
SQUARE ENIX.
不動の人気タイトルを軸に
"新しいゲーム体験" を創り出す！
IT業界に特化した求人サイト Green

191 LGTM 0

@kojionilk 2014年07月16日に更新

Bundle

Android

⚠ この記事は最終更新日から5年以上が経過しています。

Bundle とは

Android アプリ開発のさまざまなところに出てくる **オブジェクトの入れ物** である。本当に頻出するので Bundle の扱いに慣れているといろいろと捗る。以下例を挙げる。

例1: Activity の状態保存

画面回転や長時間放置してシステムに殺された場合などは `onSaveInstanceState()` で状態保存して `onRestoreInstanceState()` で復帰する。主に Activity のフィールドやカスタムビューなどは状態を復元する必要がある。

```
public class MyActivity extends Activity {
    @Override
    protected void onSaveInstanceState(Bundle outState) {
        outState.putString("hoge", "hoge");
        outState.putInt("fuga", 3);
        outState.putBoolean("is_hage", true);
    }

    @Override
    protected void onRestoreInstanceState(Bundle savedInstanceState) {
        savedInstanceState.getString("hoge");  // hoge
        savedInstanceState.getInt("fuga");  // fuga
        savedInstanceState.getBoolean("is_hage");  // true
    }
}
```

例2: インテント発行時の引数

Android では Activity or Fragment から Activity へ画面遷移を行う際に Intent を以下のように発行する:

```
final Intent intent = new Intent(this, MyActivity.class);
intent.putExtra("hoge", "hoge");
intent.putExtra("fuga", 3);
intent.putExtra("is_hage", true);
startActivity(intent);
```

遷移先の Activity で以下のように引数 (Extras) を取り出せる:

```
public class MyActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        getIntent().getStringExtra("hoge");  // hoge
        getIntent().getIntExtra("fuga", -1);  // 3
        getIntent().getBooleanExtra("is_hage", false);  // true
    }
}
```

Intent のインスタンスは内部的に Bundle を持っており、上記 `intent.putExtra()` は実際はその内部の Bundle に対し値を設定しているだけである (Intent のソースを見ると容易に確認できる)。なので以下のように書いても同じだ:

```
Bundle bundle = new Bundle();
bundle.putExtra("hoge", "hoge");
bundle.putExtra("fuga", 3);
bundle.putExtra("is_hage", true);

final Intent intent = new Intent(this, MyActivity.class);
intent.putExtras(bundle);  // 内部的に Bundle.putAll() が呼ばれる
startActivity(intent);
```

例3: Fragment に対する引数

どんな Fragment に対しても使うので DialogFragment に対しても頻繁に使用するだろう:

```
public class MyActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getFragmentManager().findFragmentByTag("blank") == null) {
            final BlankFragment fragment = new BlankFragment();
            final Bundle bundle = new Bundle();
            bundle.putString("hoge", "hoge");
            bundle.putInt("fuga", 3);
            bundle.putBoolean("is_hage", true);
            fragment.setArguments(bundle);

            getFragmentManager().beginTransaction()
                    .add(android.R.id.content, fragment, "blank")
                    .commit();
        }
    }
}
```

他にもまだまだ Bundle を使う箇所はいっぱいあるはずだ。

Java EE 開発でよく使う JavaBeans や POJO は?

JavaBeans は **引数なしのコンストラクタを持ちプロパティに対して setter, getter を提供するもの** で、例えば以下の様なものだ:

```
public class Beans {
    private String hoge = null;
    private int fuga = 0;
    private boolean isHage = false;
    public String getHoge() {
        return hoge;
    }
    public String setHoge(String hoge) {
        this.hoge = hoge;
    }
    // ... 省略 ...
}
```

必要に応じて Serializable を implements する。

POJO は **Plain Old Java Object** の略であり「普通の Java オブジェクト」の意味なのだが、これは **JavaBeans ではないのだが適当な入れ物用のクラス** を作る時によく使われる。例えば以下の様なものだ:

```
public class POJO {
    public final String hoge;
    public final int fuga;
    public final boolean isHage;
    public POJO(String hoge, int fuga, boolean isHage) {
        this.hoge = hoge;
        this.fuga = fuga;
        this.isHage = isHage;
    }
}
```

上記を new した結果の POJO は immutable (不変) オブジェクトとなる。つまり一度 new すると hoge, fuga, isHage を変更することはできない (注: もし hoge, fuga, is_hage のいずれかがプリミティブ型若しくは immutable class でない場合 POJO は immutable ではなくなる)。これは JavaBeans には出来ない芸当だ (いや、setter をコールした際に全て UnsupportedOperationException をコールするようにすれば出来るか?)。JavaBeans は setter / getter があるので値をセット及び取得する際に処理を仕込むことができるのがメリットなのだが、それが有効に使われている場面は少ないように思う。あとやはり setter / getter を介さずに直接フィールドを参照した方がコード上シンプルに纏まるのが大きい。

ただこれは **所謂お固い普通の Java 開発** の場合であり Android でこれらの技術を使用しても良いのだが上記の通り **画面の状態保存や画面遷移の度に Bundle が必要なので詰め直す必要がある。** これが面倒くさい。だったらはじめから Bundle に詰めて管理した方が有利だろう。

