2015 年 10 月 22 日（木）実施

サービス

サービスとは

Android アプリの動作環境は，スマートフォンの様に電話が掛かってきた場合の対応等も考慮しなければならないが，アクティビティはこの様な場合には，処理を中断してしまう。こういう場合にもバックグラウンドで動作し続けるのが**サービス**である。

サービスはインテントで起動することも可能であるが，**バインド**という仕組みを用いてサービスが提供するインターフェイスとアクティビティとを結び付けて起動しないと，アクティビティ側からサービスへの働き掛けが出来ない。

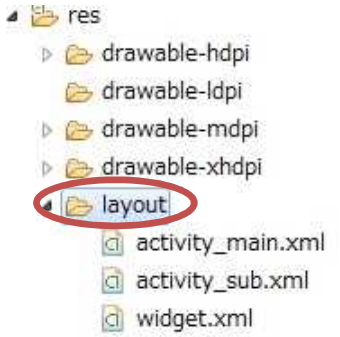
ここでは，AIDL（Android Interface Definition Language； Android インターフェイス定義言語）ファイルから ADT によってインターフェイスを自動生成し，その機能を別の新規クラスに実装し，アクティビティにサービスへのバインドを記述するという手法を学ぶ。

課 題

今回は，前回のプロジェクトに，新規のレイアウトを追加して，テキストビュー，エディットテキスト及びボタンを配置し，最初の画面で背景の画面をクリックすると，そのレイアウト画面に切り替わるアプリを作成することにより，サービスへのバインド処理の基本を学ぶ。

Android アプリの作成

Eclipse を起動し，パッケージ・エクスプローラーの『First』→『res』を展開して『layout』を選択し，『ファイル』→『新規』→『その他』と選択する。



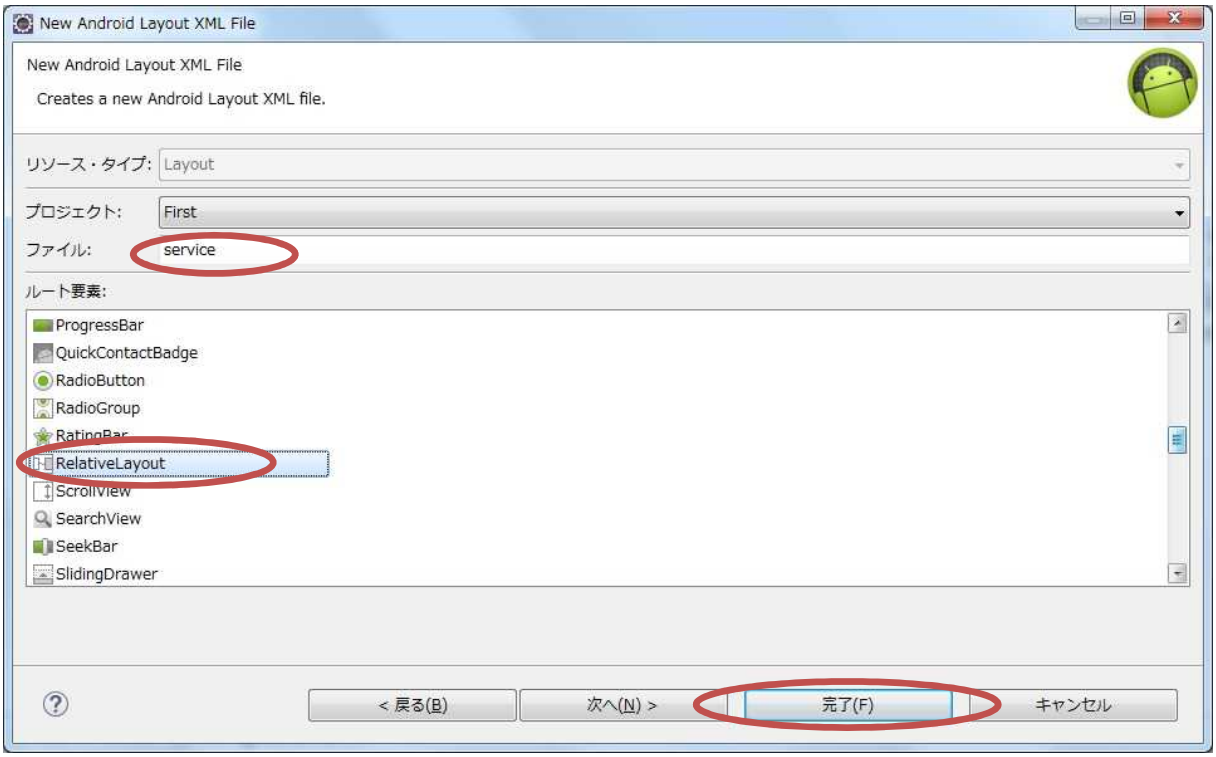
『Android』を展開し，『Android XML レイアウト・ファイル』を選択する。

『次へ』をクリックする。

『ファイル』の欄には「service」と入力する。

『RelativeLayout』を選択して，『完了』をクリックする。

（図は次のページ）



『res』→『values』と展開し，『strings. xml』を開き，『追加』をクリックする。『String』を選択して OK をクリックし，『名前』を「height_label」，『Value』を「高さ(センチメートル)」と入力する。同様に，『名前』:「width_label」，『Value』:「幅(センチメートル)」，『名前』:「area_label」，『Value』:「長方形の面積(平方センチメートル)」，『名前』:「init_height」，『Value』:「10」，『名前』:「init_width」，『Value』:「20」の 4 つの String を追加する。

