

7 アクティビティに「表示機能」を追加する



ファイル名 : res/layout/activity_main.xml (追加)

```

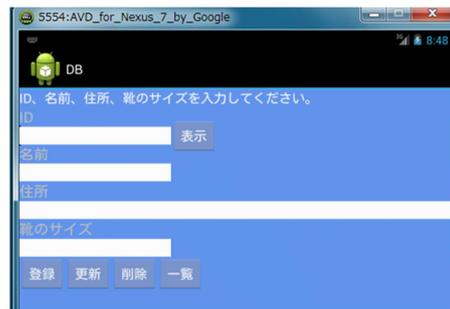
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" //①
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#6495ED"
    tools:context=".MainActivity" >
    <TextView
        android:id="@+id/text_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:text="@string/label_title"
        android:textAppearance="?android:attr/textAppearanceMedium" />
    <TextView
        android:id="@+id/text_id"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/text_title"
        android:text="@string/label_id"
        android:textSize="20sp" />
    <EditText
        android:id="@+id/edit_ID"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/text_id"
        android:background="#FFFFFF"
        android:ems="10" //②
        android:digits="01234567890" //③
        android:inputType="number1" //④
        android:textColor="#000000" >
        <requestFocus /> //⑤
    </EditText>

```

```

<TextView
    android:id="@+id/text_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/edit_ID"
    android:text="@string/label_name"
    android:textSize="20sp" />
<EditText
    android:id="@+id/edit_Name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/text_name"
    android:background="#FFFFFF"
    android:ems="10"
    android:inputType="textPersonName"
    android:textColor="#000000" />
<TextView
    android:id="@+id/text_address"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/edit_Name"
    android:text="@string/label_address"
    android:textSize="20sp" />
<EditText
    android:id="@+id/edit_Address"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/text_address"
    android:background="#FFFFFF"
    android:inputType="textPersonName"
    android:textColor="#000000" />
<TextView
    android:id="@+id/text_size"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/edit_Address"
    android:text="@string/label_size"
    android:textSize="20sp" />
<EditText
    android:id="@+id/edit_Size"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/text_size"
    android:background="#FFFFFF"
    android:ems="10"
    android:inputType="numberDecimal"
    android:digits="0123456789."
    android:textColor="#000000" />

```



```

<Button
    android:id="@+id/button_insert"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/edit_Size"
    android:text="@string/button_insert" />
<Button
    android:id="@+id/button_update"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/edit_Size"
    android:layout_toRightOf="@+id/button_insert"
    android:text="@string/button_update" />
<Button
    android:id="@+id/button_delete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/edit_Size"
    android:layout_toRightOf="@+id/button_update"
    android:text="@string/button_delete" />
<Button
    android:id="@+id/button_list"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/edit_Size"
    android:layout_toRightOf="@+id/button_delete"
    android:text="@string/button_list" />
<Button
    android:id="@+id/button_show"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/edit_ID"
    android:layout_alignBottom="@+id/edit_ID"
    android:layout_toRightOf="@+id/edit_ID"
    android:text="@string/button_show" />
</RelativeLayout>

```

①relative layout

relativeLayout は、名前のおり相対的にウィジェットの配置を行うレイアウトです。以下のパラメータを使って、親ウィジェットに対する相対位置を指定して、ウィジェットの配置を行うことができます。

横軸の位置を指定するパラメーター

- ・左側に配置 : layout_alignParentLeft
- ・中央に配置 : layout_centerHorizontal
- ・右側に配置 : layout_alignParentRight

縦軸の位置を指定するパラメーター

- ・上側に配置 : layout_alignParentTop
- ・中央に配置 : layout_centerVertical
- ・下側に配置 : layout_alignParentBottom

