

「一般」内の「既存プログラムをワークスペースへ」を選択し、「次へ」を選択します。



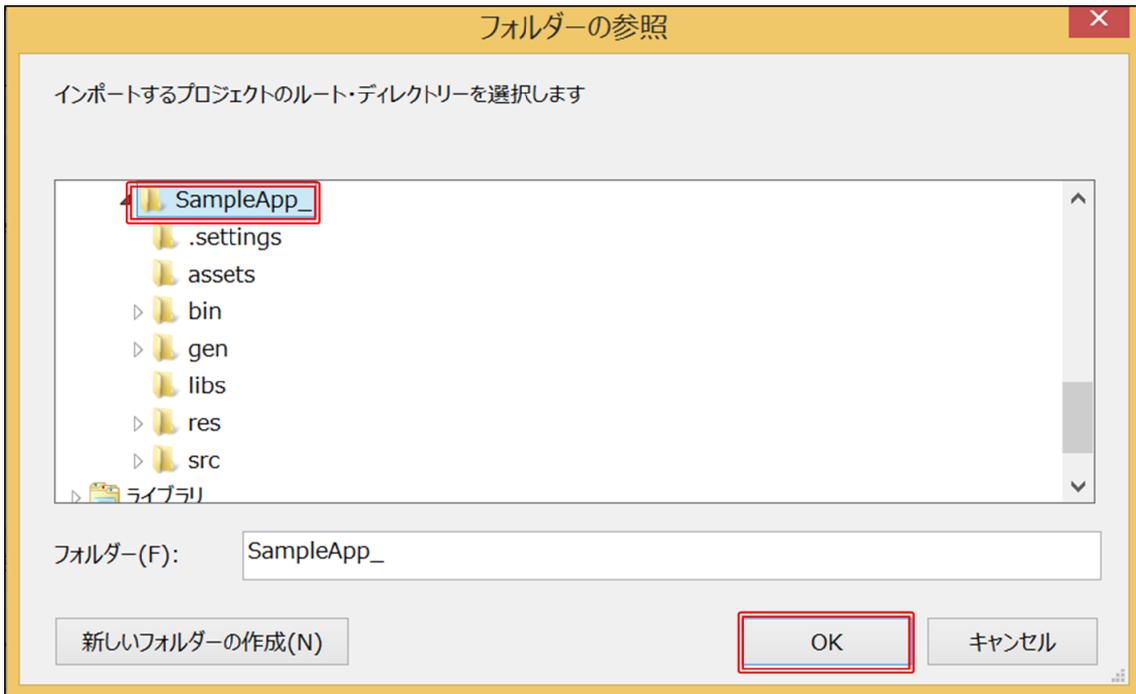
② ルートディレクトリの選択

「ルートディレクトリの選択」から「参照」を選択します。



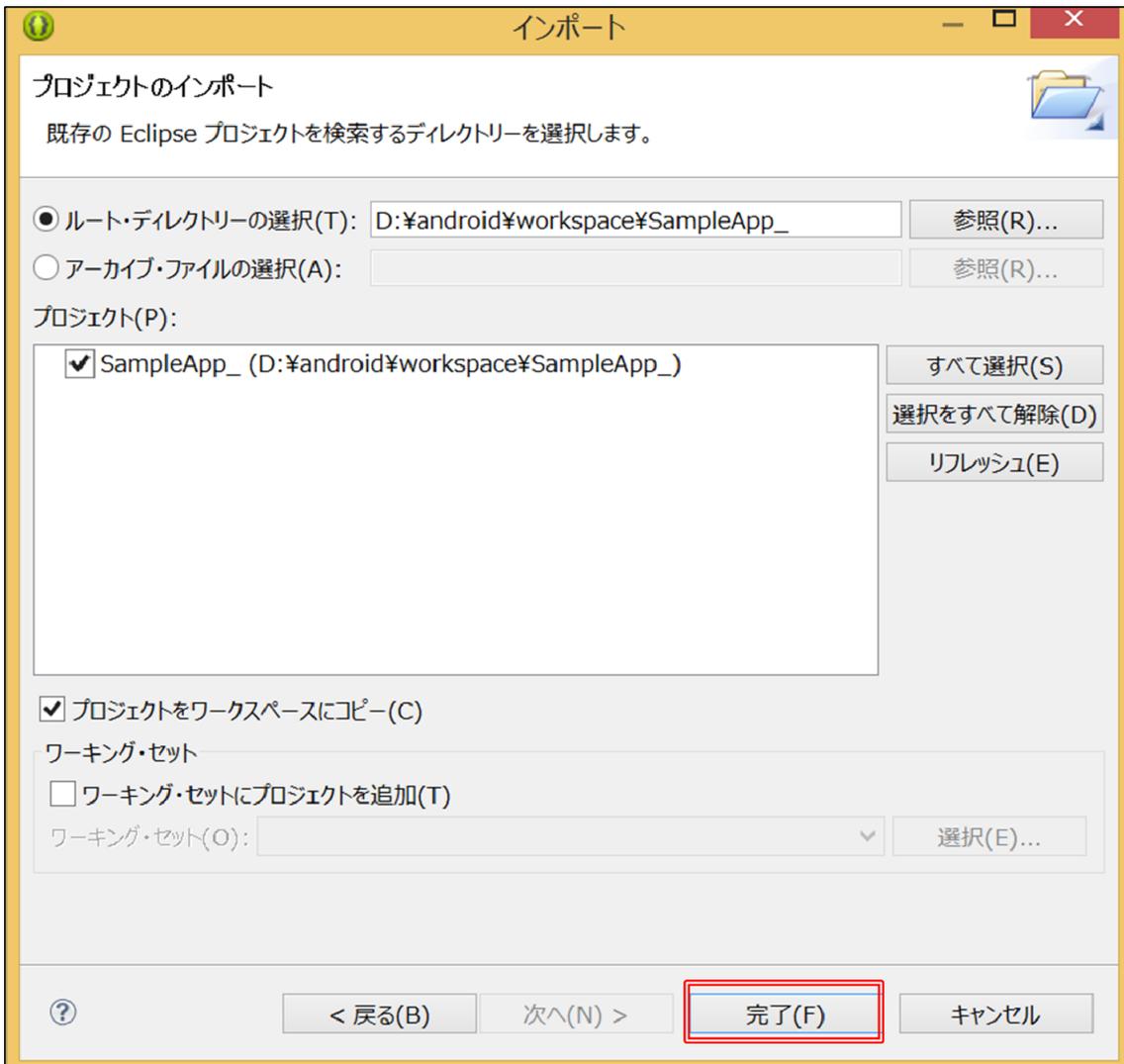
③ フォルダの参照

インポートするプロジェクト名を選択し、「OK」を選択します。



④ インポートの完了

「プロジェクト」項目からインポートするプロジェクト名を選択し、「完了」を選択します。



