

MICROOH 麦可网

Android-从程序员到架构师之路

出品人：Sundy

讲师：高焕堂（台湾）

<http://www.microoh.com>

A02_c

架构设计的UML图形思考(c) (Graphic Thinking)

By 高煥堂

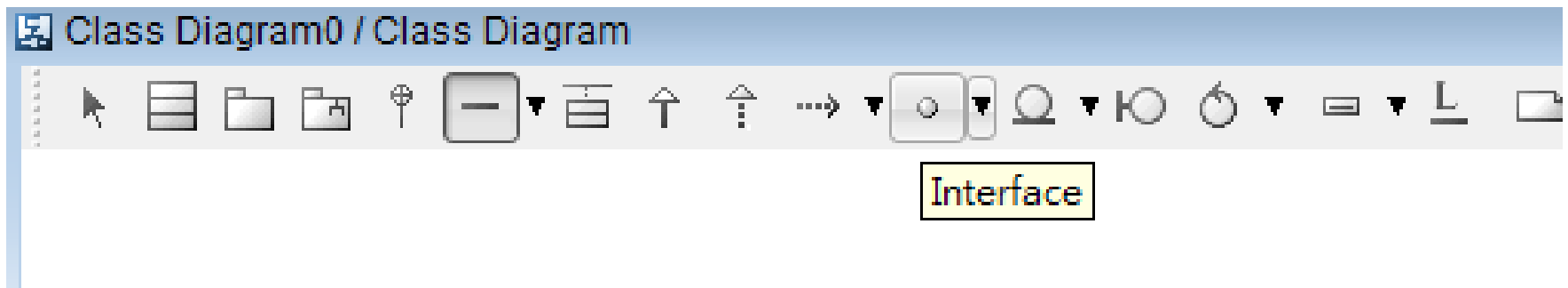
4、绘制UML类别图： 表达接口(Interface)

接口的表达

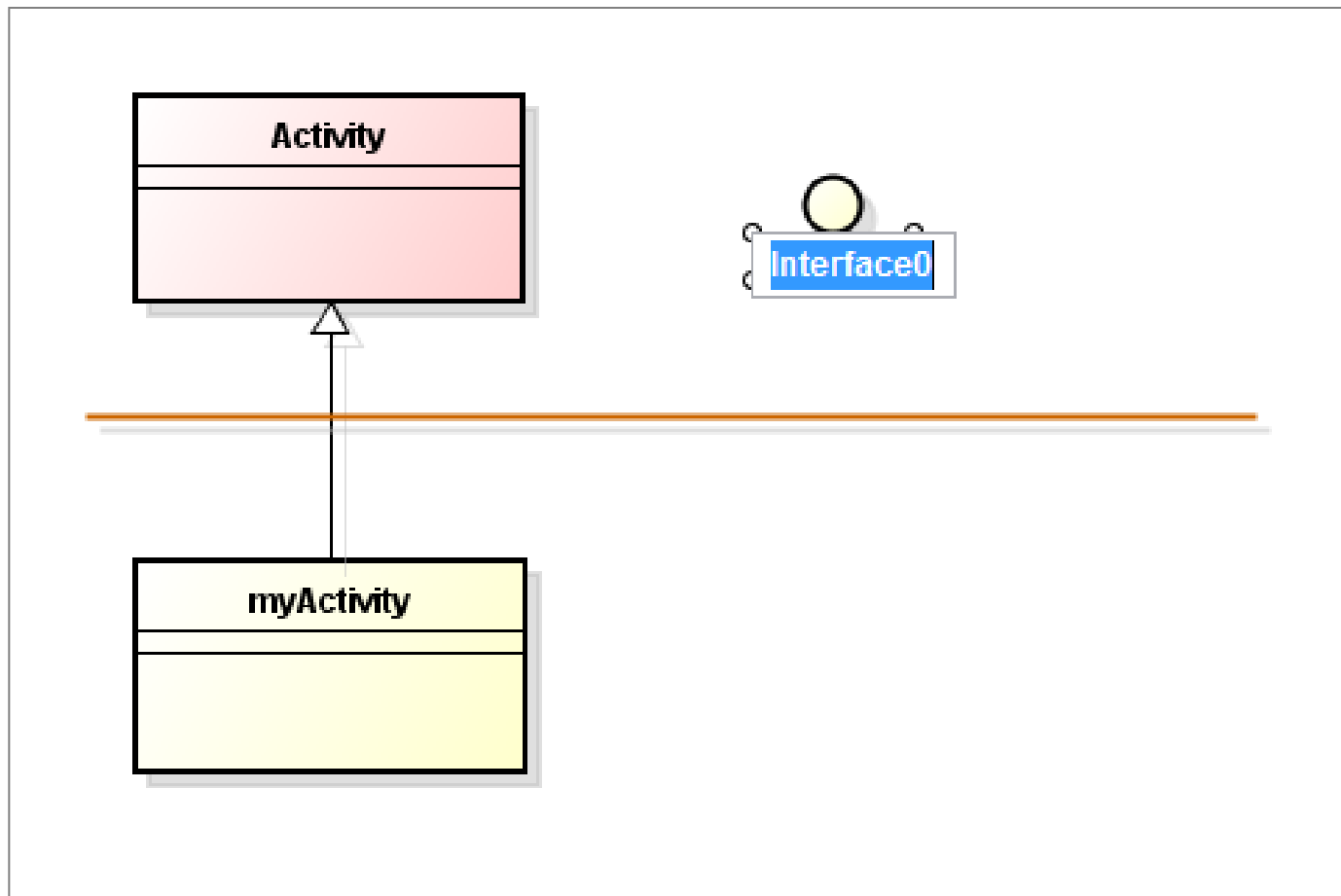
- ◎ 对于架构师而言，「接口」(Interface)的角色比「类别」(Class)来得重要多了。
- ◎ 例如，对照到大家所熟悉的代码：

```
public class myActivity extends Activity  
                implements OnClickListener {  
    // .....  
    // .....  
}
```