横軸、縦軸ともに中央に配置するパラメーター

・横軸，縦軸ともに中央に配置：layout_centerInParent

これは一つのパラメータで、横軸に centerHorizontal を、縦軸を layout_centerVertical 指定したのと同じ効果を持ちます。

これらのパラメーターには、値として true を指定します。

それ以外に次のパラメーター（属性）があります。

属性	用途
android:layout_above	指定した View の上部とこの View の下部を合わせるように配置します。
android:layout_alignBaseline	指定した View のベースラインに合わせて配置します。
android:layout_alignTop	指定した View の上方向に配置します。
android:layout_alignBottom	指定した View の下方向に配置します。
android:layout_alignLeft	指定した View の左方向に配置します。
android:layout_alignRight	指定した View の右方向に配置します。
android:layout_alignParentTop	親の上方向に配置します。
android:layout_alignParentBottom	親の下方向に配置します。
android:layout_alignParentLeft	親の左方向に配置します。
android:layout_alignParentRight	親の右方向に配置します。
android:layout_alignWithParentIfMissing	親がアンカーとして使われるとき、アンカーは layout_toLeftOf, layout_toRightOf などを受け付けません。
android:layout_below	指定した View の下部とこの View の上部を合わせるように配置します。
android:layout_centerHorizontal	子 View の水平配置を親 View の幅に対して中央となるように配置します。
android:layout_centerInParent	親の水平方向の中央に配置します。
android:layout_centerVertical	子 View の上下配置を親 View の幅に対して中央となるように配置します。
android:layout_toLeftOf	指定した View の左端と View の右端を合わせるように配置します。
android:layout_toRightOf	指定した View の右端と View の左端を合わせるように配置します。

②android:ems="10"

ウィジェットの幅を EM 値で指定します。EM 値とは、font-size プロパティの値を 1 とする大きさです。つまりフォントサイズの大きさを 1 として考え、10 文字分の入力枠を設定します。

③android:digits="0123456789."

EditText で使える属性であり、入力制限や IME の制御をする属性は次のとおりです。

属性	説明
digits ※指、アラビア数字の意	入力可能な数値、記号の文字セットを指定する。指定した文字以外への入力ができなくなる。
inputType	入力する内容に応じたソフトウェアキーボードを表示する。

④android:inputType="numberDecimal"

inputType の入力制限は、次のものが使用できます。制限は「|」で区切って複数指定できます。

種類	説明
none	入力不可

text	普通のテキスト
textCapCharacters	すべて大文字で入力する場合
textCapWords	単語の先頭を大文字で入力する場合
textCapSentences	文章の先頭を大文字で入力する場合
textAutoCorrect	文字の入力を自動で修正する場合
textAutoComplete	文字の補完入力する場合
textMultiLine	文字を複数行入力する場合
textImeMultiLine	通常の文字入力時は複数入力を許可せず、IME によって複数行入力を設定する場合
textUri	URL を入力する場合
textEmailAddress	メールアドレスを入力する場合
textEmailSubject	メールの件名を入力する場合
textShortMessage	ショートメッセージを入力する場合
textLongMessage	ロングメッセージを入力する場合
textPersonName	名前を入力する場合
textPostalAddress	住所を入力する場合
textPassword	パスワードを入力する場合
textVisiblePassword	パスワードの文字を見せて入力する場合
textWebEditText	HTML を入力する場合
textFilter	他のデータでフィルタされた文字を入力
textPhonetic	発音記号を入力する場合
number	数値入力する場合
numberSigned	符号付きの数値を入力する場合
numberDecimal	小数付数値を入力する場合
phone	電話番号を入力する場合
datetime	日付時刻を入力する場合
date	日付を入力する場合
time	時刻を入力する場合