※プロジェクトをワークスペース内に保存する場合、「プロジェクトをワークスペースにコピー」を選択します。

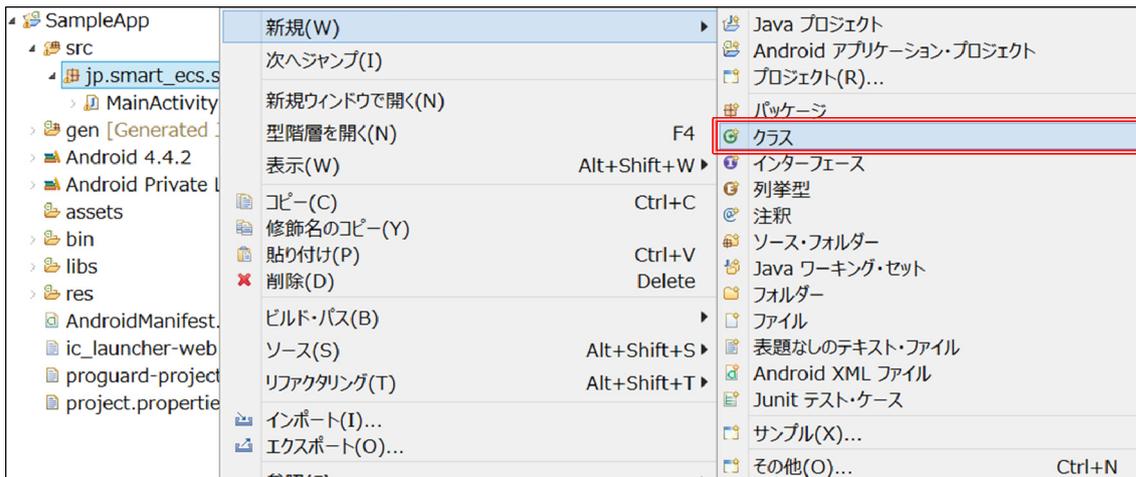
(補足) 新規ファイルの作成とアプリへの追加

新しくファイルを作成してプロジェクトに追加する方法について説明します。

<Java ファイルの場合>

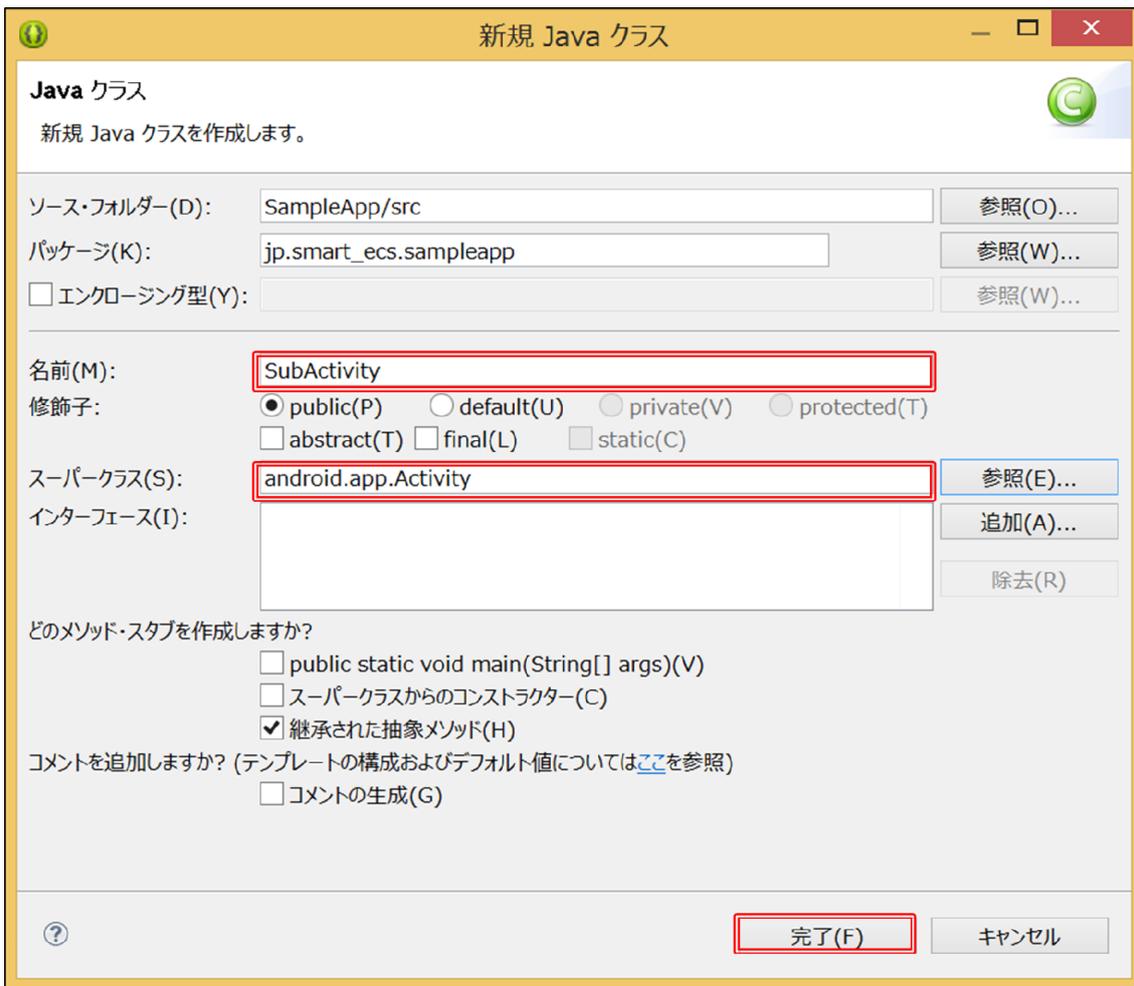
① 新規 Java クラスの作成

