

『Android プログラミング パーフェクトマスター』

<正誤表>

●96 ページ 書き換え後のコード(Welcome.java)

【訂正】 コード3行目の下に以下の行を追加します。

```
package net.sample.welcome;

import android.app.Activity;
import android.os.Bundle;
import android.view.Window; ←※追加になります

public class Welcome extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(new WelcomeShow(this));
    }
}
```

●354 ページ Activity1.java)

【訂正】 コード下から4〜3行目の間に以下の赤字の行を追加します。

```
public void dialogShow(View v) {

    // Dialog クラスのインスタンスを生成
    varDialog = new Dialog(Activity1.this);
    // ダイアログのインスタンスに設定済みのビューを追加
    varDialog.setContentView(R.layout.dialog_set);
    // ダイアログのタイトルを設定
    varDialog.setTitle("Original Dialog");
    // ダイアログを表示
}
```

●574 ページ XML コードのキャプション

【誤】 ▼Activity.Java

【正】 ▼main.xml

●574 ページ ギャラリーの作成

【訂正】 Activity.Java のソースコードが欠落しています。正しくは以下のソースコードが入ります（ダウンロード提供しているサンプルには問題ありません）。

▼Activity.Java

```
package net.sample11_01_01_05;

import android.app.Activity;
import android.os.Bundle;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ViewFlipper;

public class Activity1 extends Activity {
    private ViewFlipper varViewflipper;
    private float lastTouchX;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

setContentView(R.layout.main);
varViewflipper = (ViewFlipper)this.findViewById(R.id.flipper);

this.varViewflipper.setOnTouchListener(new OnTouchListener() {

    public boolean onTouch(View v, MotionEvent event) {
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN: //A
                lastTouchX = event.getX();
                break;
            case MotionEvent.ACTION_UP: //B
                float currentX = event.getX();
                if (lastTouchX < currentX) {
                    varViewflipper.showNext();
                }
                if (lastTouchX > currentX) {
                    varViewflipper.showPrevious();
                }
                break;
        }
        return true;
    }
});
}
}
}

```

＜Android SDK パッケージの追加について＞

新しい Eclipse において、53 ページの「Android SDK のパッケージの追加」の操作で、メニューに「Android SDK Manager」が表示されない場合があります。この場合は、以下の操作を行います。

- ①Eclipse の[ウィンドウ]メニューをクリックして[パースペクティブのカスタマイズ]を選択します。
- ②[パースペクティブのカスタマイズ]ダイアログボックスが表示されるので、[Android SDK および AVD マネージャー]にチェックを入れます。
- ③[OK]ボタンをクリックします。

もし、これで表示されない場合は、以下の操作を行なってみてください。

- ①[ウィンドウ]メニューをクリックして[すべてのパースペクティブを閉じる]を選択します。
- ②[ウィンドウ]メニューをクリックして、[パースペクティブを開く]→[その他]を選択します。
- ③[パースペクティブを開く]ダイアログボックスが表示されるので、[Java (デフォルト)]を選択します。
- ④[OK]ボタンをクリックします。

＜Eclipse のエラー表示について＞

Eclipse のバージョンによっては、main.xml でボタンなどのウィジェットにテキストを表示する際、android:text 属性で直接、テキストを設定すると次のようなメッセージが表示される場合があります。

▼android:text 属性で直接、テキストを設定する
 android:text="Button"

▼表示されるメッセージ
 Hardcoded string "Button", should use @string resource

【原因】

ウィジェットに表示するテキストは strings.xml で定義するように促しているものと思われます。ただし、9 章で紹介しているレイアウトエディターでボタンを配置した場合も同様のメッセージが表示されることから、これは整合性のない仕様であり、混乱の元になってしまっています。

【対策】

プログラム自体は問題なく実行することができますが、メッセージが表示されないようにするには、次のように操作します。

- strings.xml において、ウィジェットに表示するテキストを、識別用の ID と一緒に登録する。

例：<string name="button1">Button</string>

button1 は識別用の ID

Button は表示用のテキスト

- main.xml の android:text 属性で、上記の ID を使って、表示用のテキストを呼び出す。

例：android:text="@string/button1"

string は strings.xml のことを指している

button1 は ID

これによって Button というテキストがボタンに表示される。

※これについては、p.175「Hint strings.xml ファイルを使ってテキストをコントロールする」で詳しく解説していますので、併せて参照してください。

「5.2.1 イベント処理の仕組み」のサンプルプログラム

▼書き換え前の main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button" /> ←★この部分を以下の例のように書き換えます
```

```
</LinearLayout>
```

▼書き換え後の main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button1" /> ←★このように書き換えます
```

```
</LinearLayout>
```

▼strings.xml の記述

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
```

```
    <string name="hello"></string>
```

```
    <string name="app_name">05_02_01_00</string>
```

```
    <string name="button1">Button</string>    ←★記述します
```

```
</resources>
```

<本書サポートサイト>

<http://www.shuwasystem.co.jp/support/7980html/3372.html>

<秀和システム>

<http://www.shuwasystem.co.jp/>