

BMI計算アプリ

身長と体重をユーザが入力し、その値を計算して、BMI 値を表示するアプリケーションを作る。

1. プロジェクトを作る

新規 Android アプリケーション・プロジェクトを作る。HelloWorld アプリケーションをつくる時と同じで良いが、アプリケーション名、プロジェクト名、パッケージ名は以下のように設定する。



図 1 新規アプリケーションの設定をする

2. レイアウトを設定する

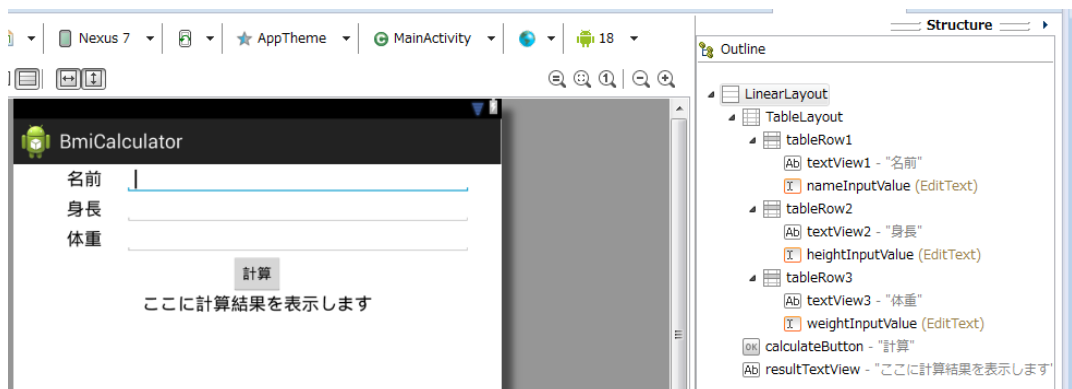


図 2 このようなレイアウトを目指す

(1) レイアウトを削除する

プロジェクトフォルダの中にある、「res/layout/activity_main.xml」ファイルをダブルクリックで開く。

編集画面の下の「グラフィカル・レイアウト」タブの隣にある「activity_main.xml」をクリックする。このタブを選ぶことで、グラフィカルに編集する画面と、エディタで XML