名前	height_label
Value*	高さ(センチメートル)

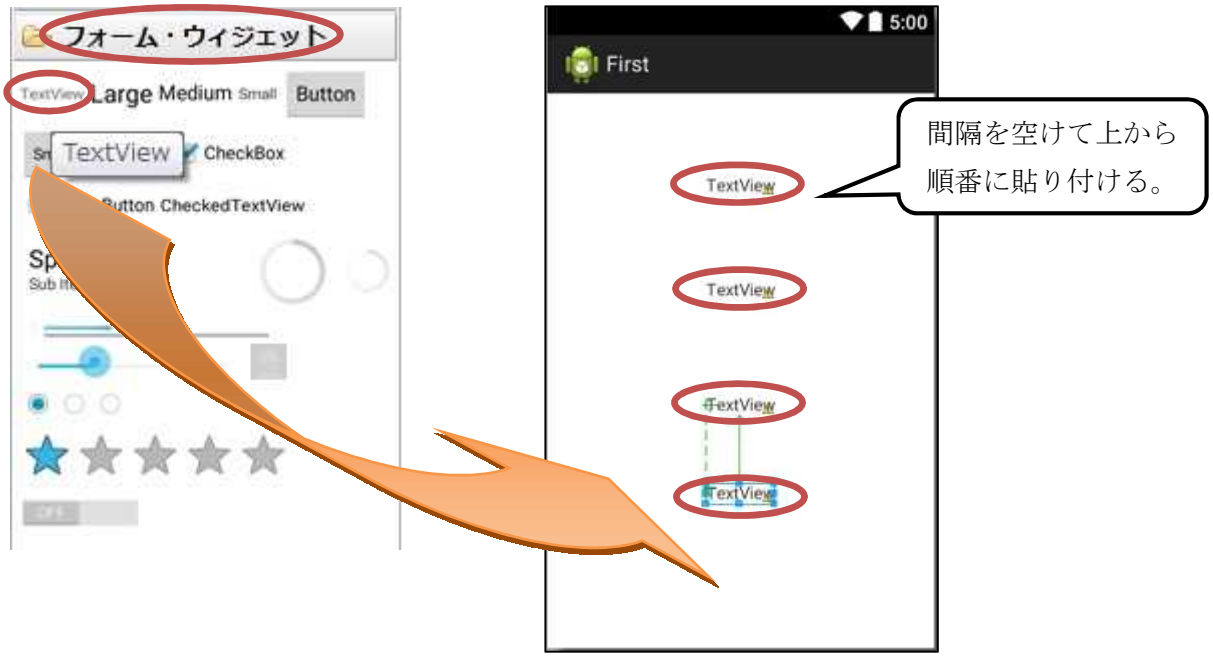
名前	width_label
Value*	幅(センチメートル)

名前	area_label
Value*	長方形の面積(平方センチメートル)

