



1. viewのクリックイベント処理
2. Intentにより、画面遷移を実装
3. クラスの継承
4. オブジェクトのキャスト
5. Listviewの実装

1. viewのクリックイベント処理

```
TextView btnView = (TextView) findViewById(R.id.btn);
if (btnView != null) {
    // クリック時の処理

    btnView.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            //ここに処理を書く
        }
    });
}
```

//以下のような書き方も出来る。

```
// View.OnClickListener myClickListener = new View.OnClickListener() {
//     public void onClick(View v) {
//         //ここに処理を書く
//     }
// };
//
// クリック時の処理
// greetingView.setOnClickListener(myClickListener);
```

```
//javascriptの書き方と考え方は同じ
//$("#a").click(function(){
// alert("aタグ .click() のイベント！");
//});
```


2. Intentにより、画面遷移を実装

MainActivityからSubActivityへと画面遷移させる。
SubActivityを予め作っておく。

AndroidManifest.xml

- ・ Activity作成段階で、AndroidManifest.xmlにActivityが既に登録されている。
- ・ スタートアップActivityの記述を確認
- ・ アプリアイコンの設定
- ・ アプリ名の

MainActivityのクリックアクション内で、以下を記述することで、SubActivityへ画面遷移する。

```
Intent intent = new Intent(getApplicationContext(),  
SubActivity.class);  
startActivity(intent);
```

1. クラスの継承とキャスト

MySelfクラスはHumanクラスを継承させる。
HumanクラスはMySelfクラスの親クラス（スーパークラス）である。
MySelfクラスはHumanクラスの子クラス（サブクラス）である。

Humanクラスに名前・年齢のセッターとゲッターを用意する。

MySelfクラスに、挨拶関数を用意する。

1. クラスの継承とキャスト

MySelf.java

```
public class MySelf extends Human{

    public String doGreeting () {
        String greeting = "こんにちは！"+getName+"です。 \n";
        greeting += "年齢は"+getAge+"歳です。 ";
        return greeting;
    }
}
```

Human.java

```
public class Human {
    private int age;
    private String name;

    //年齢を設定
    public void setAge (int age) {
        this.age = age;
    }

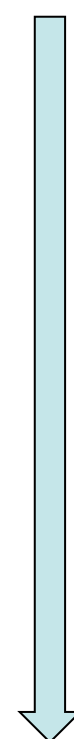
    //年齢を取得
    public int getAge () {
        return this.age;
    }

    //名前を設定
    public void setName (String name) {
        this.name = name;
    }

    //名前を取得
    public String getName () {
        return this.name;
    }
}
```