ファイルを直接編集する画面を切り替えることができる。

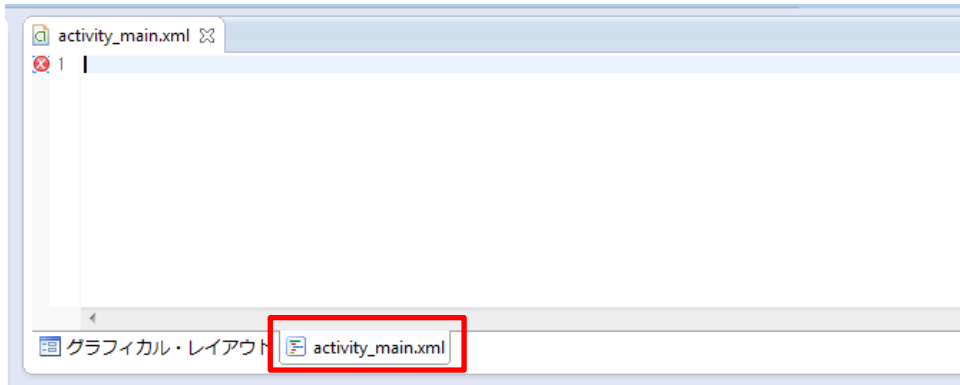


図 3 標準のレイアウトを削除する

(2) グラフィカル・レイアウト画面で「LinearLayout(Vertical)」を追加する

パレットウィンドウの「レイアウト」の中にある「LinearLayout(Vertical)」をドラッグして、編集画面にドロップする。

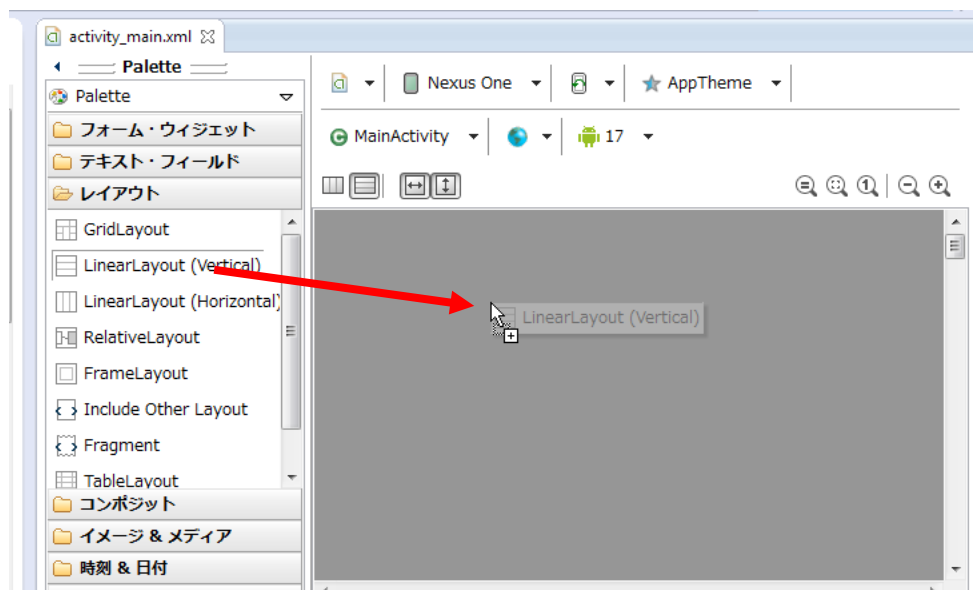


図 4 LinearLayout(Vertical)の追加

(3) 「LinearLayout(Vertical)」に「TableLayout」を追加する

レイアウトの構造を明確にして追加したいときには、Structure ウィンドウの Outline に表示されているコンポーネントにドラッグ&ドロップするとよい。

今回は、Palette/レイアウトにある「TableLayout」をドラッグして、「LinearLayout(Vertical)」にドロップする。テーブルレイアウトに、TableRow が追加されている様子が確認できる。

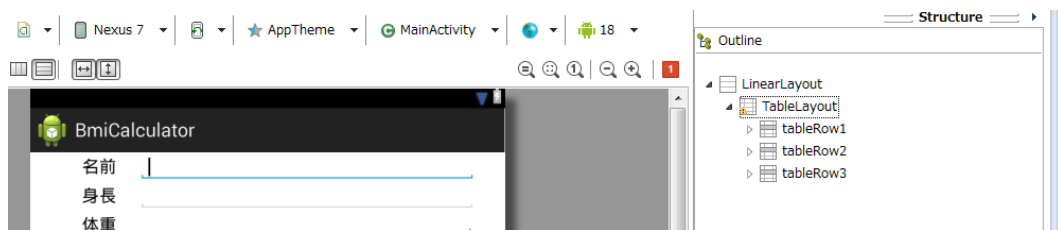


図 5 TableLayout と tableView が追加されている様子

(4) 項目名, 入力エリア, ボタン, 結果表示場所を追加する

(a) 項目名と入力エリア

第 1 に, 名前を入力するための行として, 「tableView」に, TextView (Medium Text) と EditText(Person Name)を追加する.

第 2 に, 身長を入力するための行として, 「tableView」に, TextView (Medium Text) と EditText(番号)を追加する.

第 3 に, 体重を入力するための行として, 「tableView」に, TextView (Medium Text) と EditText(番号)を追加する.