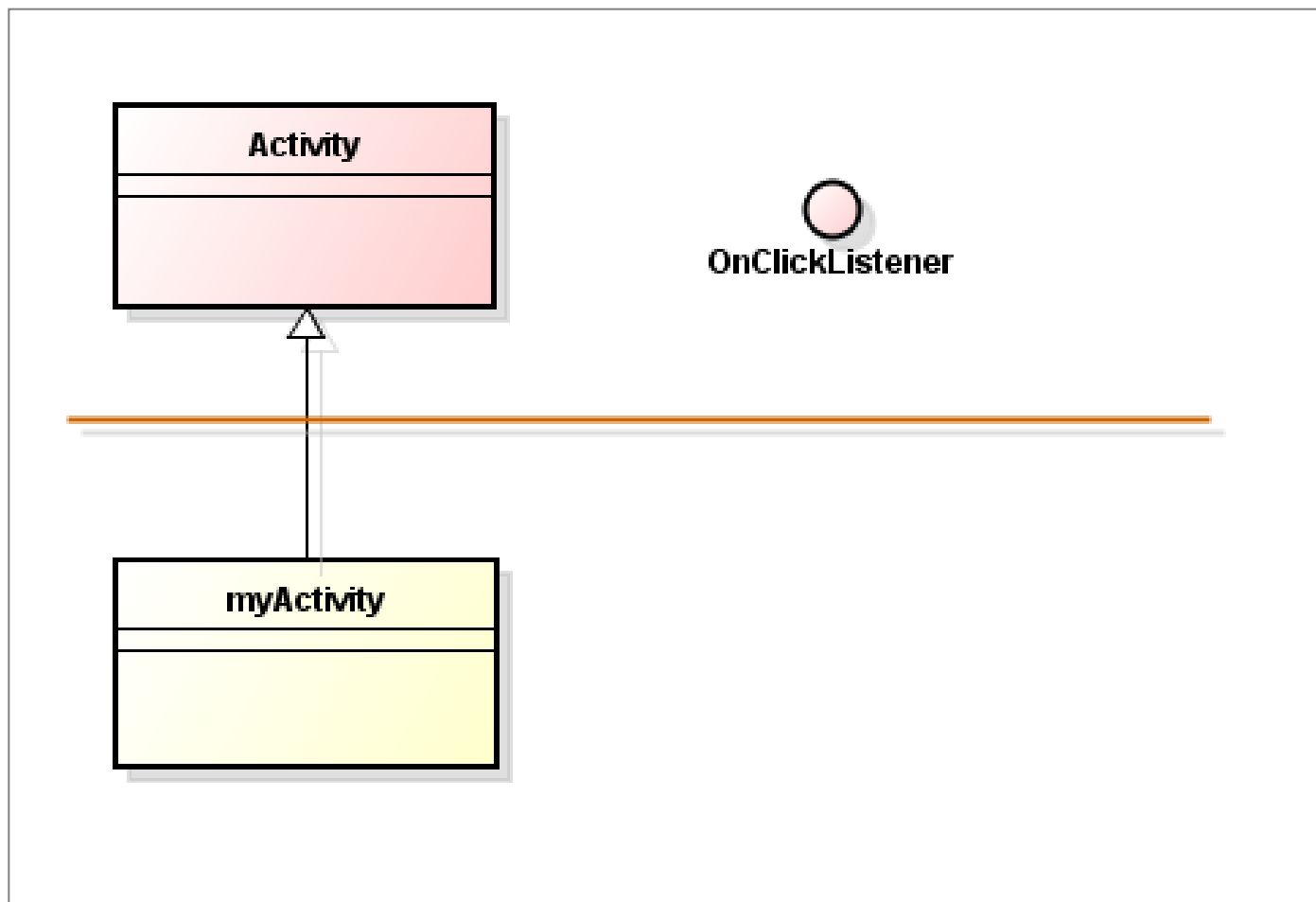
- 于是，就来看看如何将接口图素呈现类别图上。例如，选取<Interface>图素，如下：