3. クラスの継承

親クラス(スーパークラス)
Humanクラス



子クラス (サブクラス)
MySelfクラス

メンバー変数
`int age;`

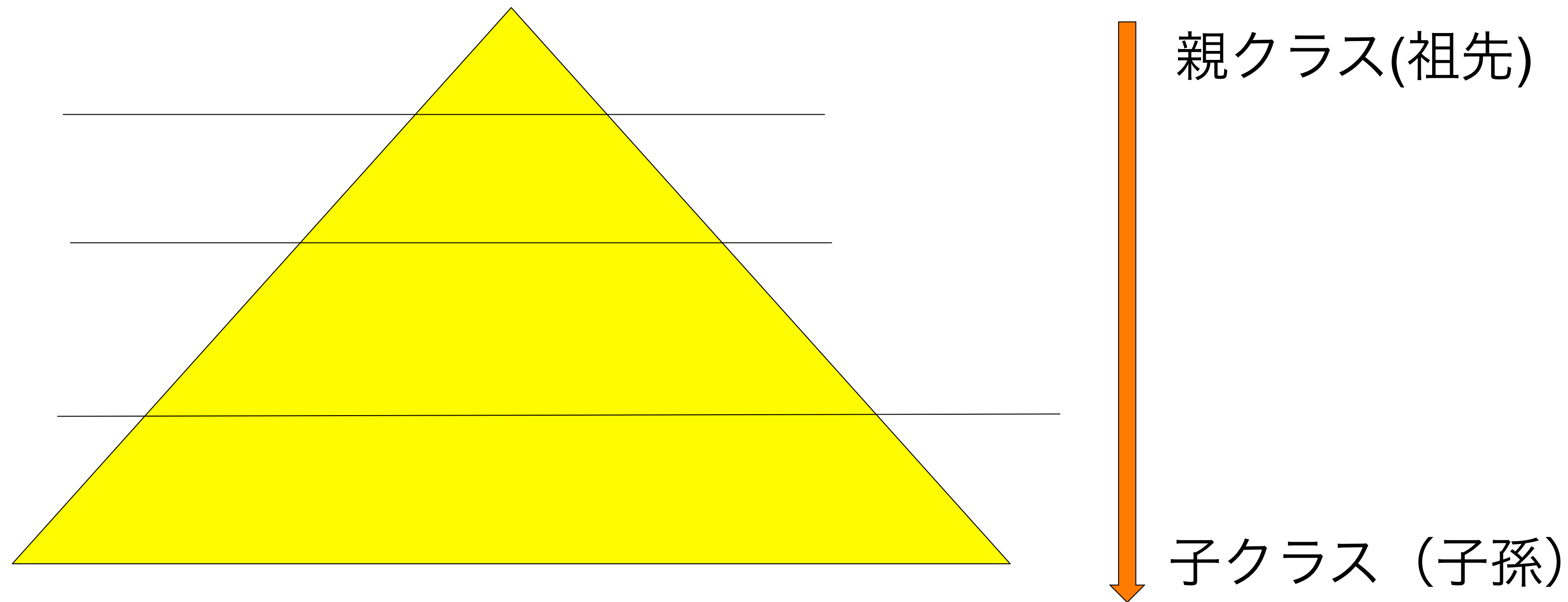
メンバー関数
`setAge()`
`getAge()`

親クラスの
メンバー変数
メンバー関数
をそのまま引き継ぐことが出来る。

*親クラス各メンバー変数、メンバー関数の
アクセス修飾子がprivate以外の場合

3.クラスの継承

上から下へと継承していくイメージ！！
つまり、下へ行くにつれて、機能が增強していくイメージです！！



4.オブジェクトのキャスト

キャストとは、データの型を変換すること。

暗黙的キャスト（アップキャスト）

//機能が減る

//サブクラスからスーパークラスへの型変換は暗黙的に行われる

明示的キャスト（ダウンキャスト）

//機能が増える

//スーパークラスからサブクラスへと型変換をするには明示的にキャストが必要になる。

4.オブジェクトのキャスト

MySelfとHumanの継承とキャスト

暗黙的キャスト（アップキャスト）

```
MySelf mySelf = new MySelf(this);  
Human human = mySelf;  
//機能が減る  
//HumanはMySelfの親クラスである。  
//サブクラスからスーパークラスへの型変換は暗黙的に行われる
```

明示的キャスト（ダウンキャスト）

```
Human human = new Human();  
//機能が増える  
//スーパークラスからサブクラスへと型変換をするには明示的にキャストが必要になる。  
mySelf = (MySelf) human;
```

4.オブジェクトのキャスト

Viewクラス

```
View view = findViewById(R.id.btn);
TextView btnView = (TextView)view;
```

TextViewクラスはViewクラスを継承したクラス。

findViewById()はActivityクラスが持つ関数。

findViewById()で取り出せるオブジェクトはデータ型Viewなので、サブクラスのTextViewにダウンキャストしている。

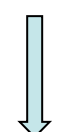
つまりTextViewはViewのサブクラス。

TextViewクラスはViewクラスには無いメンバー関数やメンバー変数を持っている。

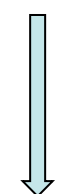
つまり機能が增強されている。

Viewと各子クラス間
(親または孫と子間)
のキャストは可能だが、
子同士のキャストは不可

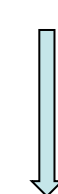
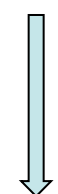
親クラス(スーパークラス)



親クラス(スーパークラス)
Viewクラス



継承



子クラス (サブクラス)
TextViewクラス

子クラス (サブクラス)
ImageViewクラス

子クラス (サブクラス)
LinearLayoutクラス

4.オブジェクトのキャスト

ContextとActivityのthis 暗黙的キャスト

```
Context context = getApplicationContext();
```

```
//サブクラスからスーパークラスへの型変換は暗黙的に行われる
//thisはActivityであり、Contextクラスの親クラスである。
context = this;
```