⑤<requestFocus />

xml レイアウトファイルに requestFocus タグを指定する事で、フォーカスを設定することができます。

フォーカス移動の指示(カーソル操作等)があった場合、デフォルトでは指示のあった方向に一番近いウィジェットにフォーカスが移ります。

もし、フォーカス移動の順番を変えたい場合、ウィジェットに次のフォーカス先を指定します。

XML 属性	メソッド	説明
android:nextFocusUp	View#setNextFocusUpId	フォーカスを上に移動しようとしたときのフォーカス先を指定
android:nextFocusDown	View#setNextFocusDownId	フォーカスを下に移動しようとしたときのフォーカス先を指定
android:nextFocusLeft	View#setNextFocusLeftId	フォーカスを左に移動しようとしたときのフォーカス先を指定
android:nextFocusRight	View#setNextFocusRightId	フォーカスを右に移動しようとしたときのフォーカス先を指定

ファイル名： src/jp/edu/mie/ View020.java

(1)View010.java をコピーして、View020.java に名前変更してください。

(2)MainActivity.java の①に View020 と入力してください。

(3) AndroidManifest.xml の①に View020 と入力してください。

```

/*
 * View020
 */

. . .

import android.widget.Button;
import android.widget.EditText;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

public class View020 extends Activity
{
    Button listButton;
    SQLiteDatabase db;
    EditText idText;
    EditText nameText;
    EditText addressText;
    EditText shoeSizeText;
    Button showButton;
    String strSQL;
    String name;
    String address;
    double shoeSize;
    int id;

    protected void onCreate(Bundle savedInstanceState)
    {
        . . .

        listButton.setOnClickListener(new MainOnClickListener());
        MyOpenHelper helper = new MyOpenHelper(this); //①データベースオブジェクトの取得
        db = helper.getWritableDatabase(); //データベースを開く
        idText = (EditText) findViewById(R.id.edit_ID);
        nameText = (EditText) findViewById(R.id.edit_Name);
        addressText = (EditText) findViewById(R.id.edit_Address);
        shoeSizeText = (EditText) findViewById(R.id.edit_Size);
        showButton = (Button) findViewById(R.id.button_show);
        showButton.setOnClickListener(new MainOnClickListener());
    }
    class MainOnClickListener implements OnClickListener
    {
        public void onClick(View view)
        {
            //一覧
            if (view == listButton)
            {
                . . .
            }
            //表示
            else if (view == showButton)
            {
                try{

```

```
//該当のレコードを画面に表示する
id = Integer.parseInt(idText.getText().toString());
strSQL = "SELECT * FROM product WHERE id = '" + id + "'";
Cursor cr = db.rawQuery(strSQL, null);//②
cr.moveToFirst();
int i = cr.getColumnIndex("id");//列名を指定して列番号を取得する
int n = cr.getColumnIndex("name");
int a = cr.getColumnIndex("address");
int s = cr.getColumnIndex("shoeSize");
id = Integer.parseInt(cr.getString(i)); //id の内容を設定する
name = cr.getString(n); //name の内容を設定する
address = cr.getString(a); //address の内容を設定する
String shoeSize = cr.getString(s);
//Log.d("db","update3");
nameText.setText(name);
addressText.setText(address);
shoeSizeText.setText(shoeSize);
} catch (Exception e) {
    e.printStackTrace();
}
}
}
}
```

①MyOpenHelper helper = new MyOpenHelper(this);

SQLiteOpenHelper のサブクラス「MyOpenHelper」を使って、実際にデータベースを開くには、getWritableDatabase() メソッドを使用します。

②Cursor cr = db.rawQuery(strSQL, null);

execSQL メソッドは値を返さないため、データベースのデータを取得することはできません。代わりに、query メソッドか.rawQuery メソッドを使います。query メソッドは、単純な SQL によるデータを取得には便利ですが、複雑な SQL をおこなうには、rawQuery メソッドを使った方が便利です。

rawQuery メソッドは、引数に SQL 文と、SQL 文にパラメータが使われている場合は、パラメータ値を文字列配列で指定します。(パラメータが使われていない場合は null)

query メソッドと rawQuery メソッドは、どちらも Cursor オブジェクトを返します。カーソルを操作する事で、1 行ずつデータを取得します。