- 点选了这个<Interface>图素，接着将鼠标移动到类别图里的任何位置，并按键，出现如下：

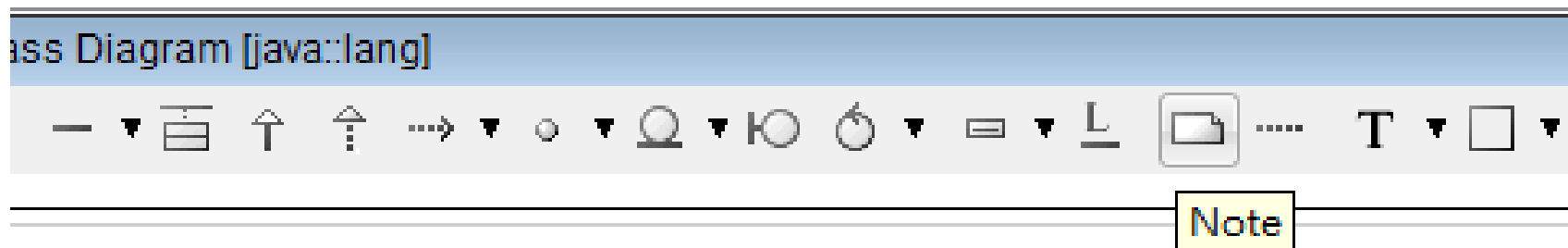


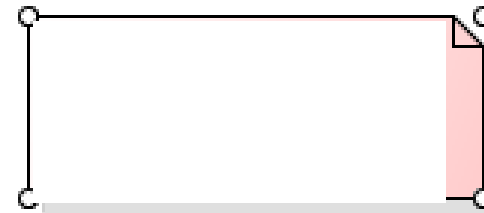
◎ 就可以填入接口的名称了，例如：



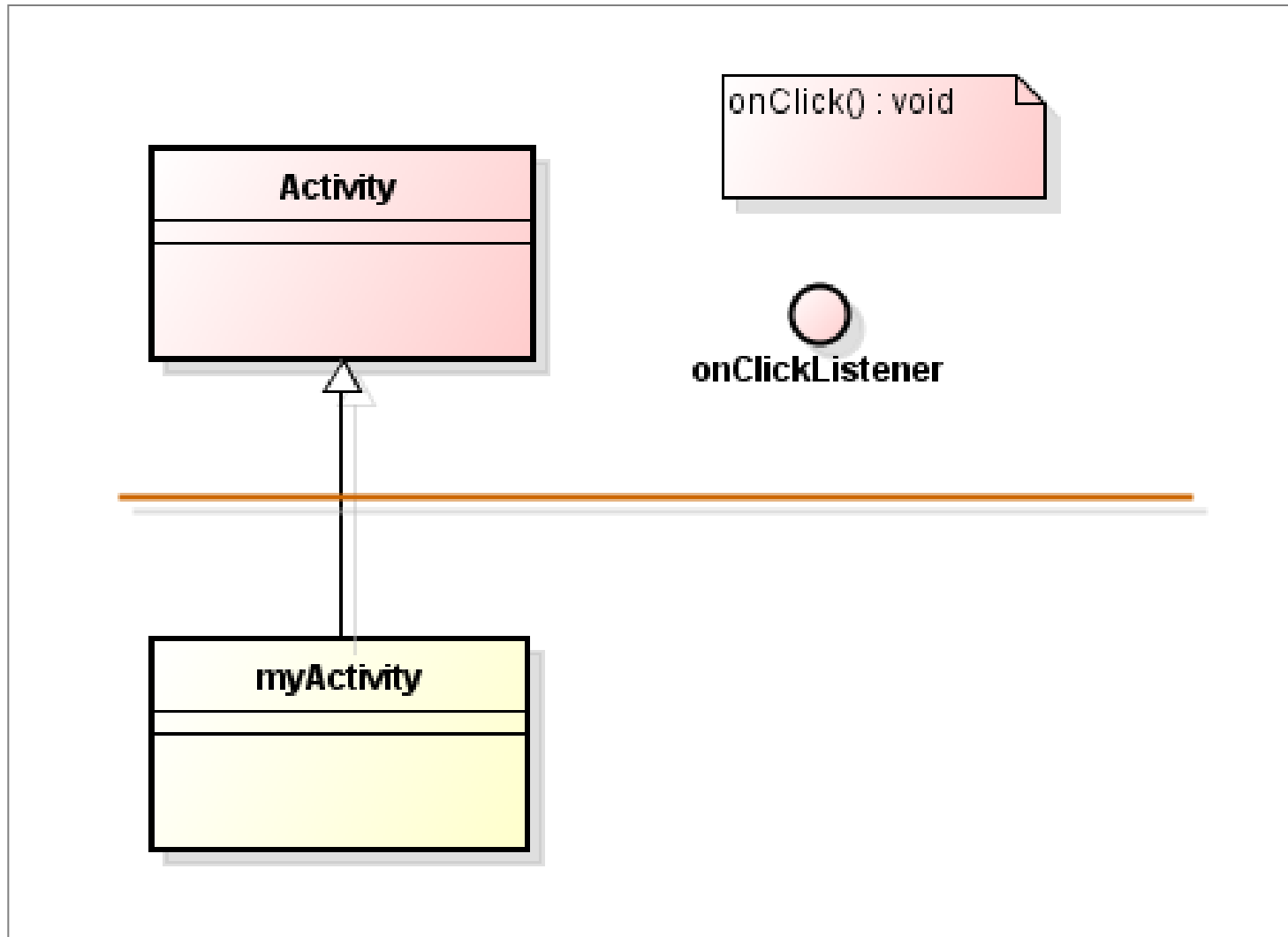
接口的注释(Note)