明示的キャスト

```
Context context = getApplicationContext();
```

```
//スーパークラスからサブクラスへと型変換をするには
//明示的にキャストが必要になる。
Activity act = (Activity) context;
```

親クラス(スーパークラス)
Contextクラス

継承 ↓

子クラス (サブクラス)
Activityクラス

4.オブジェクトのキャスト

//Context:アプリ全体の状態を持っていて、何から起動されたかどういう状態か、何にアクセスしようとしているか、といった情報を受け渡す

```
Context context = getApplicationContext();
```

// サブクラスからスーパークラスへの型変換は暗黙的に行われる
context = this;//thisはMainActivity

// サブクラスからスーパークラスへとアップキャスト。暗黙的にキャスト
Activity act = this;

// スーパークラスからサブクラスへダウンキャスト
act = (Activity) context;

// キャストなし。thisはMainActivity
MainActivity mainActivity = this;

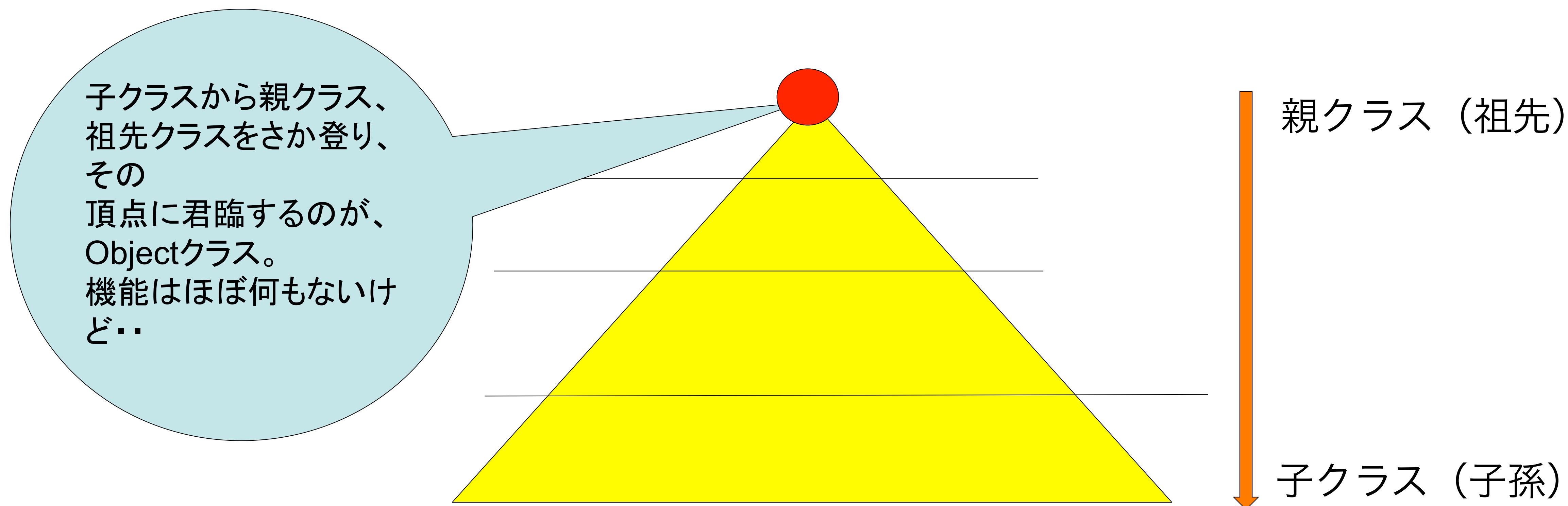
//ContextからMainActivityへダウンキャスト
mainActivity = (MainActivity) getApplicationContext();

```
Context context = this;  
Intent intent = new Intent(getApplicationContext(), SubActivity.class);  
startActivity(intent);
```

