■実習方法２

HopProにHop020.javaを追加して、次のテンプレートソースを入力し、実習５から実習７までの実習課題を指定された場所へソースを追加入力して、コンパイルし、実行結果を確認してください。

プロジェクト名（既存）： HopPro　　アプリケーション名：Hop020.java

```
/*    年    組    席  名前
 * Hop020 実習5,6,7
 */
package jp.edu.mie;
import android.app.Activity;
import android.app.AlertDialog;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class Hop020 extends Activity
{
    EditText numericalvalue1;
    EditText numericalvalue2;
    private Button button5;  //ボタン
    private Button button6;  //ボタン
    private Button button61; //ボタン
    private Button button62; //ボタン
    private Button button7;  //ボタン

    //アプリの初期化
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        button5=(Button)findViewById(R.id.button_switch);
        button6=(Button)findViewById(R.id.button_for);
        button61=(Button)findViewById(R.id.button_while);
        button62=(Button)findViewById(R.id.button_dowhile);
        button7=(Button)findViewById(R.id.button_array);
        numericalvalue1=(EditText)findViewById(R.id.edittext_numericalvalue1);
        numericalvalue2=(EditText)findViewById(R.id.edittext_numericalvalue2);

        //リスナの通知
        button5.setOnClickListener(new HopClickListener());
        button6.setOnClickListener(new HopClickListener());
```

```
        button61.setOnClickListener(new HopClickListener());
        button62.setOnClickListener(new HopClickListener());
        button7.setOnClickListener(new HopClickListener());
    }

    //ダイアログの表示
    private void showDialog(HopClickListener  h, String title, String text)
    {
        AlertDialog.Builder ad=new AlertDialog.Builder(this);
        ad.setTitle(title);
        ad.setMessage(text);
        ad.setPositiveButton("OK",null).show();
    }

    //ボタンクリックイベントの処理
    public class HopClickListener implements OnClickListener
    {
        int i;
        //ボタンクリックイベントの処理
        public void onClick(View view)
        {
            int num1=Integer.parseInt(numericalvalue1.getText().toString());
            int num2=Integer.parseInt(numericalvalue2.getText().toString());
            if (view == button5)
            {
                //【実習5】条件分岐2


            }else if (view == button6){
                //【実習6】繰り返し処理 for



            }else if(view == button61){
                //【実習6-1】繰り返し処理 while



            }else if(view == button62){
                //【実習6-2】繰り返し処理 do



            }else if(view == button7){
                //【実習7】配列


            }
```

```
        }
    }
}
```

プロジェクト名（既存）：HopPro　　ファイル名：AndroidManifest.xml

```
<?xml version="1.0"  encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="jp.edu.mie"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10"></uses-sdk>
    <application android:icon="@drawable/icon"
                 android:label="@string/app_name">
        <activity android:name=".Hop010"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Hop020"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

プロジェクト名（既存）：HopPro　　ファイル名：Main.xml

```
<?xml version="1.0"  encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"   >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/label_input1" />
    <EditText
        android:id="@+id/edittext_numericalvalue1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
```

```
        android:inputType="number"
        android:maxLength="5"
        android:text="@string/label_fixednumber1" />
    <Button
        android:id="@+id/button_switch"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_switch" />
    <Button
        android:id="@+id/button_for"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_for" />
    <Button
        android:id="@+id/button_while"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_while" />
    <Button
        android:id="@+id/button_dowhile"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_dowhile" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/label_input2" />
    <EditText
        android:id="@+id/edittext_numericalvalue2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:inputType="number"
        android:maxLength="5"
        android:text="@string/label_fixednumber1" />
    <Button
        android:id="@+id/button_array"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_array" />
</LinearLayout>
```

プロジェクト名（既存）：HopPro　　ファイル名：Strings.xml

```
<?xml version="1.0"  encoding="utf-8"?>
<resources>
    <string name="app_name">Hop</string>
    <string name="label_input1">５桁以下の数値を入力してください</string>
    <string name="button_switch">【実習５】switch 文の実行結果</string>
```