◎ 可选取<Note>图素，如下：

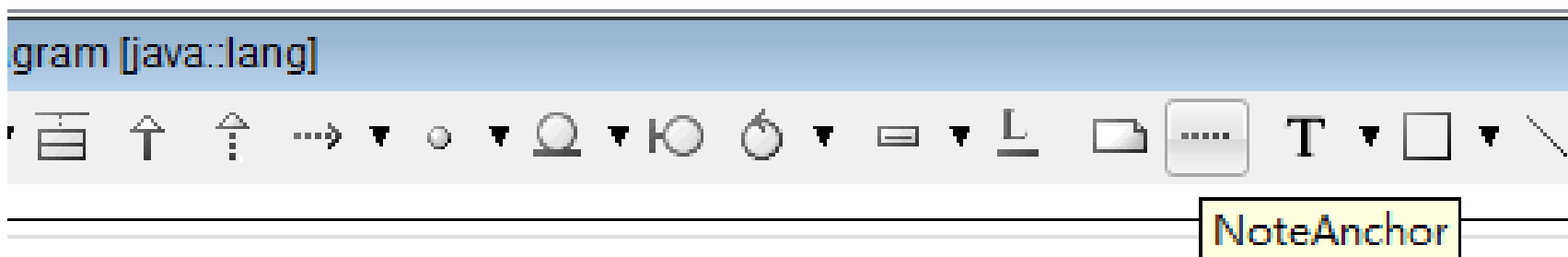


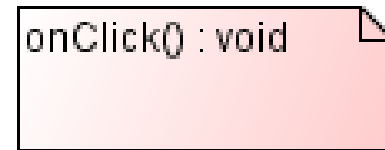


OnClickListener

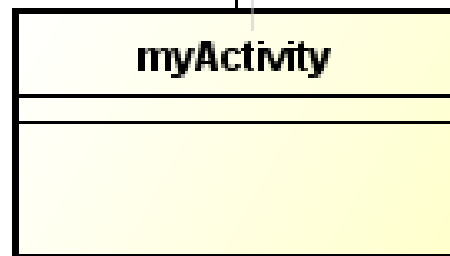


◎ 接着，可选取<NoteAnchor>图素，如下：





onClickListener

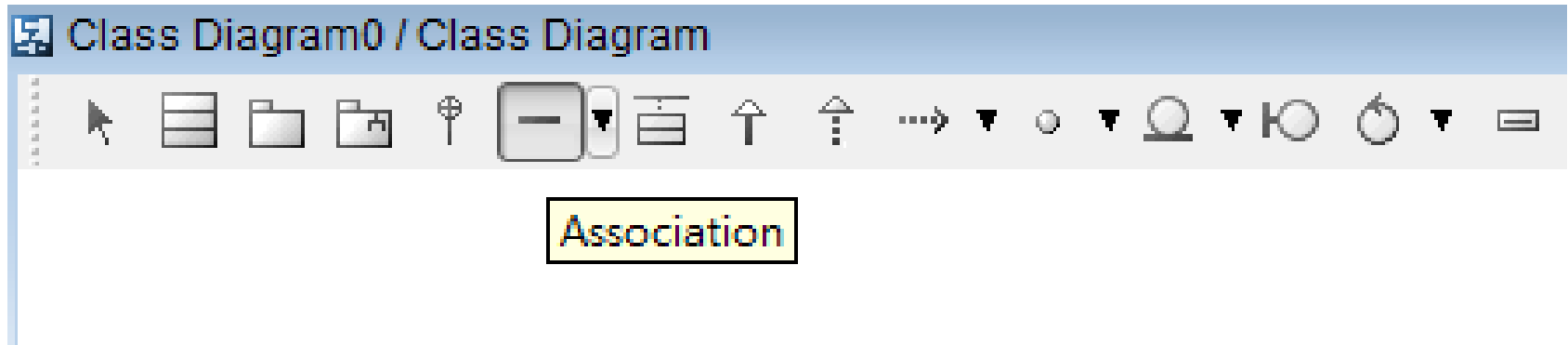


类<实现>接口的表达

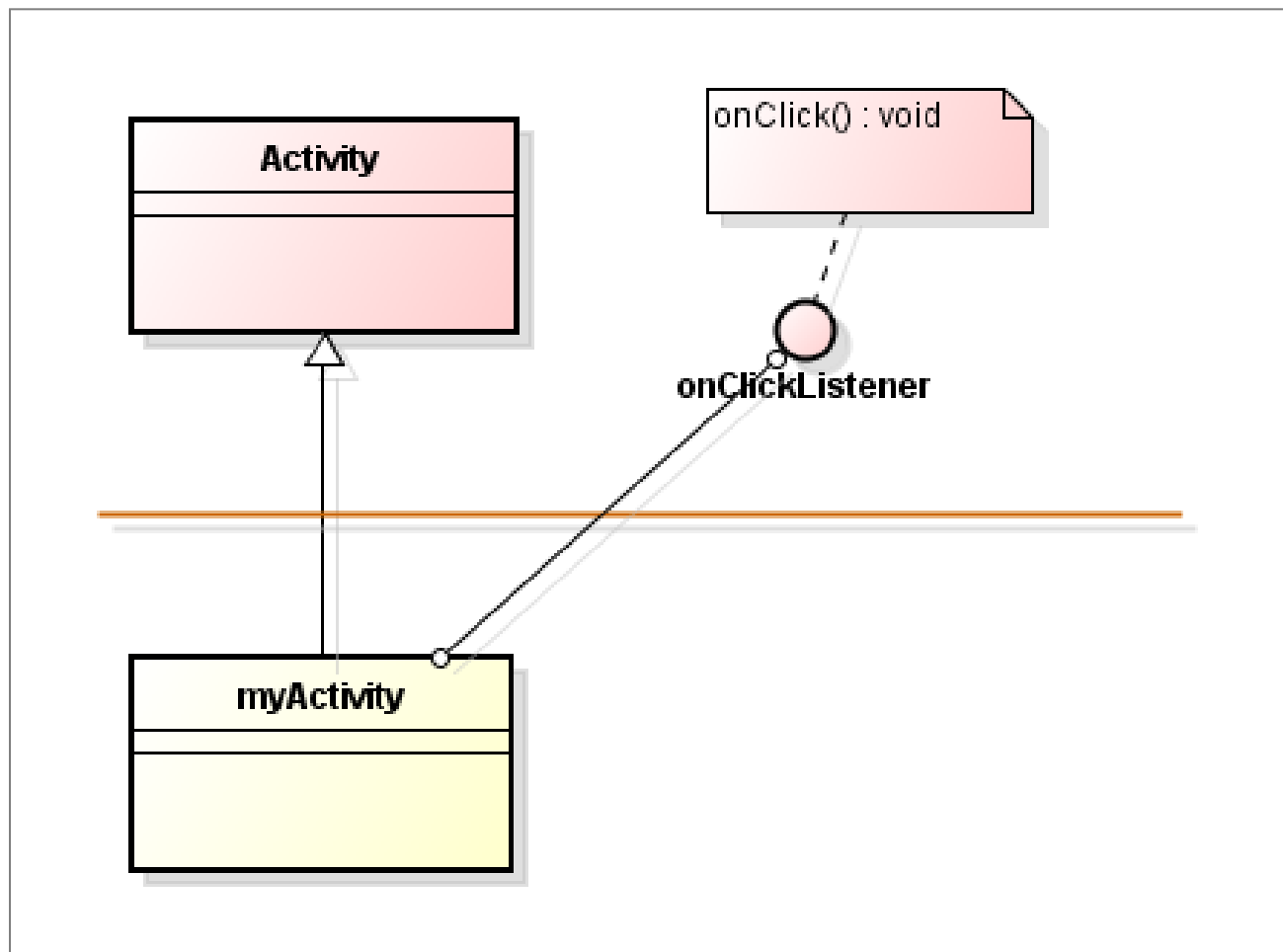
◎ 对照到大家所熟悉的代码：

```
public class myActivity extends Activity
    implements OnClickListener {
    // .....
    // .....
}
```