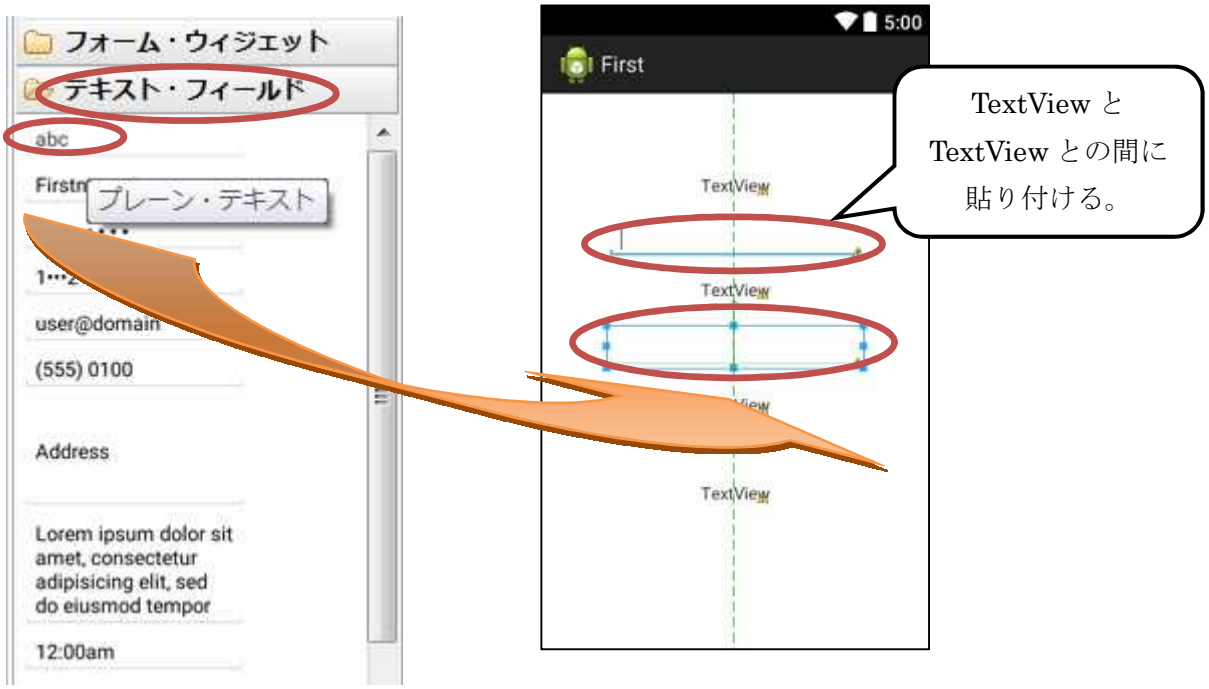
名前	init_height
Value*	10

名前	init_width
Value*	20

『フォーム・ウィジェット』から『TextView』を 4 個，ドラッグして貼り付ける。

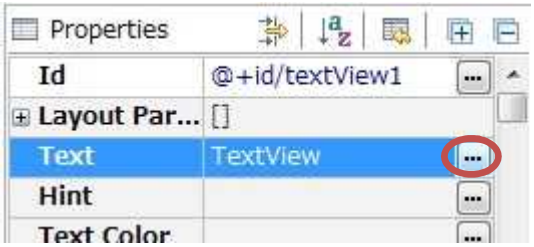


『テキスト・フィールド』から『プレーン・テキスト』を 2 個，ドラッグして貼り付ける。

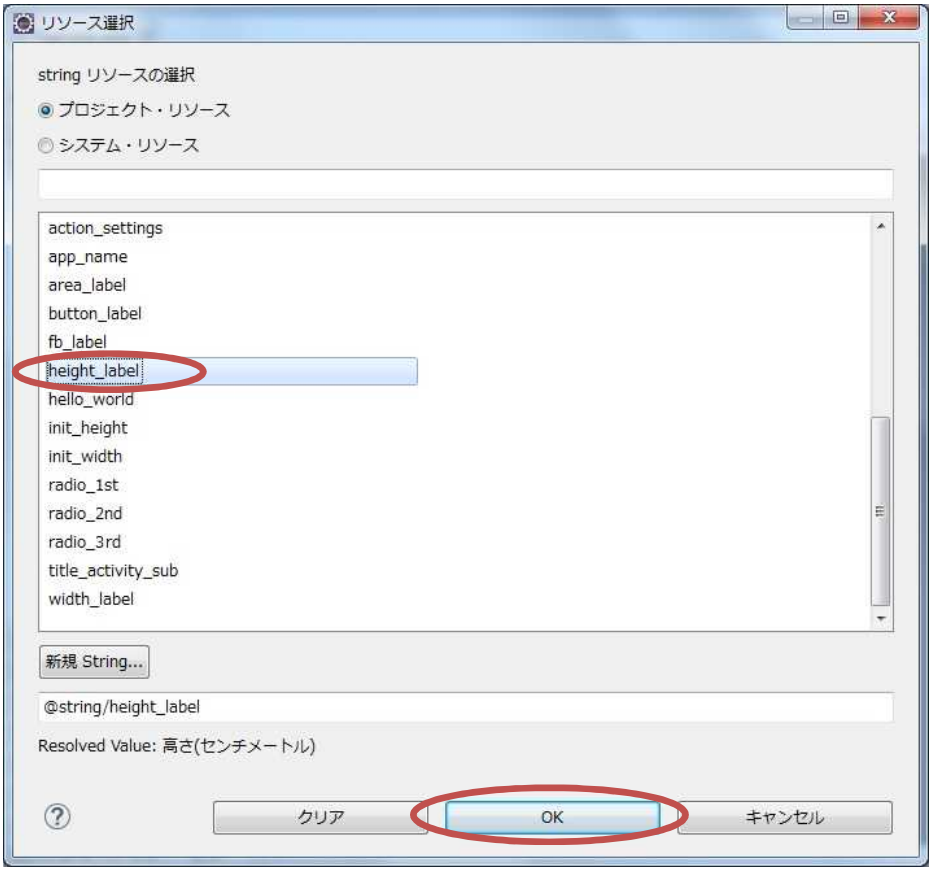


『アウトライン』で textView1～4 及び editText1・2 が貼り付けられたことを確認する。

一番上の『TextView』(textView1) を選択し，『Properties』にある『Text』の右側の『・・・』ボタンをクリックする。