```
    <string name="button_for">【実習６】for 文の実行結果</string>
    <string name="button_while">【実習６－１】while 文の実行結果</string>
    <string name="button_dowhile">【実習６－２】dowhile 文の実行結果</string>
    <string name="label_input2">【実習７】５桁以下の数値を入力してください
    </string>
    <string name="button_array">【実習７】配列文の実行結果</string>
    <string name="label_fixednumber1">0</string>
</resources>
```

Androidの参考資料

１　レイアウトの作成手順

1.1 画面をあらわすアクティビティクラスを拡張する。

public class Hop020 extends Activity

画面を持ったアプリケーションを作成する。

1.2 アクティビティクラスのメソッドを定義する。

public void onCreate(Bundle savedInstanceState)

アプリケーションの画面が起動されたときに呼び出されるメソッド（オブジェクトを処理する命令）

アプリケーションの画面が起動されたときに、このメソッド内に記述したコードが処理される。

1.3 アクティビティにレイアウトを設定する。

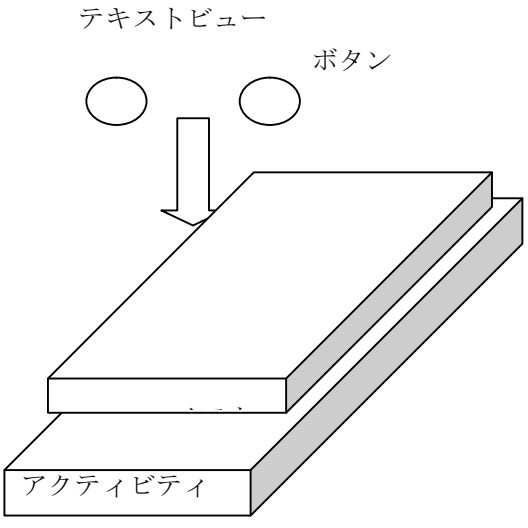
LinearLayout layout=new LinearLayout(this);

setContentView(layout);

1.4 ビューを作成する。

setContentView(R.layout.main);

ビューとは画面に実際に配置する部品のこと（テキストビュー、ボタンなど）



２　エディトテキストについて

ユーザーからの入力を受け付ける機能です。

３　ダイアログについて

ユーザーに対してメッセージを表示したり、簡単な入力を行う機能です。

【実習５】　条件分岐２

switch 命令で条件分岐をします。

１　ソースコード

テンプレート Hop020 の「//【実習 5】条件分岐 2」の箇所へ次のソースを追加入力します。

```
switch(num1)
{
    case 1:
        showDialog(this, "【実習 5】", num1 + "はビジネス科です。");
        break;
    case 2:
        showDialog(this, "【実習 5】", num1 + "は情報システム科です。");
        break;
    default:
        showDialog(this, "【実習 5】", num1 +
                                        "は科に該当しない数値です。");
}
```

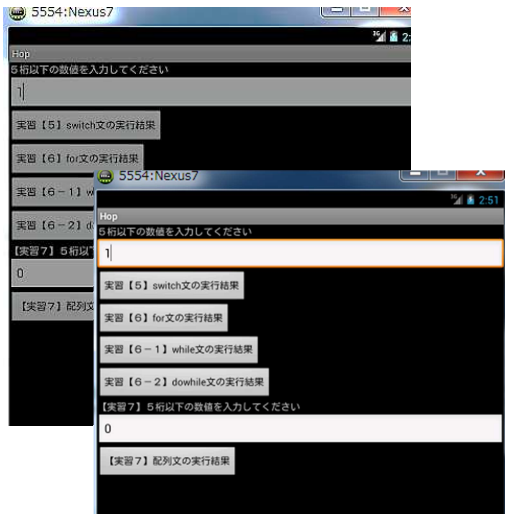
２　実行結果

数値を入力する画面が表示されるので、１を入力すると、「１はビジネス科です。」と表示され、２を入力すると、「２は情報システム科です。」と表示されます。

その他の数字の場合は、（数字）は科に該当しない数値です。」と表示されます。

３　解説

「break」というのは break 以下の処理を飛ばしてブロックを抜け出すという意味です。case 以下の文は break までが実行され、break 文によって switch 文のブロックを抜け出します。