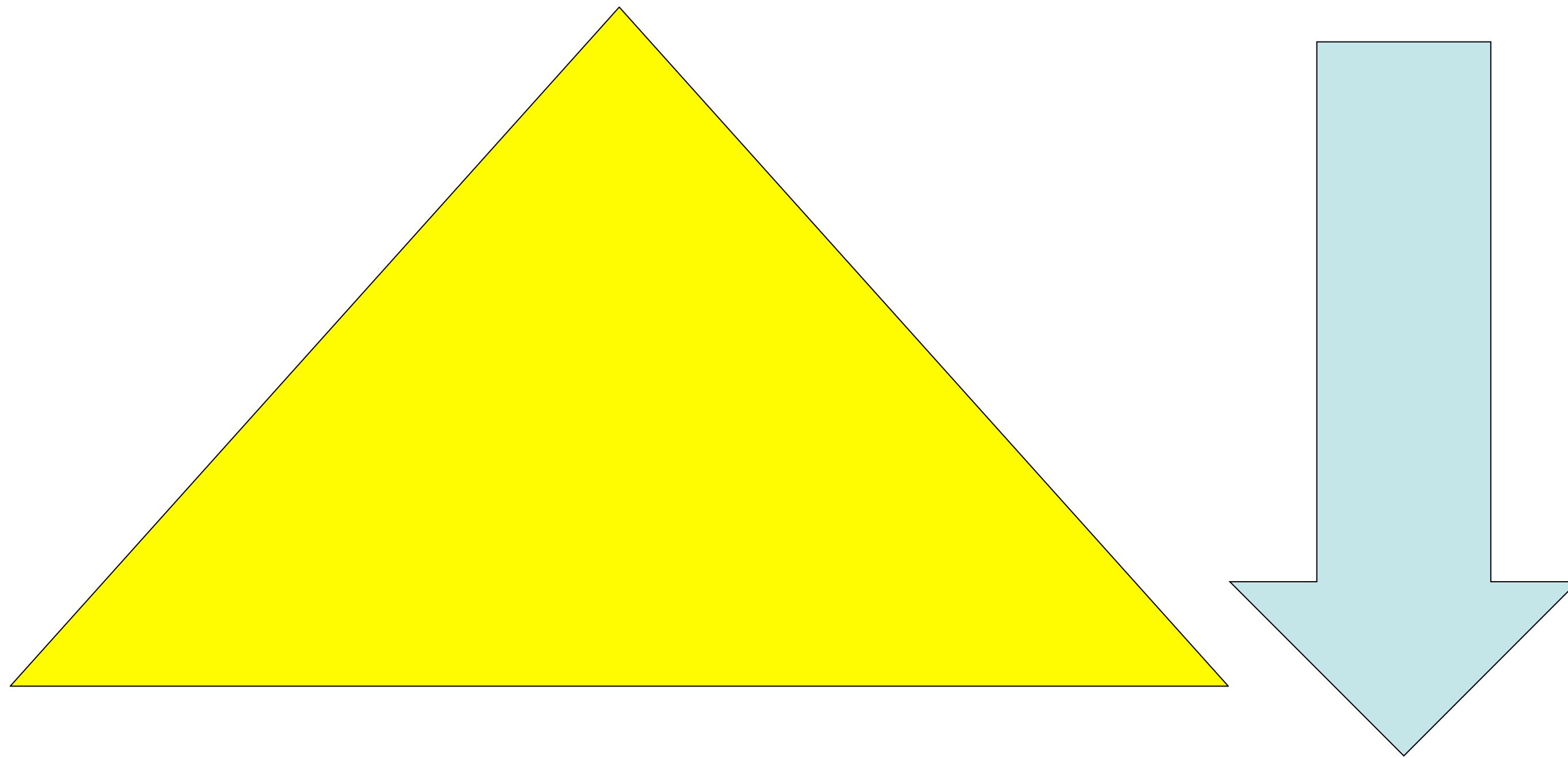
this,act,contextどれも
OKだが、
getApplicationContext()
で覚えておけばよい！

4.オブジェクトのキャスト / クラスの継承

上から下へと機能が継承されて、機能が增強していくイメージです！！



機能を増強するなら明示しろ！！！！



Adapterとは:

ListViewと表示するデータ（配列オブジェクト）を結びつける。
配列オブジェクトの各要素のデータを取り出し、リスト一つ分のレイアウトを作成し、ListViewにそのレイアウトを追加する。

配列オブジェクトの要素数分繰り返すことによって、
リストを完成させる。

```
ListView lv = (ListView) findViewById(R.id.listView1);

String[] members = { "Tokyo", "Fukushima", "Osaka", "Kanagawa", "Nagano", "Nara", "Shizuoka" };

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_expandable_list_item_1, members);

lv.setAdapter(adapter);
```