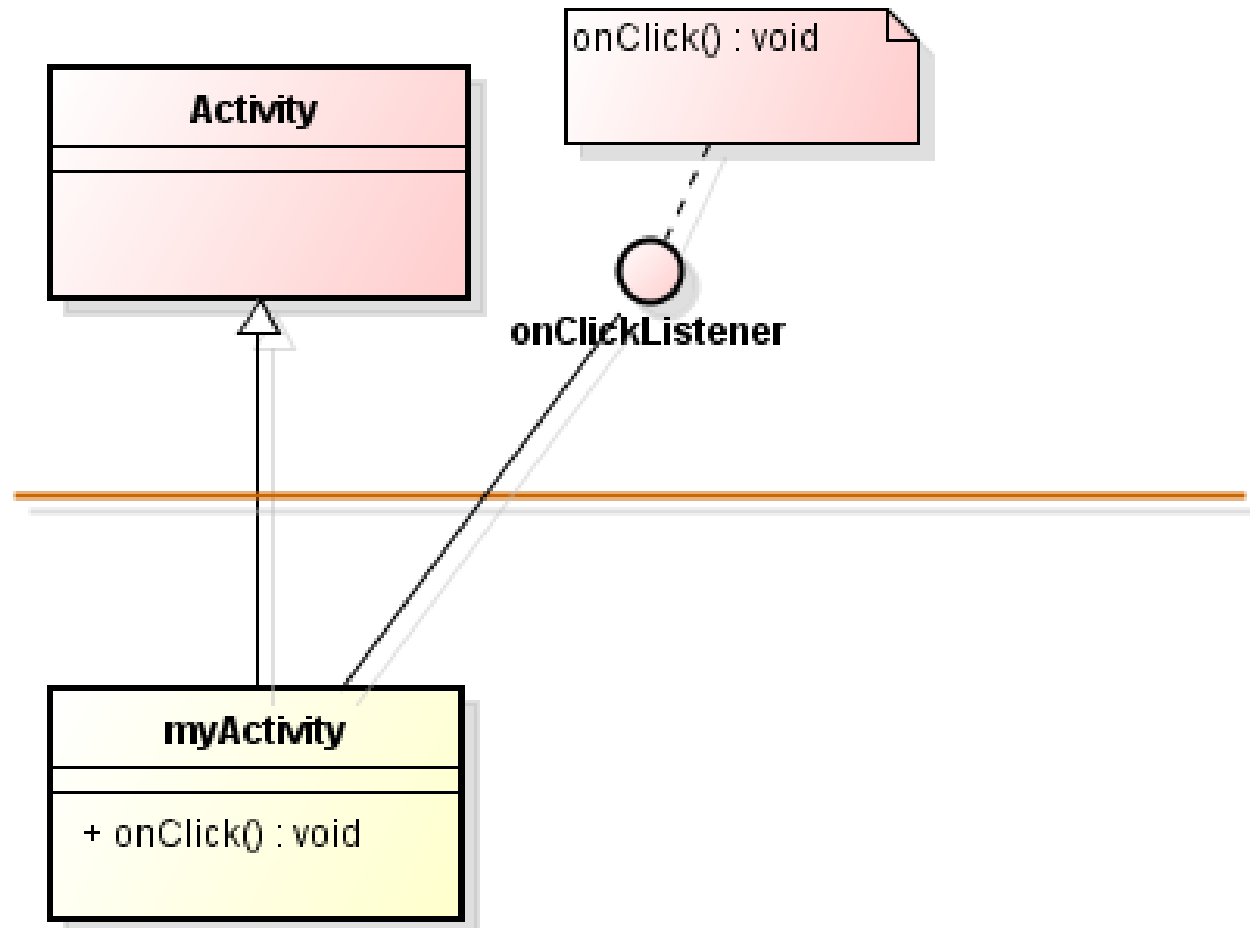
- ◎ 接着，可以在图形上表达类别与接口之间的关系。例如，选取<Association>图素，如下：



- ◎ 先點選這個圖素，從myActivity拉出一條線到OnClickListener接口，出現：



- ◎ 此关系说明了：myActivity类别「实现」了OnClickListener接口。其意味着，myActivity类别里含有一个实现函数：onClick()函数，如下图：



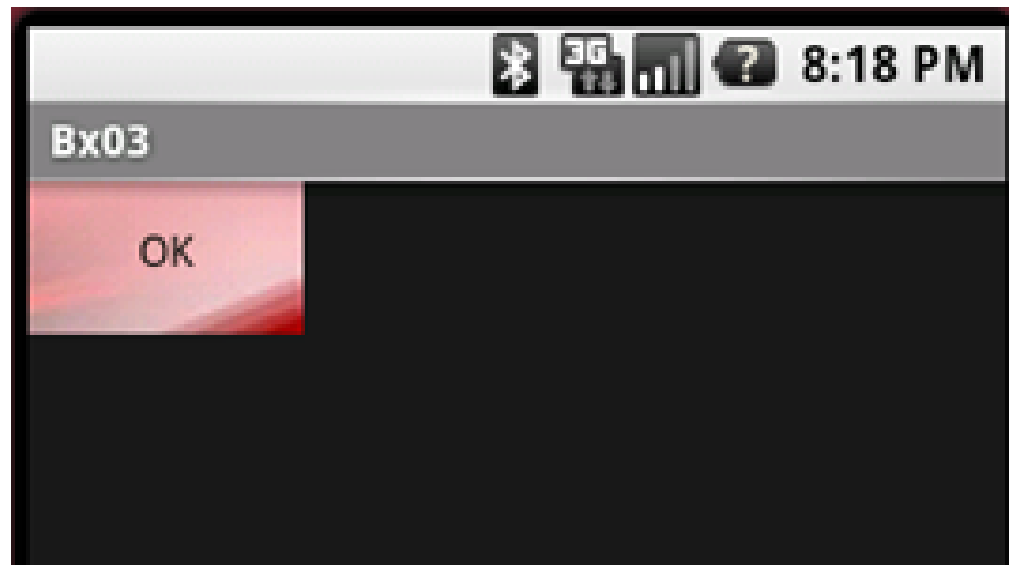
接口的使用(调用接口的函数)

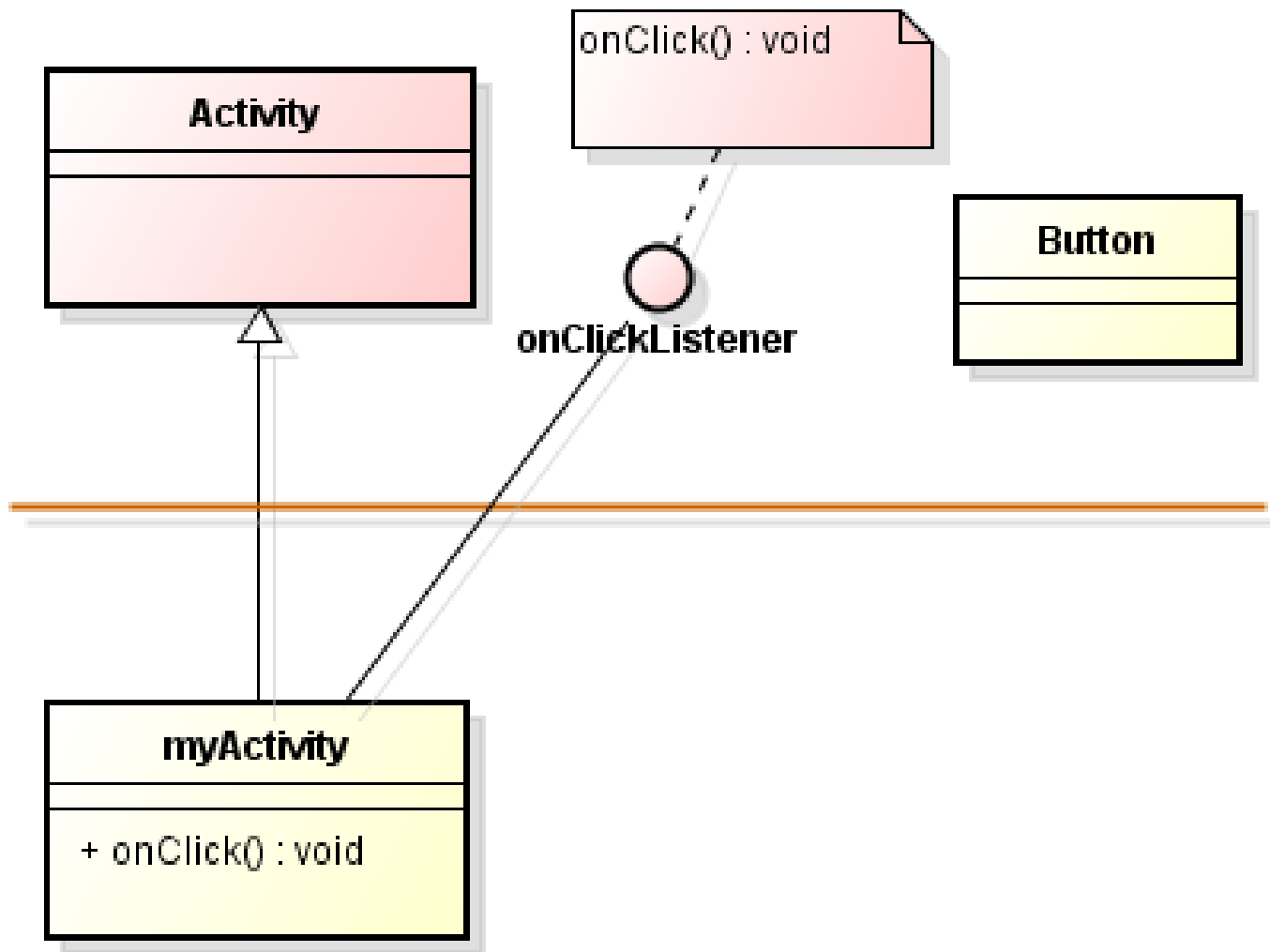
- ◎ myActivity实现了onClickListener接口之后，其它类别就能透过此接口来调用myActivity里的onClick()函数，