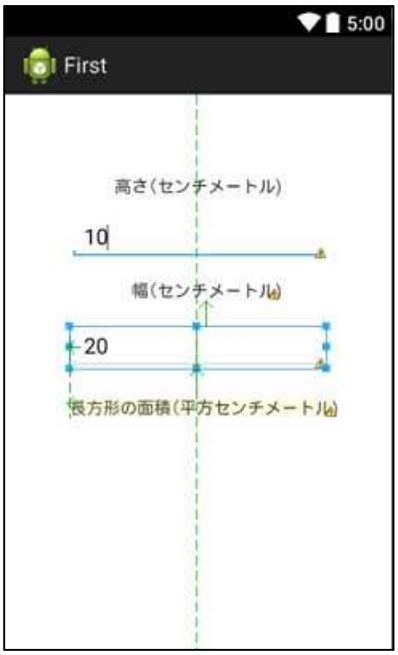


『height_label』を選択し，OK をクリックする。

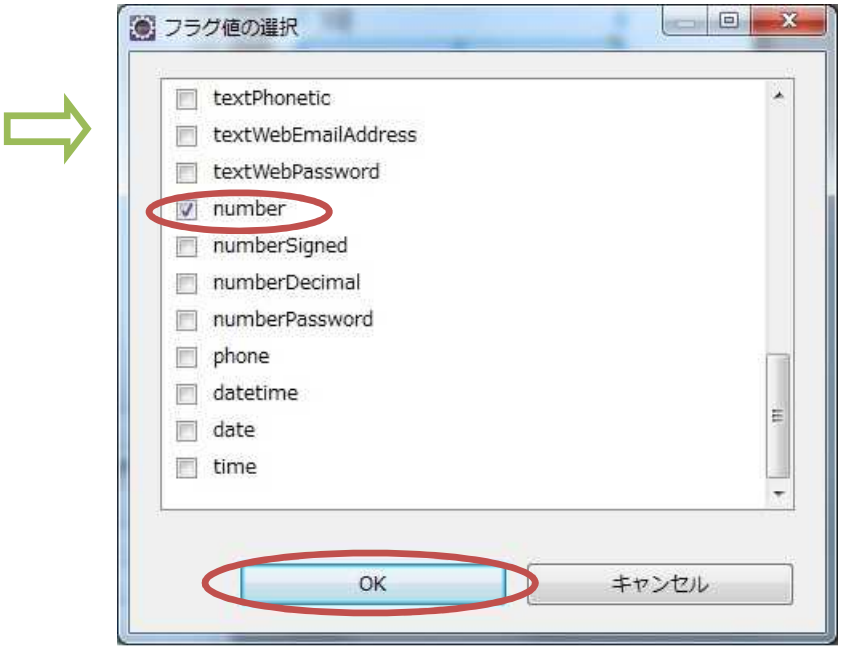


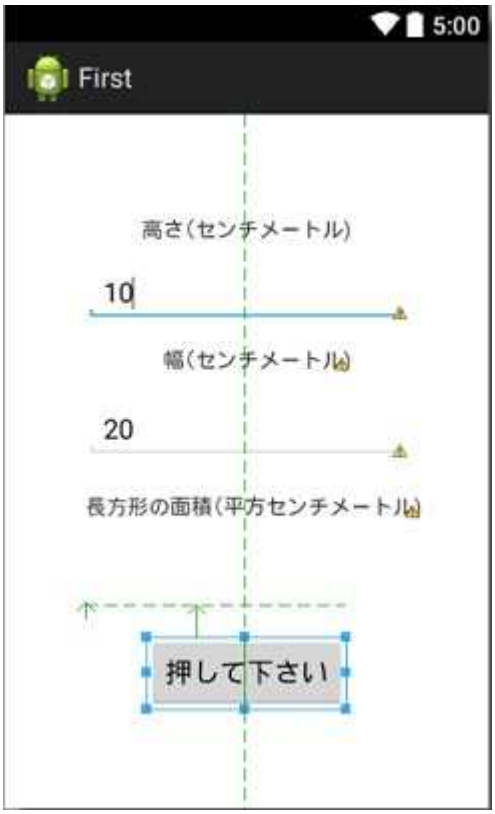
他のウィジェットの『Text』も次の様に設定する。

・二番目の『TextView』(textView2)：「width_label」
・三番目の『TextView』(textView3)：「area_label」
・四番目の『TextView』(textView4)：文字列を削除
・一番目の『EditText』(editText1)：「init_height」　← 高さの初期値
・二番目の『EditText』(editText2)：「init_width」　← 幅の初期値