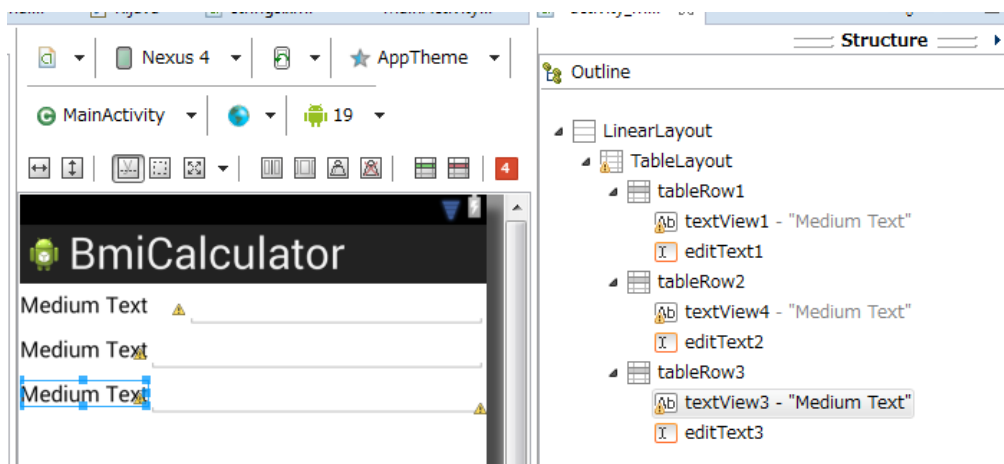


図 6 ビューを追加する

(b) ボタンと結果表示場所

ボタンを「LinearLayout(Horizontal)」の第 4 番目の項目として追加する.

TextView (Large Text) を第 5 番目の項目として追加する.

ボタンと結果表示場所は, 中央にあった方が適切ななので, 真ん中に移す.

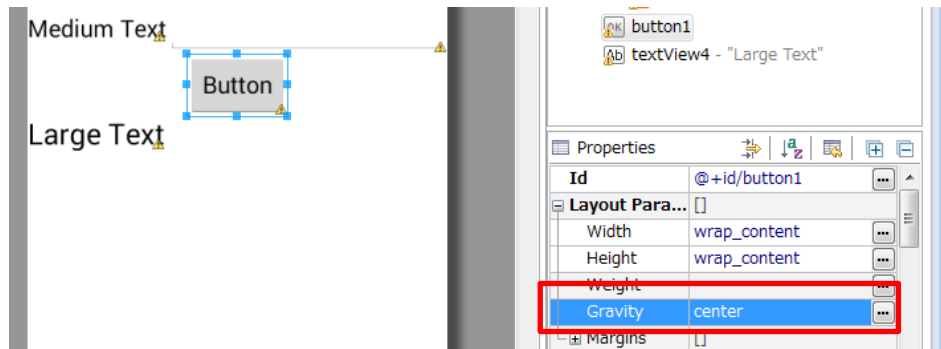


図 7 ボタンを中央に移動する

同様に、結果表示のための TextView も中央に表示されるようにする。

(c) レイアウトを調整する

Layout Parameter にある Width, Height, Gravity を調整して、レイアウトを整える。

Width, Height では、wrap_content, match_parent などから選ぶ。wrap_content では、表示する内容がちょうどはまるように合わせて調整される。match_parent では、そのビューが格納されているレイアウトいっぱいに表示される。

Gravity では、表示位置を指定する。top, bottom, center_vertical, right, left, center などから選ぶ。

ここでは、TableLayout について表 1 のように、設定することで、どのような横幅の端末であっても、中央に表示されるようになる。

表 1 TableLayout の設定

項目	値
Width	wrap_content
Height	wrap_content
Gravity	center

(5) 表示する文字列を指定する

入力の項目名、ボタンの名前、結果表示場所に表示する文字列を strings.xml に定義して、レイアウトにそれらの文字列を表示するように指定する。

(a) strings.xmlに文字列を追加する

res/values/strings.xml を開き、以下の名前文字列を定義する。

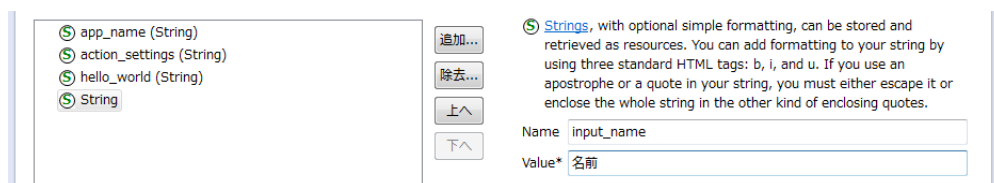


図 8 文字列リソースの入力

表 2 strings.xml に設定する項目

Name	Value
input_name	名前
input_height	身長
input_weight	体重
button_name	名前
result_default	ここに結果を表示します