パッケージ名を右クリックし、「新規」から「クラス」を選択します。



ソース・フォルダとパッケージは自動で入力されるため、作成するクラスの名前とスーパークラス（継承元クラス）を入力し、「完了」を選択すると、新規 Java ファイルが作成されます。

(例) 名前 : SubActivity、スーパークラス : android.app.Activity を指定



② AndroidManifest.xml にコードを追加

アクティビティを追加するには、次のコードを追加します。

AndroidManifest.xml

```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".MainActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

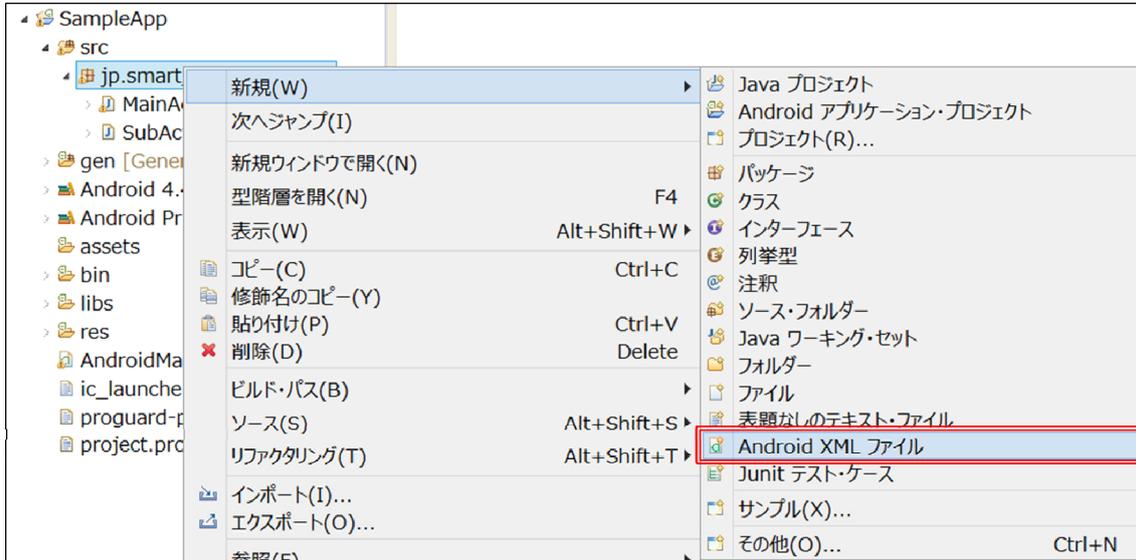
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=" SubActivity">
    </activity>
```

```
</application>
```