```
switch　（式）{
    case　値１：　文（複数可）；
        break；
    case　値２：　文（複数可）；
        break；
    case　値３：　文（複数可）；
        break；
    default：　文（複数可）；
        break；
}
```

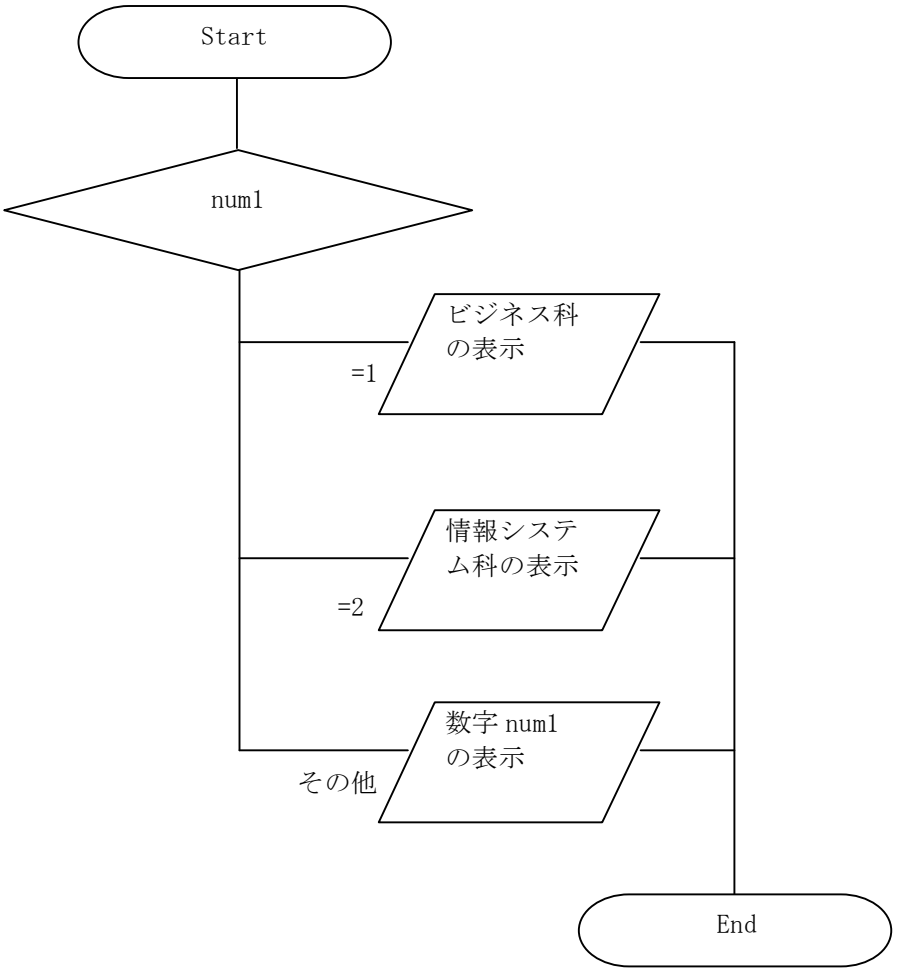
(参考)

```
//ダイアログの表示
private void showDialog(HopClickListener h, String title, String text)
{
    AlertDialog.Builder ad=new AlertDialog.Builder(this); //ダイアログの作成
    ad.setTitle(title);   //タイトル設定する
```

```
    ad.setMessage(text);  //メッセージを設定する
    //OK ボタンを設定してダイアログを表示する
    ad.setPositiveButton("OK",null).show();
}
```

AlertDialog は、メッセージを表示する際などに使うことができます。AlertDialog.Builder にはダイアログの設定を行い、自分自身を返すメソッドがあるため、続けて設定を行うメソッドを呼び出し、最後に show()メソッドで表示することができます。

４　流れ図



【演習】

■次のプロジェクトとプログラムを作成してください。

プロジェクト名：HopPro（既存のプロジェクト）

アプリケーション名：HopEx050

HopPro に HopEx050.java を追加して、次のテンプレートソースを入力し、演習 51 から演習 64 までの実習課題を指定された場所へソースを追加入力して、コンパイルし、実行結果を確認してください。

プロジェクト名（既存）： HopPro　　アプリケーション名：HopEx050.java

```
/*    年    組    席   名前
 * HopEx050 演習 51,61,62,63,64
 */

package jp.edu.mie;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.graphics.Color;
import android.os.Bundle;
import android.text.InputFilter;
import android.text.InputType;
import android.text.method.DigitsKeyListener;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;

//エディットテキスト
public class HopEx050 extends Activity implements View.OnClickListener
{
    private EditText     editText1;//エディットテキスト
    private Button       button51;//ボタン
    private Button       button61;//ボタン
    private Button       button62;//ボタン
    private Button       button63;//ボタン
    private Button       button64;//ボタン
```