范例 (Android)

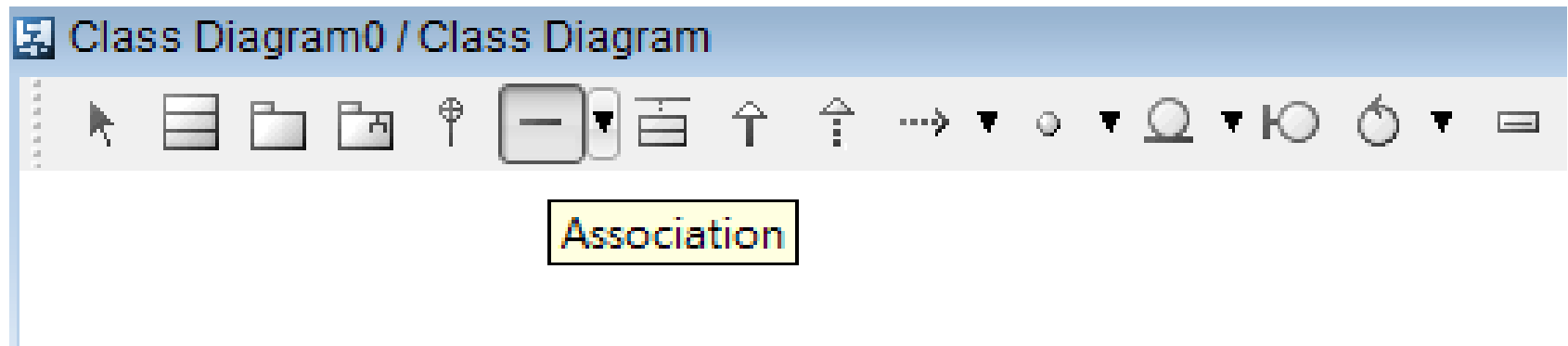
// myActivity.java

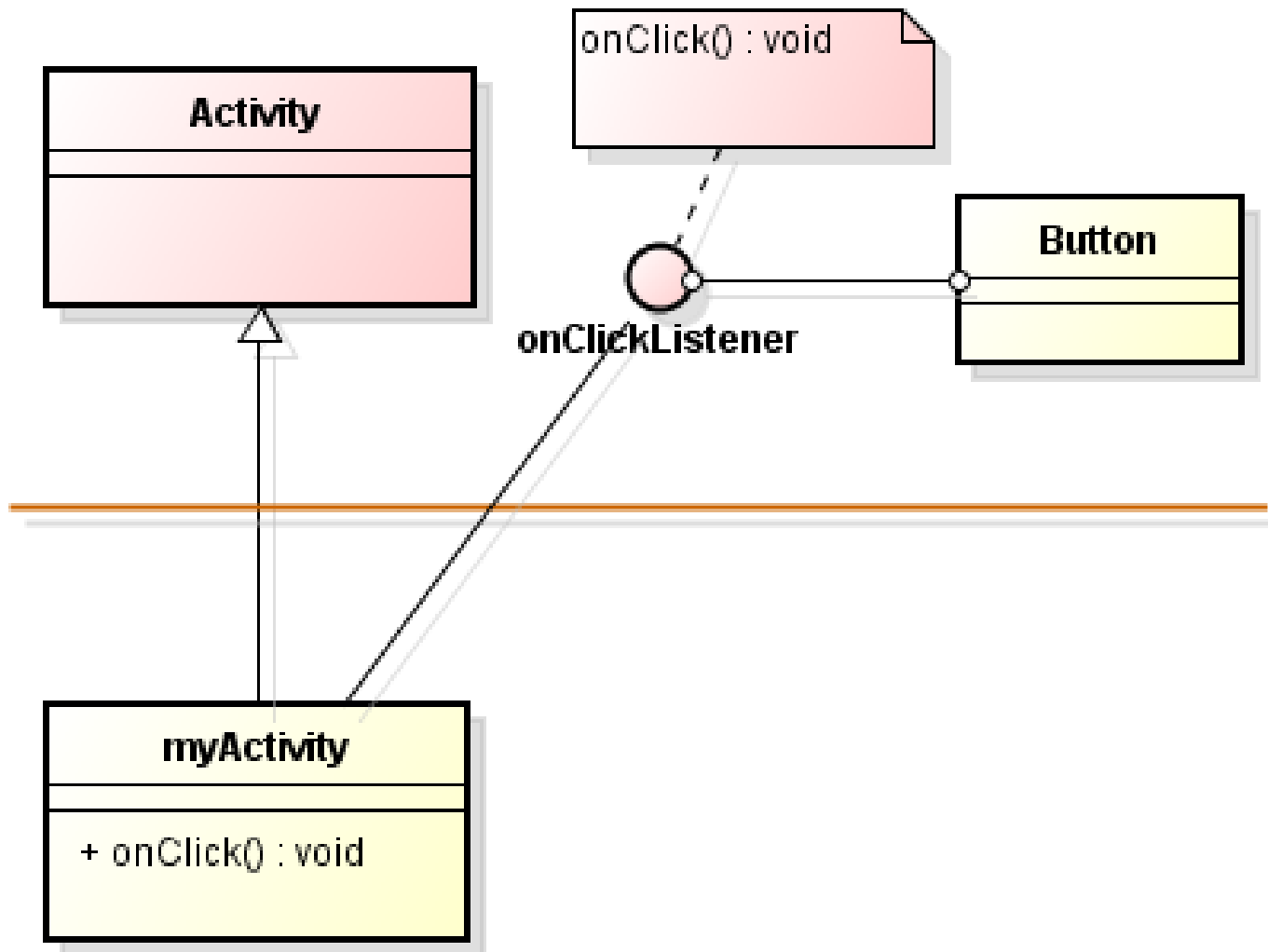
```
public class myActivity extends Activity {  
    @Override public void onCreate(Bundle icle) {  
        super.onCreate(icle);  
        setContentView(R.layout.main);  
        Button btn = new Button(this);  
        btn.setText("OK");  
        btn.setBackgroundResource(R.drawable.heart);  
        btn.setOnClickListener(this);  
        setContentView(button);  
    }  
}
```