(b) レイアウトで定義を参照する

activity_main.xml を開き、textView をクリックし、プロパティの Text 項目にある右の「・・・」ボタンを選ぶ。

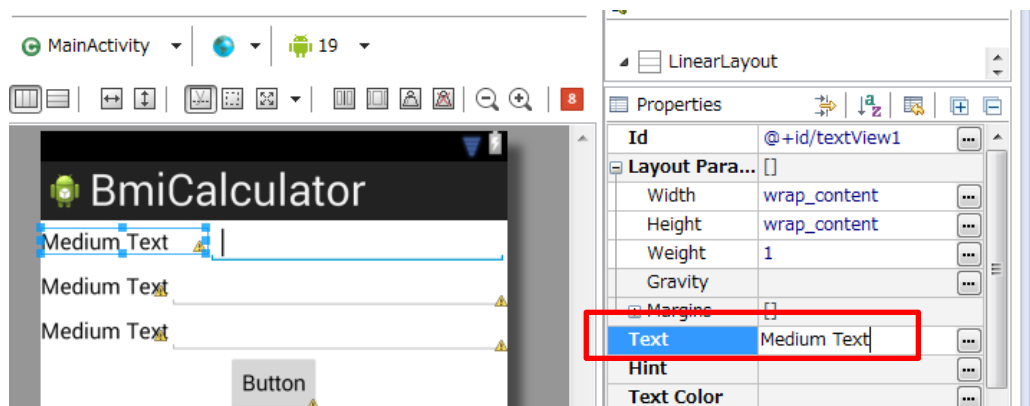


図 9 文字列を変更するビューを選び、プロパティを変更する

先に strings.xml で定義した文字列の名前があるので、その場所に表示したい文字列を選ぶ。

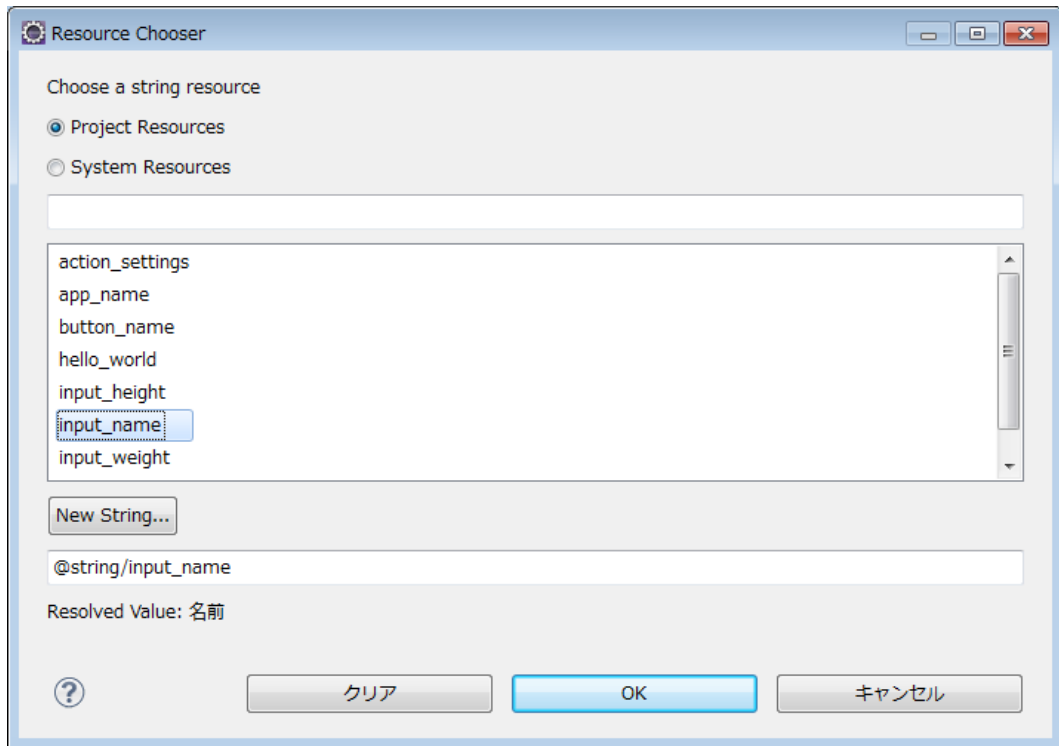


図 10 定義した文字列リソースを選ぶ

(6) プログラムで参照するコンポーネントのリソース名を変更する

プログラムから、入力したデータを取り出したり、結果を画面に表示させたりするために使うビューに、参照しやすい分かりやすいリソース名をつける。

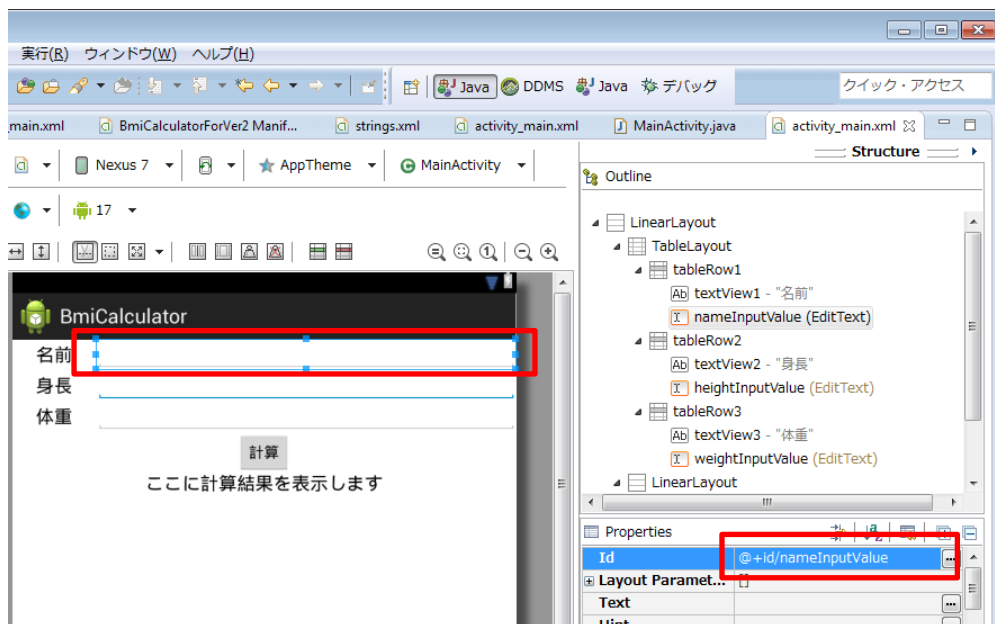


図 11 リソース名を変更するための画面選択

- (a) IDをつける対象のViewをクリックする。
- (b) PropertiesのId項目の右の「・・・」を選び、値として、「nameInputValue」をつける。
注意として、Properties画面にて直接編集するときには「@+id/nameInputValue」という表示になる。
- (c) 同様に、身長入力のEditText、体重入力のEditText、計算Button、結果表示TextViewにもIdをつける。Idは以下のような値とする。

表 3 リソース名を変更する一覧

項目	Id
名前入力 EditText	nameInputValue
身長入力 EditText	heightInputValue
体重入力 EditText	weightInputValue
計算 Button	calculateButton
結果表示 TextView	resultTextView

