

Androidアプリ塾

概要編

Androidアプリの概要

Androidとは

- モバイルプラットフォームの1つ
- 主にスマートフォンやタブレット端末に搭載される
- CPUやメモリ、ストレージなどは一般的にPCより劣る

Androidアプリの特徴

- カメラや位置情報、加速度センサーなどを制御できる
- スマホなど、小さな画面で動作を想定
- 画面全体を1つのアプリが占有
- ユーザーがホームボタンを押しても、アプリは停止しない
- OSにより強制的に終了されることがある

Androidアプリの要素

<パッケージ名>

- Androidアプリを一意に識別する
 - 他の開発者のアプリと重複しないようにする
- 慣例として、ドメイン名を逆にして命名する
例) jp.co.iseise.edabee.launcher
- Google Play ストアのURLの一部になる
- 他のアプリのパッケージ名を知ることで、アプリ間の連携が可能となる

Androidアプリの要素

<Activity>

- Androidアプリの1画面を制御する
- Activityの開始、終了、他Activityのバックグラウンドに隠れるなどのライフサイクルに応じた処理をコールバックメソッドに記述する

スタック

- 画面が切り替わる仕組み

(参考) http://qiita.com/yuya_presto/items/ab2162078e5d5076c718

ライフサイクルと主なコールバックメソッド

- onCreate、onResume、onPause、onDestroyなど

(参考) <http://www.javadrive.jp/android/activity/index2.html>

Androidアプリの要素

<View>

- Androidアプリの画面を表現するUI部品
- 画面レイアウトを構成する以下の部品は全てView
 - LinearLayoutなどのコンテナ
 - テキストボックス
 - ボタン
 - その他、画面上で表示されるUI要素
- イベントを処理できる
 - ボタンがクリックされた
 - 画面がタッチされた
 - 文字が入力された

Androidアプリの要素

<Service>

- 時間のかかる処理を行う
- バックグラウンドServiceとフォアグラウンドService
- OSにより強制的に終了されることがある
 - メモリなどの条件が改善された場合再起動されることがある
 - 強制終了を前提にした設計が望ましい

(利用例)

- ファイルなどのダウンロード
- 音声データなどの再生
- バッテリー監視・記録

Androidアプリの要素

<Service>(続き)

ライフサイクルと主なコールバックメソッド

- onCreate
- onStartCommand
- onBind
- onDestroy

Androidアプリの要素

<Widget>

- ホーム画面に配置され、何らかの有用な表示やアクションを行う

(利用例)

- 時計Widget
- カレンダーWidget
- 天気予報Widget

Androidアプリの要素

<BroadcastReceiver>

- Androidシステムやアプリから発せられるBroadcastを受信して何らかの処理を行う

(利用例)

- OS起動時に処理を行う
- バッテリー残量が少なくなった場合に処理を行う
- Serviceからの通知を受け、Activityを起動する

(参考)Broadcast Actionsの一覧

<http://hyoromo.hatenablog.com/entry/20091002/1254498107>

Androidアプリの要素

<Intent>

- Activity間や他アプリとの連携に使われる
- 明示的インテントと暗黙的インテントがある

(利用例)

- 画面間のデータの受け渡しに使用
- BroadcastReceiverでのデータの受け取りに使用
- Serviceへのデータの受け渡しに使用
- 他アプリ(メーラー、ブラウザ、Twitterなど)へのデータの受け渡し

Androidアプリの権限

権限の宣言

- マニフェストファイルにアプリが使用する権限を宣言
- アプリに無関係な権限は使用してはならない
- ユーザーはアプリダウンロード時にアプリが使用する権限を確認できる

<主な権限>

- インターネットアクセス
- SDカードアクセス
- アカウント情報へのアクセス
- 連絡先データへのアクセス

アプリのローカライズ(多言語対応)

文字列のリソース化

- メッセージやガイド、ボタンに表示する文字列などはプログラムに直接記述せず、XMLのリソースファイルに記述する
- リソースファイルは対応する言語毎に作成する

ロケールを意識して実装する

- 日付・時刻、大文字・小文字変換などはロケールを指定して行う

Fragmentとタブレット対応

Fragment

- 画面をFragmentという単位で定義することにより、柔軟かつ効率的なUIが実現可能
- タブによる画面切り替え
- ViewPagerによる画面切り替え
- 別ActivityでのFragmentの再利用

タブレット対応

- 一覧画面、詳細画面をそれぞれFragmentで定義し、タブレットでは並べて表示することも可能

アプリの公開と課金

Google Play ストア

- 開発したアプリを配布するオンラインストア
- 開発者登録(初期費用25ドル)を行えばストアにアプリを公開できる

<無料アプリ>

- 広告を表示して収入を得ることも可能

<有料アプリ>

- Googleの決済システムを使用し有料アプリを公開可能

<アプリ内課金>(消費型、非消費型、定期購読型)

- 追加アイテムなどに課金することが可能

開発環境と必要な技術

<開発環境>

- JDK 1.7以降
- Android SDK
- Android Studio または Eclipse + ADTプラグイン

※ADTプラグインは2015年末でサポート終了

<必要な技術>

- Java言語
- SQLite(一般的なSQLの知識)
- XML(画面レイアウトや設定)