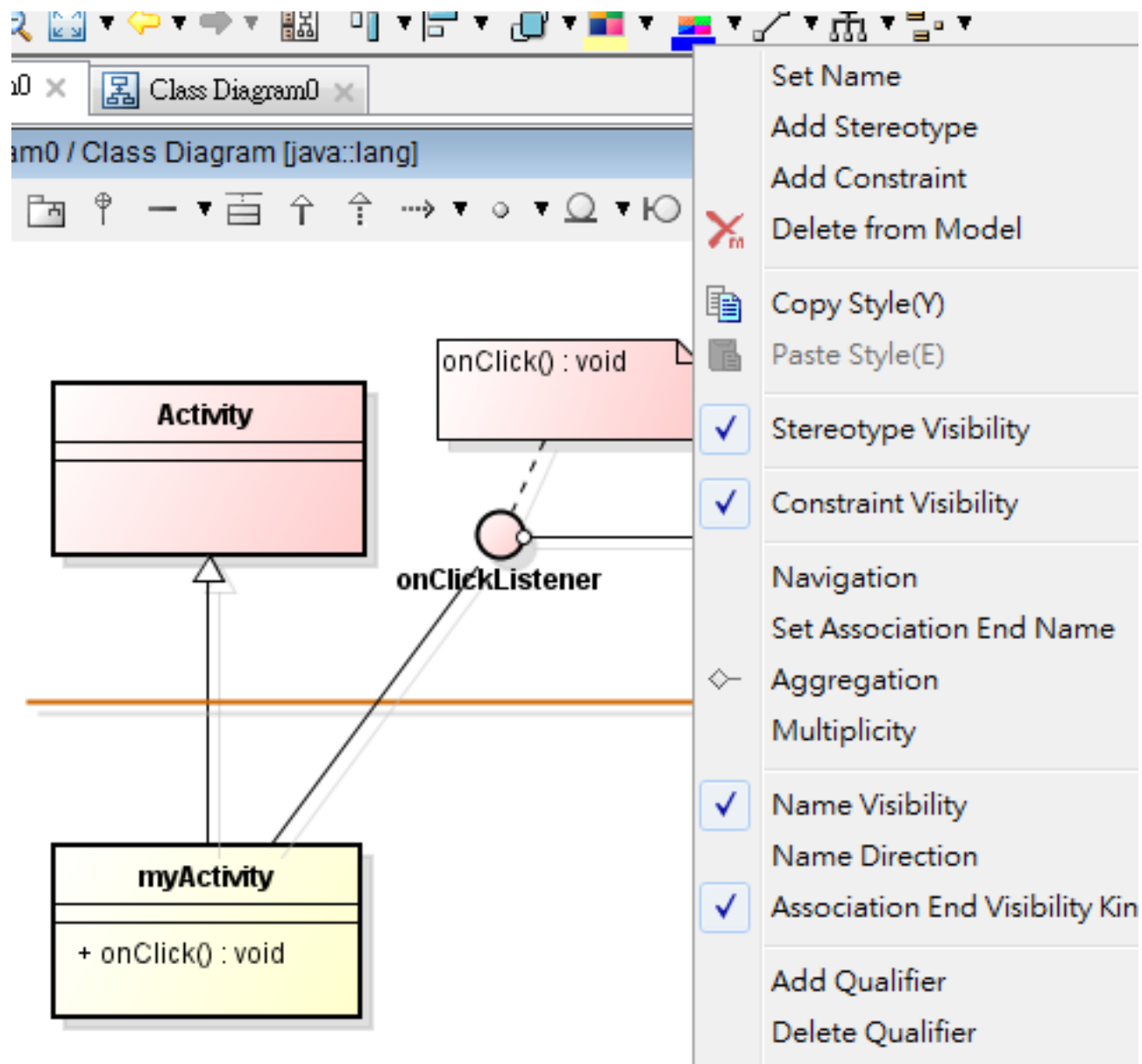


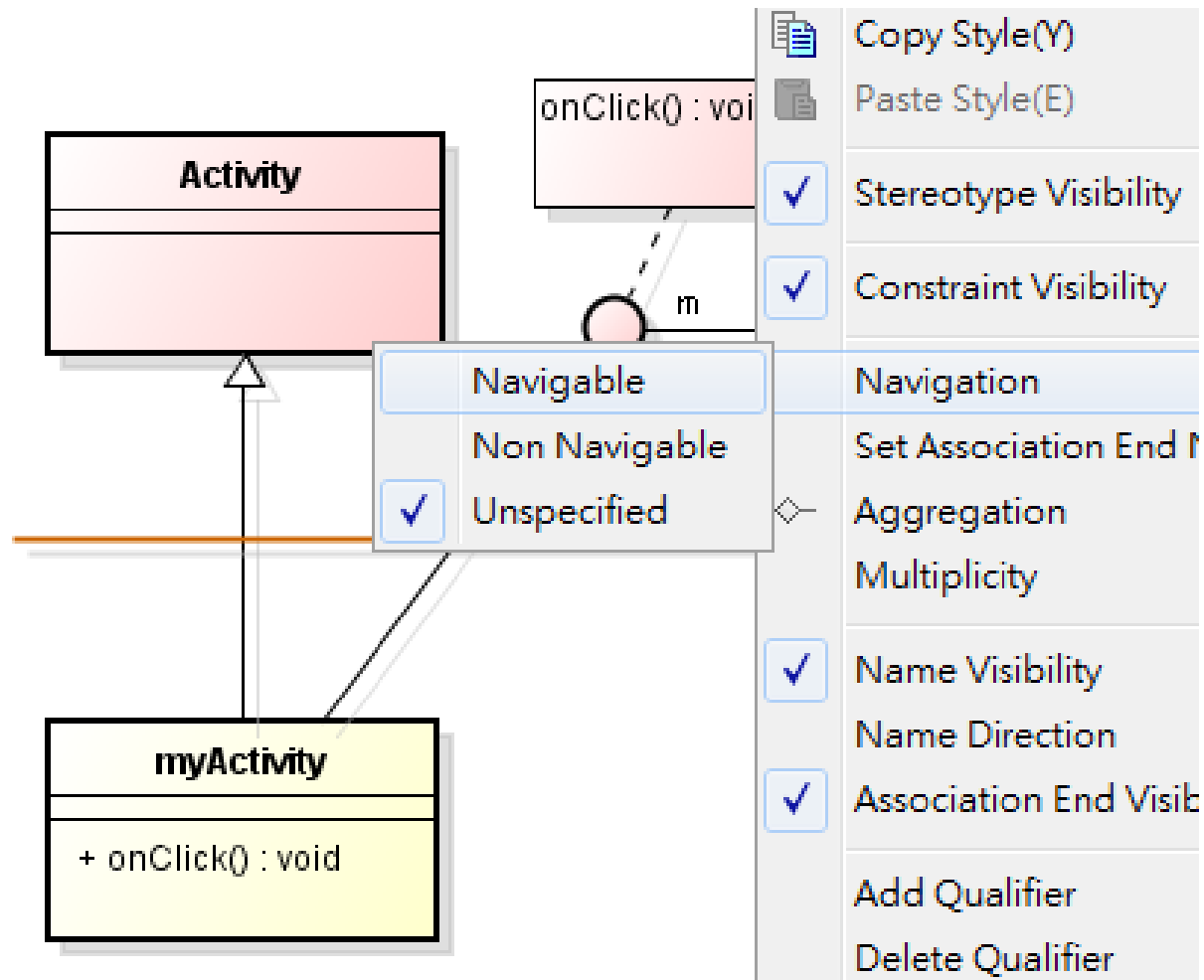