注意として Serializable か Parcelable を implements していれば Bundle に直接突っ込めるのでそんなに面倒では無いのだがそれらの実装自体が面倒なので本末転倒に思える……。

意外といろいろ入る Bundle

画面遷移の時に String とか int, boolean あたりを詰めている人が多いかもしれないが実はもっといろいろ入るので知っておくと予めそれを想定したプログラム設計に出来て有利だ。

例えば Bundle には Serializable インターフェースを実装したものを入れられるので以下の様なことができる。配列が入れられるのとか気づきにくいのではないか:

```
final Bundle bundle = new Bundle();
bundle.putSerializable("date", new Date());  // java.util.Date
bundle.putSerializable("calendar", Calendar.getInstance());  // java.util.Calendar

// Java の配列は Serializable なので配列にしてしまえば何でも入れられる ... がいいのか?
bundle.putSerializable("objects", new Object[] {new Object()});
```

あと Bundle には Parcelable インターフェースを実装したものが入れられる。Android で Parcelable を実装しているものはかなり多い:

```
final Bundle bundle = new Bundle();
bundle.putParcelable("intent", new Intent());  // インテントを Bundle に詰める
bundle.putParcelable("lat_lng", new LatLng(0, 0));  // Google Maps Android API v2
bundle.putParcelable("bitmap", BitmapFactory.decodeFile("hoge"));  // 大きいものは止めたほうがい
```

あと **Bundle に Bundle を入れられる** ので全部 Bundle で管理しておけば、例えば画面遷移の時に状態を渡したい場合管理している Bundle をそのまま突っ込むとかできる:

```
Bundle bundle1 = new Bundle();  // 状態 1 を管理しているとする
Bundle bundle2 = new Bundle();  // 状態 2 を管理しているとする

Bundle bundle = new Bundle();
bundle.putBundle("bundle1", bundle1);
bundle.putBundle("bundle2", bundle2);
```

いろいろ便利な Bundle のメソッド

今この Bundle に何が入っているか知りたいといった時は `Bundle#toString()` を使えば人間に分かりやすい形式の文字列に変換してくれる。これはデバッグ時に便利だ:

```
public class MyActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        final Bundle bundle = new Bundle();
        bundle.putString("hoge", "hoge");
        bundle.putInt("fuga", 3);
        bundle.putBoolean("is_hage", true);
        Log.d(getClass().getSimpleName(), bundle.toString());
    }
}
```

```
07-16 16:34:46.273    4270-4270/com.kojion.test D/MyActivity﹕ Bundle[{is_hage=true, hoge=ho
```

後はメソッド名を見ればだいたい想像が付くメソッドばかりなので迷うことはない:

```
final Bundle bundle = new Bundle();
bundle.putString("hoge", "hoge");
bundle.putInt("fuga", 3);
bundle.putBoolean("is_hage", true);

// 指定キーが存在するかのチェック
bundle.containsKey("hoge");  // true
bundle.containsKey("naiyo");  // false

// Bundle が空かどうかのチェック
bundle.isEmpty();  // false
new Bundle().isEmpty();  // true

// キーの数を返す
bundle.size();  // 3

// キーのぶんだけ何かしたい場合は keySet()
for (final String key : bundle.keySet()) {
    final Object object = bundle.get(key);
    // なにかする
}

// 特定のキー削除
bundle.remove("hoge");

// 全クリア
bundle.clear();
```

だがちょっと面倒な気がする

Bundle に値をセットしていくのがちょっと面倒な気がする。なにぶん頻出するので 1 文字でも少なくしたいと思うのが人情だろう。筆者がひとつの解決策を考えたのでそれは次記事を参照して頂きたい。

編集リクエスト ストック LGTM 191

Hideyuki Kojima @kojionilk
Android, Java, PHP あたりのプログラマ。Python の思想が好み。開発環境として OS X や Ubuntu を使う。
http://www.kojion.com/
フォロー


モンストの "SRE" 募集中!
IT業界に特化した求人サイト Green
フルリモートワーク、フルフレックス環境での仕事にチ… 【東京都】大手企… 年収～800万円 詳細情報
まだ、転職するな！最強に面白い会社がココにあります！… 【東京都】弊社は… 年収～1500万円 詳細情報
『ポケットモンスター』シリーズの校正・監修業務 【東京都】・『ポ… 年収～600万円 詳細情報
Green

関連記事 Recommended by LOGLY

Activity/Fragmentの再生成と画面回転に関するまとめ
by anolivetree

DialogFragment
by kojionilk

Bundle を少しだけ簡単に書く
by kojionilk

KotlinでもIcePickが使いたい！
by boohbah

ディープラーニング未経験OK！AIエンジニアにキャリアチェンジ
PR パソナテック

この記事は以下の記事からリンクされています

【Androidアプリ開発勉強】Observerパターンで非同期処理はAsyncTaskLoaderがメジャーらしい からリンク 1 year ago
Googlecalender import パッケージ からリンク 2 years ago
JSONObject を Bundle に変換する からリンク 4 years ago
Bundle を少しだけ簡単に書く からリンク 5 years ago

コメント

この記事にコメントはありません。

コメントを投稿する

編集 プレビュー

コメントを入力

画像を選択 0B / 100MB 投稿

© 2011-2020 Increments Inc.