editText1 及び editTex2 では更に,『Input Type』の右側の『･･･』ボタンをクリックして『number』を設定する。

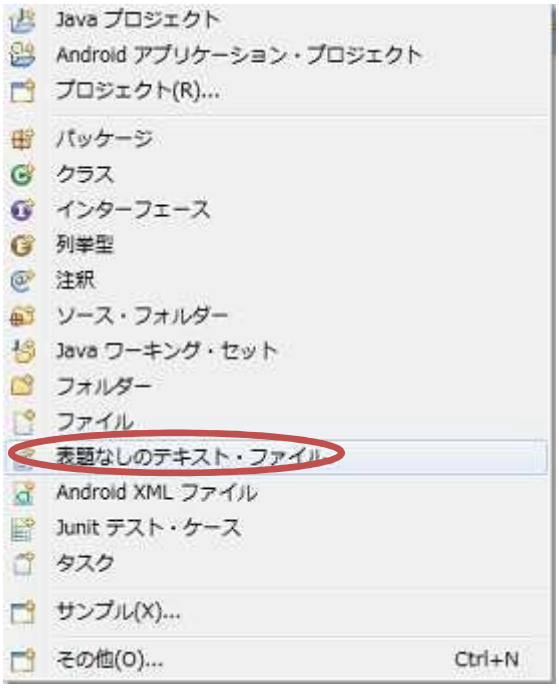




更に，ボタンを配置して，『Properties』の『Text』を『button_label』に設定する。

『保管』のアイコンをクリックして，service. xml を上書き保存する。

『ファイル』→『新規』→『表題なしのテキスト・ファイル』と選択する。



パッケージ名中の jimbo は
自分の名前に読み替える。

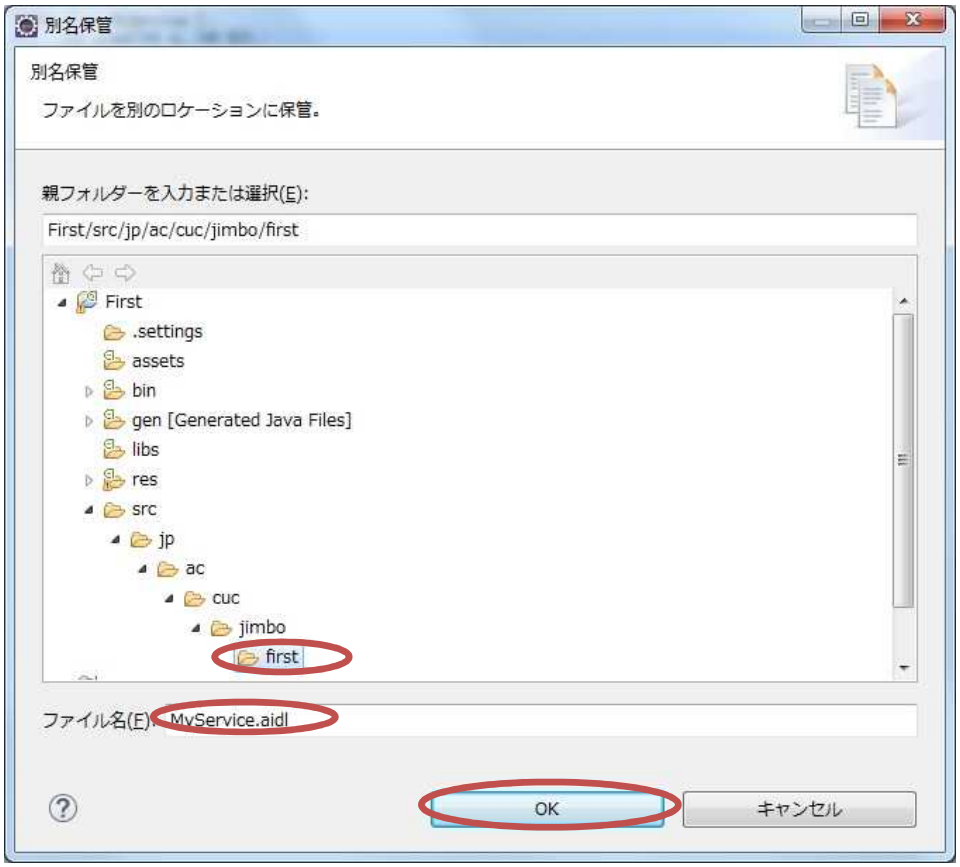
*表題なし 1

```
1 package jp.ac.cuc.jimbo.first;
2
3 interface MyService {
4     int area(int a, int b);
5 }
```

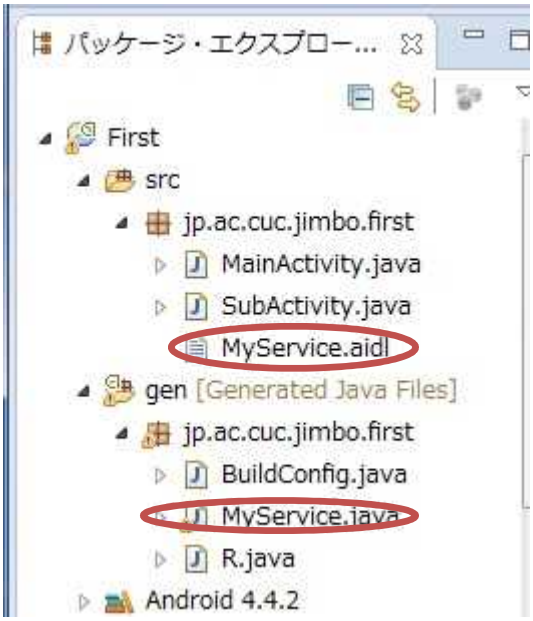
```
package jp.ac.cuc.jimbo.first;

interface MyService {
    int area(int a, int b);
}
```