新しい ID を入力した後に、のようなダイアログが出るが、「はい」として、同じ ID が振られた箇所をすべて変更する。

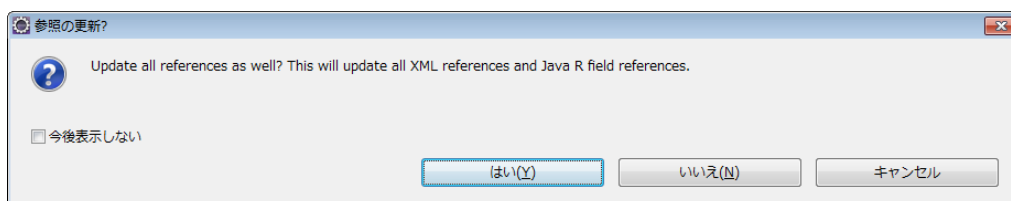


図 12 関連する変数を併せて変更するか問いあわせるダイアログ

変更後には、図 13 のように定義させるので、XML ファイル、Java ファイル、それぞれから参照する仕方を確認する。

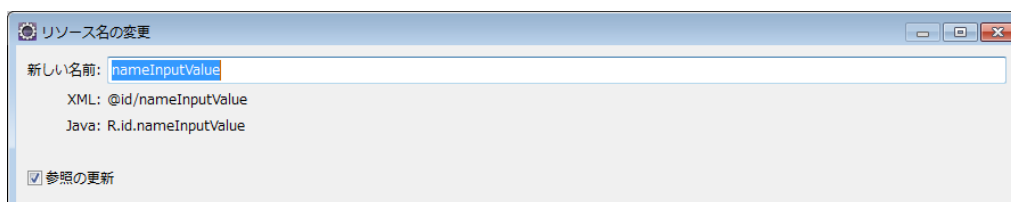


図 13 リソース名を変更するときの確認

3. イベント処理

(1) 処理内容を記述する

- (a) 前章で設定したイベント処理の方法を使う時には、以下のようにハンドラの内容を変更する。

リスト 1 イベント処理の記述 (レイアウトファイルで指定したとき)

1	public void onClickHandler(View view)
2	{

```

3   String resultString = "";
4   //
5   TextView heightInputValue =
   (TextView)findViewById(R.id.heightInputValue);
6   Float heightValue =
       Float.parseFloat(heightInputValue.getText().toString());
7
8   TextView weightInputValue =
       (TextView)findViewById(R.id.weightInputValue);
9   Float weightValue =
       Float.parseFloat(weightInputValue.getText().toString());
10
11  TextView nameInputTextView = (TextView)findViewById(R.id.nameInputValue);
12  String nameString = nameInputTextView.getText().toString();
13
14  Float BMI = (float) (weightValue/(heightValue/100.0*heightValue/100.0));
15
16  TextView resultView = (TextView)findViewById(R.id.resultTextView);
17
18  if(BMI > 25)
19  {
20      resultString = "肥満";
21  }
22  else if(BMI > 18.5)
23  {
24      resultString = "普通体重";
25  }
26  Else
27  {
28      resultString = "低体重";
29  }
30  resultView.setText(nameString+"さんの BMI は"
        +String.format("%.1f", BMI)+"で" + resultString + "です");
31  }

```

(b) ボタンにリスナーを追加する方法

なお、このときには、activity_main.xml の on Click に値を設定する必要はない。

リスト 2 イベント処理の記述 (ソースコードで指定するとき)

```

1   package nnct.fujita.bmicalculator;
2
3   import android.app.Activity;
4   import android.os.Bundle;
5   import android.view.Menu;
6   import android.view.View;
7   import android.widget.Button;
8   import android.widget.TextView;
9
10  public class MainActivity extends Activity
11  {
12      @Override
13      protected void onCreate(Bundle savedInstanceState)
14      {
15          super.onCreate(savedInstanceState);
16          setContentView(R.layout.activity_main);
17
18          Button bt = (Button)findViewById(R.id.calculateButton);
19          bt.setOnClickListener(new View.OnClickListener()
20          {
21              @Override
22              public void onClick(View view)

```



```

23     {
24         String resultString = "";
25
26         TextView heightInputValue
27             = (TextView)findViewById(R.id.heightInputValue);
28         Float heightValue
29             = Float.parseFloat(heightInputValue.getText().toString());
30
31         TextView weightInputValue
32             = (TextView)findViewById(R.id.weightInputValue);
33         Float weightValue
34             = Float.parseFloat(weightInputValue.getText().toString());
35
36         TextView nameInputTextView
37             = (TextView)findViewById(R.id.nameInputValue);
38         String nameString = nameInputTextView.getText().toString();
39
40         Float BMI
41             = (float) (weightValue/(heightValue/100.0*heightValue/100.0));
42
43         TextView resultView = (TextView)findViewById(R.id.resultTextView);
44
45         if(BMI > 25)
46         {
47             resultString = "肥満";
48         }
49         else if(BMI > 18.5)
50         {
51             resultString = "普通体重";
52         }
53         else
54         {
55             resultString = "低体重";
56         }
57         resultView.setText(nameString
58             + "さんの BMI は "
59             + String.format("%.1f", BMI) + " で "
60             + resultString + " です");
61     }
62 });
63 }
64
65 @Override
66 public boolean onCreateOptionsMenu(Menu menu)
67 {
68     // Inflate the menu; this adds items to the action bar if it is present.
69     getMenuInflater().inflate(R.menu.main, menu);
70     return true;
71 }

```

4. 実行する



図 14 BmiCalculator を実行した様子