周期的処理

一定時間間隔で，処理したり，表示を更新したりする場合には，周期的な処理を設定する必要がある．周期的な処理をするための方法は多々あり，それぞれの目的によって異なる．

一例として，Handler クラスにおけるメッセージのやり取りを利用して，一定時間間隔で処理を行う方法を示す．

(1) MainActivityクラス

onResume メソッドでは，周期的処理を扱うためのハンドラを用意し，処理を開始している．

onPause メソッドでは，フォーカスがこの activity から外れたときに，周期的処理をやめるために，ハンドラのインスタンスを null としている．この処理を実行しないと，実行がバックグラウンド状態になったときでも，周期的処理を続けることになる．

さらに，周期的に処理する内容を含める PeriodicMethod メソッドを定義している．この名称は自分でつけてよいが，PeriodicHandler クラスでこのメソッドを周期的に実行しているので，その実行する命令と合わせる必要がある．

これらの処理を MainActivity に実装すると，リスト 1 のようになる．

リスト 1 周期的処理を利用する MainActivity クラス

```
 1  public class MainActivity extends Activity
 2  {
 3      PeriodicHandler periodicHandler;
 4      private int count = 0;
 5
 6      @Override
 7      protected void onCreate(Bundle savedInstanceState)
 8      {
 9          super.onCreate(savedInstanceState);
10          setContentView(R.layout.activity_main);
11      }
12
13      protected void onResume()
14      {
15          super.onResume();
16          //周期的に処理するためのハンドラをインスタンス化する
17          periodicHandler = new PeriodicHandler(this);
18          //処理を開始するために，遅延なしで開始する
19          periodicHandler.sleep(0);
20      }
21
22      protected void onPause()
23      {
24          super.onPause();
25          //activity が正面に表示されなくなったら，周期的な処理をしない
26          periodicHandler = null;
27      }
28
29      public boolean onCreateOptionsMenu(Menu menu)
30      {
```

```
31          getMenuInflater().inflate(R.menu.main, menu);
32          return true;
33      }
34
35      //周期的に実行させるメソッド
36      public void PeriodicMethod()
37      {
38          count++;
39          TextView textView = (TextView)findViewById(R.id.textView1);
40          textView.setText(count + "回");
41      }
42  }
```

(2) PeriodicHandlerクラス

メッセージを受信すると処理するというしくみを利用して，そのメッセージを送るタイミングを遅らせることで，一定の間隔で周期的な実行を実現している．

handleMessage メソッドには，メッセージを受信したときの処理を定義する．送られるメッセージはメッセージキューに入っているので，処理するメッセージを削除してから，メッセージが来たタイミングで実行したい処理を定義する．ここでは，メッセージが来た時に，周期的に実行したいメソッドを実行している．さらに，次のメッセージを待ち時間後 sleep メソッドに実行する．

sleep メソッドでは，sendMessageDelayed をつかって，一定時間の待ったあとに，メッセージを送る．これによって，一定時間の間隔をあけて，繰り返し handleMessage メソッドが実行されることになる．

このような処理を実装した PeriodicHandler クラスをリスト 2 に示す．

リスト 2　周期的に実行するための PeriodicHandler

```
 1  public class PeriodicHandler extends Handler
 2  {
 3      MainActivity activity;
 4
 5      PeriodicHandler(MainActivity activity)
 6      {
 7          //MainActivity を持つ
 8          this.activity = activity;
 9      }
10      @Override
11      public void handleMessage(Message msg)
12      {
13          //メッセージを削除する
14          removeMessages(0,msg);
15          //周期的にメソッドを実行する
16          activity.PeriodicMethod();
17          //ハンドラが有効であれば，指定した時間ごとに繰り返す
18          if (activity.periodicHandler != null)
19          {
20              //待ち時間を指定する
21              sleep(100);
22          }
23      }
24
25      //指定されたミリ秒数を待ってからメッセージをキューする
26      public void sleep(long delayMilliSec)
27      {
```

```
28          //指定されたミリ秒数待ってからメッセージをキューする
29          sendMessageDelayed(obtainMessage(), delayMilliSec);
30      }
31  }
```

(3) 実行

実行すると，カウントが 1 から始まり，0.1 秒ごとにカウントアップし，図 1 を経て図 2 のような表示になる．

図 1　実行してすぐの表示

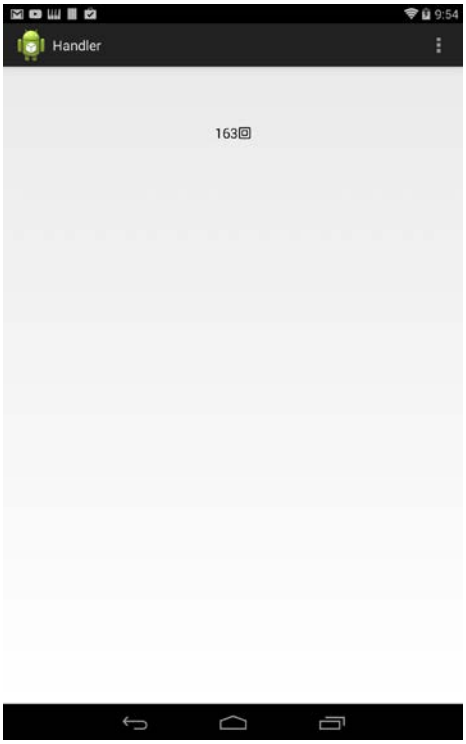

図 2　しばらく経ったときの表示