『ファイル』の『別名保管』を選択する。

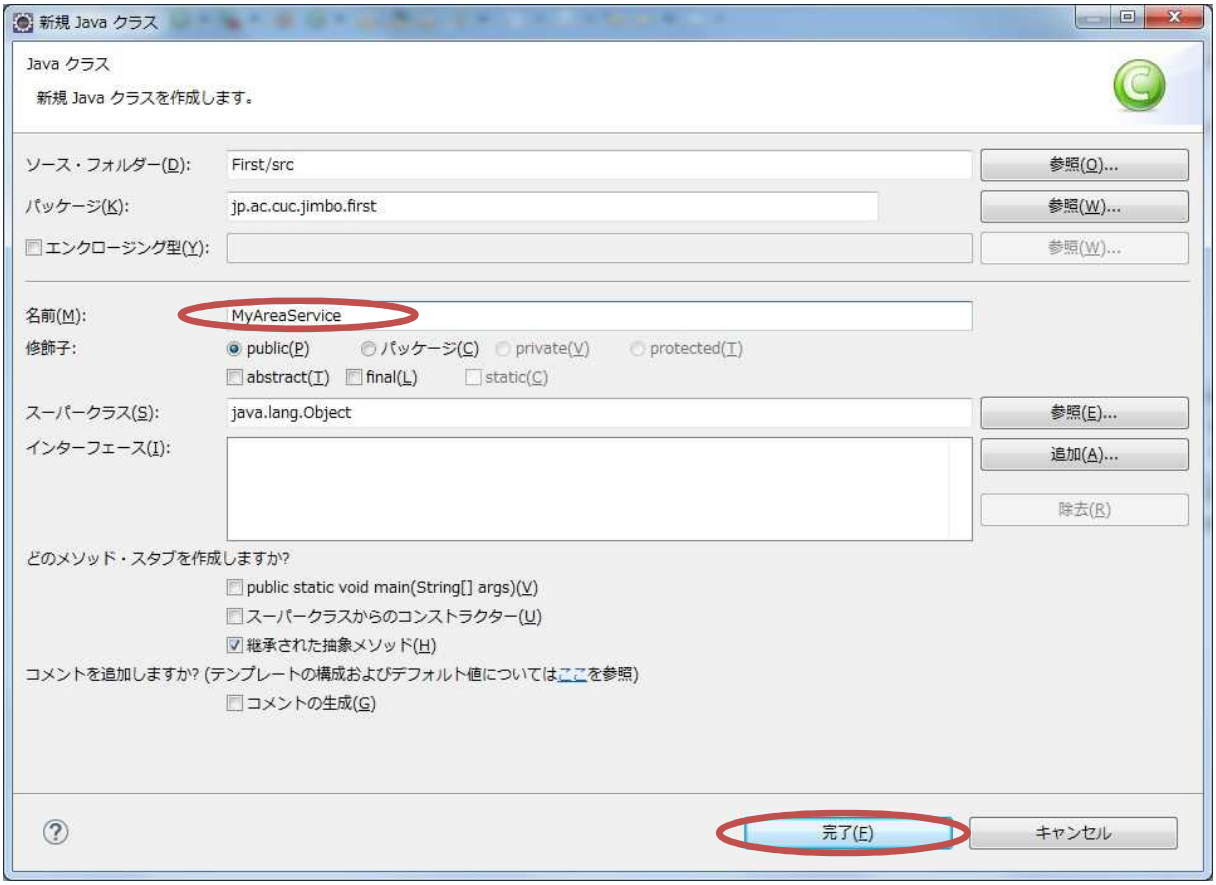
ファイルを保管する場所はアクティビティが保管されているフォルダとし，ファイル名として，「MyService. aidl」と入力する。（拡張子は AIDL の小文字）



パッケージ・エクスプローラーで『src』のパッケージの中に『MyService. aidl』が置かれ，『gen』のパッケージの中に『MyService. java』が自動生成されたことを確認する。



パッケージ・エクスプローラーで『First』→『src』→『jp. ac. cuc. jimbo. first』と選択し，『ファイル』→『新規』→『クラス』を選択する。『名前』を「MyAreaService」として，自動生成されたインターフェイスの機能を，このクラスに実装する。



MyAreaService クラスは Service クラスを継承するので，「extends Service」をクラス名の後に付け加える。自動生成された MyService.java には『Stub』という名前の内部抽象クラスが作成されているので，サービス提供者が実装する必要がある。ここでは，area メソッドとして２つの整数引数の掛け算を返す実装を行い，それを Stub 内のメソッドとしている。

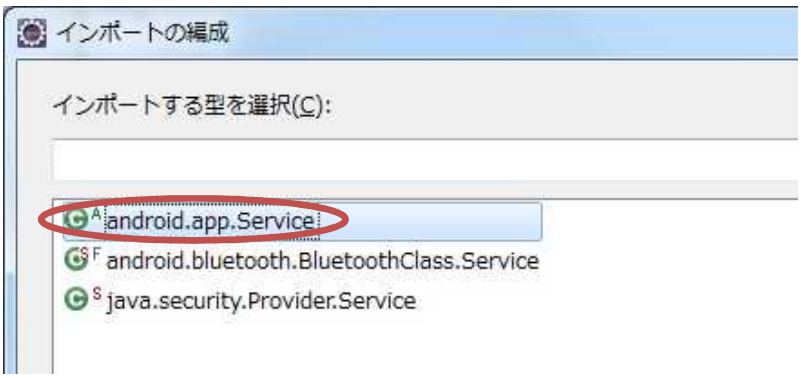
IBinder は Android の IPC（Inter-Process Communication；プロセス間通信）を担うインターフェイスである。

```
package jp.ac.cuc.jimbo.first;

import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.os.RemoteException;

public class MyAreaService extends Service {

    private final MyService.Stub mstub = new MyService.Stub() {
        @Override
        public int area(int a, int b) throws RemoteException {
            return a*b;
        }
    };

    @Override
    public IBinder onBind(Intent intent) {
        return mstub;
    }

}
```

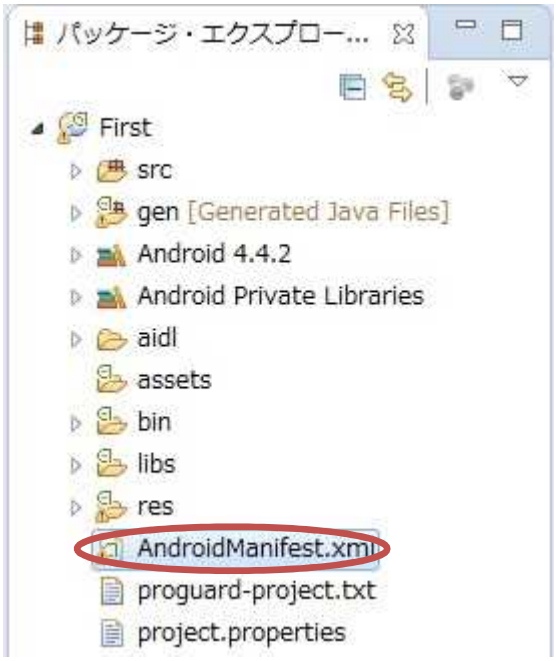
MyAreaService.java

```
1  package jp.ac.cuc.jimbo.first;
2
3  import android.app.Service;
4  import android.content.Intent;
5  import android.os.IBinder;
6  import android.os.RemoteException;
7
8  public class MyAreaService extends Service {
9      private final MyService.Stub mstub = new MyService.Stub() {
10         @Override
11         public int area(int a, int b) throws RemoteException {
12             return a*b;
13         }
14     };
15
16     @Override
17     public IBinder onBind(Intent intent) {
18         return mstub;
19     }
20 }
21
```