<XML ファイルの場合>

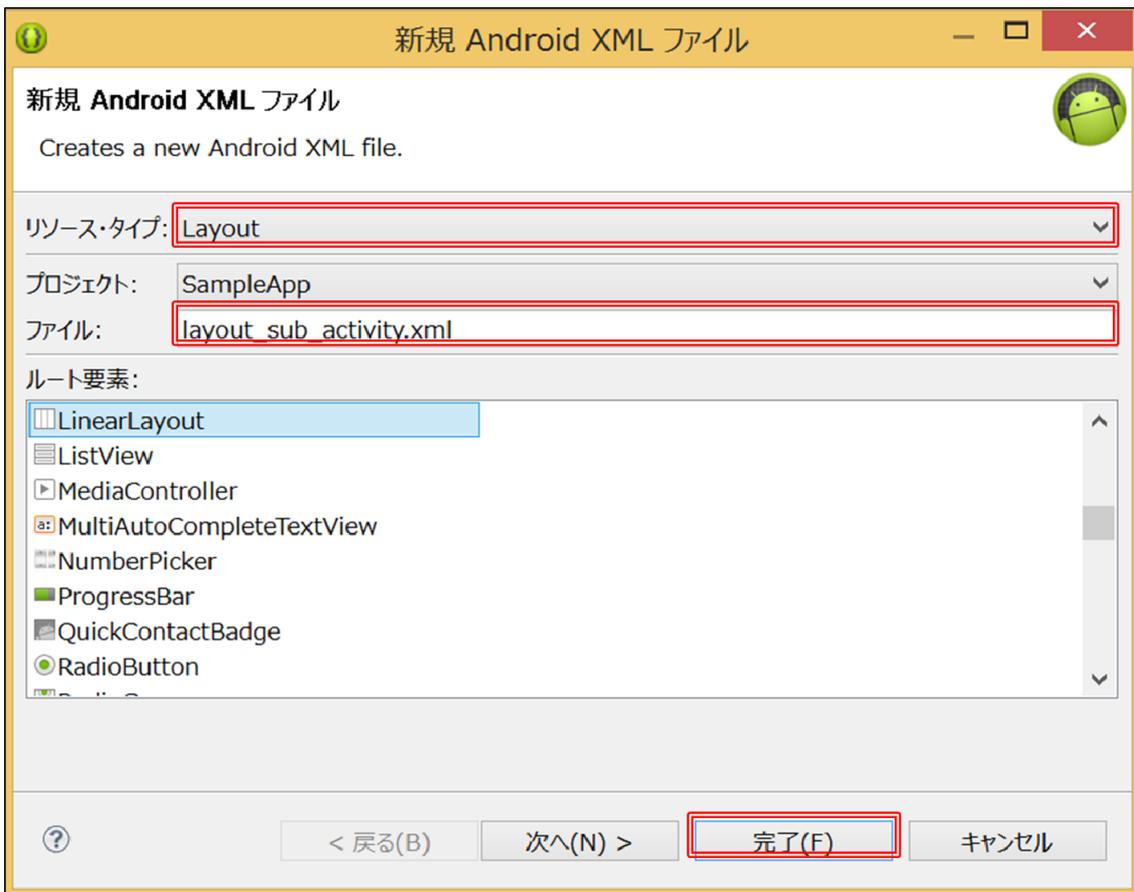
① 新規 XML ファイルの作成

パッケージ名を右クリックし、「新規」から「Android XML ファイル」を選択します。



② リソース・タイプの選択

作成する XML ファイルのリソース・タイプを選択し、ファイル名を入力して「完了」を選択すると XML ファイルが作成されます。



- レイアウトの場合
リソース・タイプ「Layout」を選択し、ルート要素としてレイアウトの種類を選択します。
- メニューの場合
リソース・タイプ「menu」を選択し、ルート要素として「menu」を選択します。
- 文字列・色などのリソース定義するファイルの場合
リソース・タイプ「Values」を選択し、ルート要素として「resources」を選択します。