```
    //アプリの初期化
    @SuppressWarnings("deprecation")
    @Override
    public void onCreate(Bundle icicle)
    {
        super.onCreate(icicle);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        //レイアウトの生成
        LinearLayout layout=new LinearLayout(this);
        layout.setBackgroundColor(Color.rgb(255,255,255));
        layout.setOrientation(LinearLayout.VERTICAL);
        setContentView(layout);
        DigitsKeyListener digitsKeyListener1 =
                                            new DigitsKeyListener(false, true);
        TextView textView1=new TextView(this);
        textView1.setText("５桁以下の数値を入力してください");
        textView1.setTextColor(Color.rgb(0,0,0));
        setLLParams(textView1);
        layout.addView(textView1);
        //５桁以下の数値のみの入力
        editText1=new EditText(this);
        editText1.setKeyListener(digitsKeyListener1);
        editText1.setInputType(InputType.TYPE_CLASS_NUMBER);
        editText1.setFilters(new InputFilter[] {
            new InputFilter.LengthFilter(5)});
        editText1.setText("0",EditText.BufferType.NORMAL);
        setLLParams(editText1,
            LinearLayout.LayoutParams.FILL_PARENT,
            LinearLayout.LayoutParams.WRAP_CONTENT);
        layout.addView(editText1);
        //ボタンの生成
        button51=new Button(this);
        button51.setText("【演習 51】結果表示");
        button51.setOnClickListener(this);
        setLLParams(button51);
        layout.addView(button51);
        //ボタンの生成
        button61=new Button(this);
        button61.setText("【演習 61】結果表示");
        button61.setOnClickListener(this);
        setLLParams(button61);
        layout.addView(button61);
        //ボタンの生成
        button62=new Button(this);
        button62.setText("【演習 62】結果表示");
        button62.setOnClickListener(this);
        setLLParams(button62);
        layout.addView(button62);
        //ボタンの生成
        button63=new Button(this);
```

```
        button63.setText("【演習 63】結果表示");
        button63.setOnClickListener(this);
        setLLParams(button63);
        layout.addView(button63);
        //ボタンの生成
        button64=new Button(this);
        button64.setText("【演習 64】結果表示");
        button64.setOnClickListener(this);
        setLLParams(button64);
        layout.addView(button64);
    }

    //ダイアログの表示
    private static void showDialog(final Activity activity,
        String title,String text)
    {
        AlertDialog.Builder ad=new AlertDialog.Builder(activity);
        ad.setTitle(title);
        ad.setMessage(text);
        ad.setPositiveButton("OK",new DialogInterface.OnClickListener()
        {
            public void onClick(DialogInterface dialog,int whichButton)
            {
                activity.setResult(Activity.RESULT_OK);
            }
        });
        ad.create();
        ad.show();
    }

    //ライナーレイアウトのパラメータ指定
    private static void setLLParams(View view)
    {
        view.setLayoutParams(new LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.WRAP_CONTENT,
            LinearLayout.LayoutParams.WRAP_CONTENT));
    }

    //ライナーレイアウトのパラメータ指定
    private static void setLLParams(View view,int w,int h) {
        view.setLayoutParams(new LinearLayout.LayoutParams(w,h));
    }

    //ボタンクリックイベントの処理
    public void onClick(View view)
    {
        int num1=Integer.parseInt(editText1.getText().toString());
        int su1;
        int i;
```

```
        if (view==button51)
        {
             //【演習 51】


        }else if (view==button61)
        {
             //【演習 61】


        }else if(view==button62)
        {
            //【演習 62】


        }else if(view==button63)
        {
            //【演習 63】


        }else if(view==button64)
        {
            //【演習 64】


        }
    }
}
```

プロジェクト名（既存）：HopPro　　ファイル名：AndroidManifest.xml

```
<?xml version="1.0"  encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="jp.edu.mie"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10"></uses-sdk>
    <application android:icon="@drawable/icon"
                 android:label="@string/app_name">

        ・・・

        <activity android:name=".HopEx050"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
```

```
        </activity>
    </application>
</manifest>
```

【演習51】

次の場合に応じて次のようなメッセージを出力するコードを作成してください。
（switch 文を使用する）

入力した値が偶数だった場合・・・・・「〇は偶数です。」

入力した値が奇数だった場合・・・・・「〇は奇数です。」

ただし〇は入力した整数

入力した整数　1　は奇数です

変数 num1 に入力した値を表示する。

・変数の型、名前は適宜使用してください。

・画面上に入力する枠がありますので、手動で入力してください。入力した値は、すでに変数 num1 に代入していますので新規に書く必要はありません。なお、int num1;はテンプレートのプログラム中で設定済です。

・数値の 0 は偶数です。（-1 は奇数、-2 は偶数）

なお、このプログラムの入力値には「マイナス（－）記号」は使えません。