＊『インポートの編成』を利用する場合には，Service クラスとして『android. app. Service』を選択する。



パッケージ・エクスプローラーで『First』の直下にある『AndroidManifest. xml』を開く。



次の枠内の内容を AndroidManifest. xml に記述する。

パッケージ名中の jimbo は自分の名前に読み替えて書き加える。

```
<service android:name=".MyAreaService"
         android:process=":remote">
    <intent-filter>
        <action android:name="jp.ac.cuc.jimbo.first.MyService" />
    </intent-filter>
</service>
```

パッケージ・エクスプローラーで『First』→『src』→『jp. ac. cuc. jimbo. first』と展開して，『MainActivity. java』を開く。

以下に，『MainActivity. java』の全体を掲載し，今回書き加えた箇所を赤枠で囲む。（作業量が多いので，『MainActivity_5th_java.txt』という名前のテキストファイルの中に内容を保存して，教材集ページに掲載している。これを利用して構わない。）

onCreate メソッド中に記述した「bindService(new Intent(MyService.class.getName()), msconn, BIND_AUTO_CREATE);」がアクティビティからサービスへのバインドである。

ServiceConnection の onServiceConnected メソッドは，アクティビティのサービスへの接続時に，onServiceDisconnected メソッドはサービスからの切断時に呼び出される。前者で MyService.Stub.asInterface を呼び出して，MyService インターフェイスの参照を取得して，サービス内のメソッドを利用可能にしている。onDestroy メソッドはサービスの終了前に呼び出される。

また，今回は画面に触れたかどうかを判定する onTouchEvent メソッドを書き加え，そこから drawService メソッドを呼び出している。drawService メソッドには，service.xml に対応する画面上の動作を記述している。なお，area メソッドは，android.os.RemoteException に例外を投げているので，利用の際には try 文で囲む必要がある。

```
package jp.ac.cuc.jimbo.first;

import android.app.Activity;
import android.content.ComponentName;
import android.content.Intent;
import android.content.ServiceConnection;
import android.os.Bundle;
import android.os.IBinder;
import android.os.RemoteException;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.RadioGroup.OnCheckedChangeListener;
import android.widget.RatingBar;
import android.widget.RatingBar.OnRatingBarChangeListener;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends ActionBarActivity {
    private MyService myservice;
    private String res;

    private ServiceConnection msconn = new ServiceConnection(){
        @Override
        public void onServiceConnected(ComponentName componentName, IBinder binder) {
            myservice = MyService.Stub.asInterface(binder);
        }

        @Override
        public void onServiceDisconnected(ComponentName name) {
            myservice = null;
        }
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        bindService(new Intent(MyService.class.getName()), msconn, BIND_AUTO_CREATE);

        Button btn = (Button)this.findViewById(R.id.button1);
        btn.setOnClickListener(
            new View.OnClickListener() {

                @Override
                public void onClick(View v) {
                    Intent intent = new Intent();
                    intent.setClassName("jp.ac.cuc.jimbo.prog2ex2",
                                        "jp.ac.cuc.jimbo.prog2ex2.SubActivity");
                    intent.putExtra("message", "MainActivity から移動しました。");
                    startActivity(intent);
```

```
                }
            }
        );

        btn.setOnLongClickListener(
            new View.OnLongClickListener() {

                @Override
                public boolean onLongClick(View v) {
                    drawWidget();
                    return false;
                }
            }
        );
    }
```

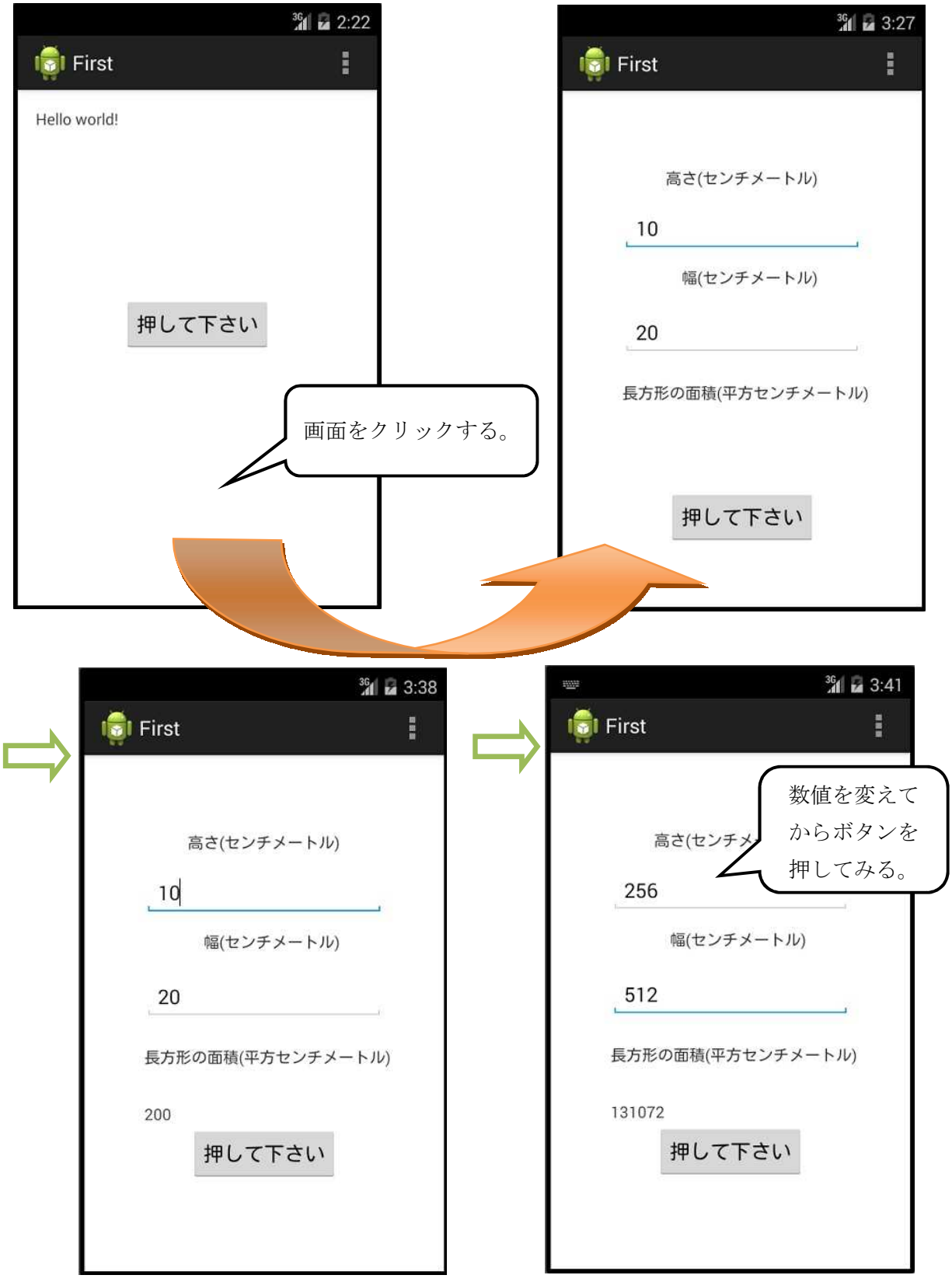
```
    @Override
    protected void onDestroy() {
        super.onDestroy();
        if (myservice != null) {
            unbindService(msconn);
        }
    }
```

```
    public boolean onTouchEvent(MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            drawService();
        }
        return false;
    }
```

```
    public void drawService() {
        setContentView(R.layout.service);

        Button btn = (Button)this.findViewById(R.id.button1);
        btn.setOnClickListener(
            new View.OnClickListener() {

                @Override
                public void onClick(View v)  {
                    EditText et1 = (EditText)findViewById(R.id.editText1);
                    EditText et2 = (EditText)findViewById(R.id.editText2);
                    int a = Integer.parseInt(et1.getText().toString());
                    int b = Integer.parseInt(et2.getText().toString());
                    try {
                        res = String.valueOf(myservice.area(a, b));
                    } catch (RemoteException e) {
                        e.printStackTrace();
                    }
                    TextView tv = (TextView)findViewById(R.id.textView4);
                    tv.setText(res);
                }
            }
        );
    }
```

```
    public void drawWidget() {
        setContentView(R.layout.widget);
        final Activity activity = this;
        Toast tst = Toast.makeText(activity, "widget レイアウトを表示します。",
                                   Toast.LENGTH_SHORT);
        tst.show();

        RatingBar rtb = (RatingBar)this.findViewById(R.id.ratingBar1);
        rtb.setOnRatingBarChangeListener(
            new OnRatingBarChangeListener() {

                @Override
                public void onRatingChanged(RatingBar ratingBar, float rating,
                                            boolean fromUser) {
                    Toast tst = Toast.makeText(activity, "評価が"+rating+
                                            "に変更されました", Toast.LENGTH_SHORT);
                    tst.show();
                }
            }
        );

        RadioGroup rg = (RadioGroup)this.findViewById(R.id.radioGroup1);
        rg.setOnCheckedChangeListener(
            new OnCheckedChangeListener() {

                @Override
                public void onCheckedChanged(RadioGroup group, int checkedId) {
                    RadioButton rbtn = (RadioButton)findViewById(checkedId);
                    Toast tst = Toast.makeText(activity, "現在"+rbtn.getText(),
                                            Toast.LENGTH_SHORT);
                    tst.show();
                }
            }
        );
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

}
```

『保管』のアイコンをクリックして，MainActivity. java を上書き保存し，パッケージ・エクスプローラーで『First』を選択して，実行ボタンをクリックする。



提出物：

1）画面のレイアウト設定ファイル service. xml

2）インターフェイスを定義したファイル MyService. aidl

3）インターフェイスの機能を実装したクラスのソースファイル MyAreaService. java

4）Android アプリの目録ファイル AndroidManifest. xml