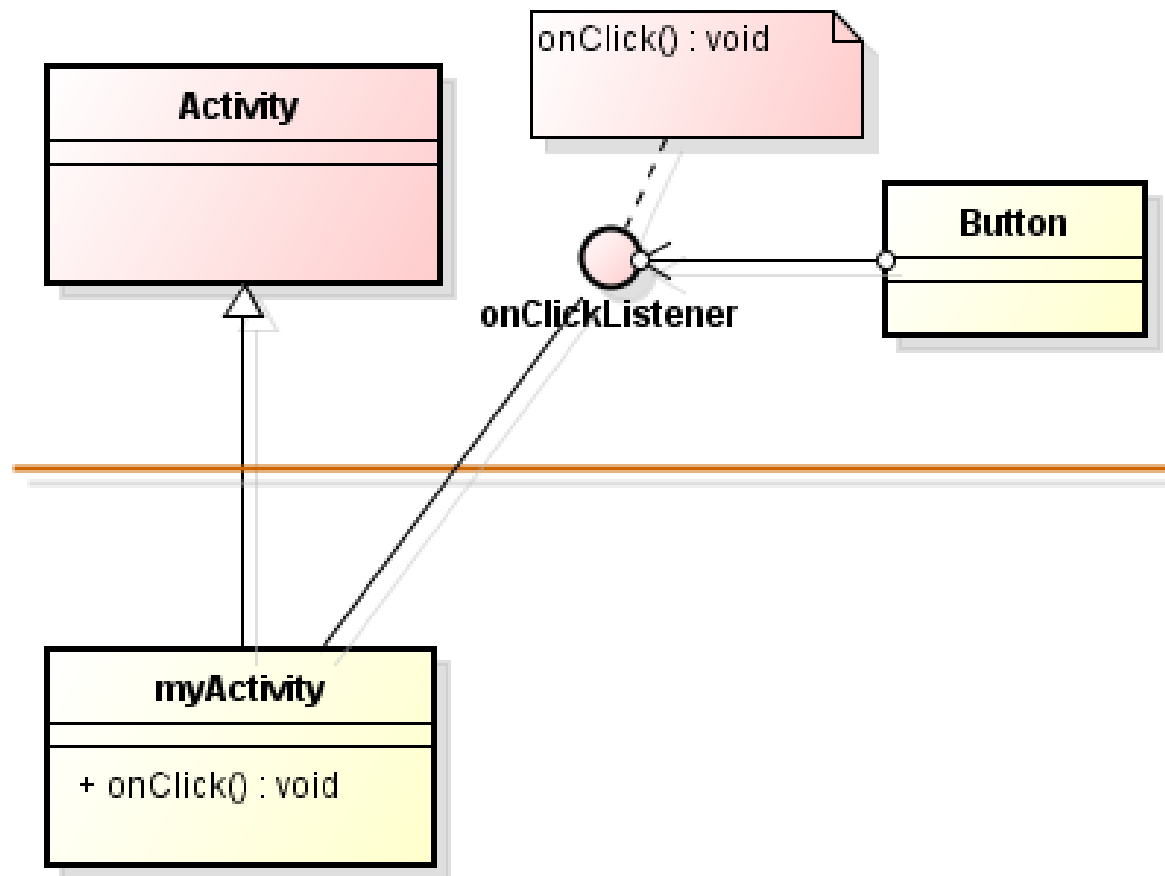
- ◎ 接着，可选取<Association>图素，如下：

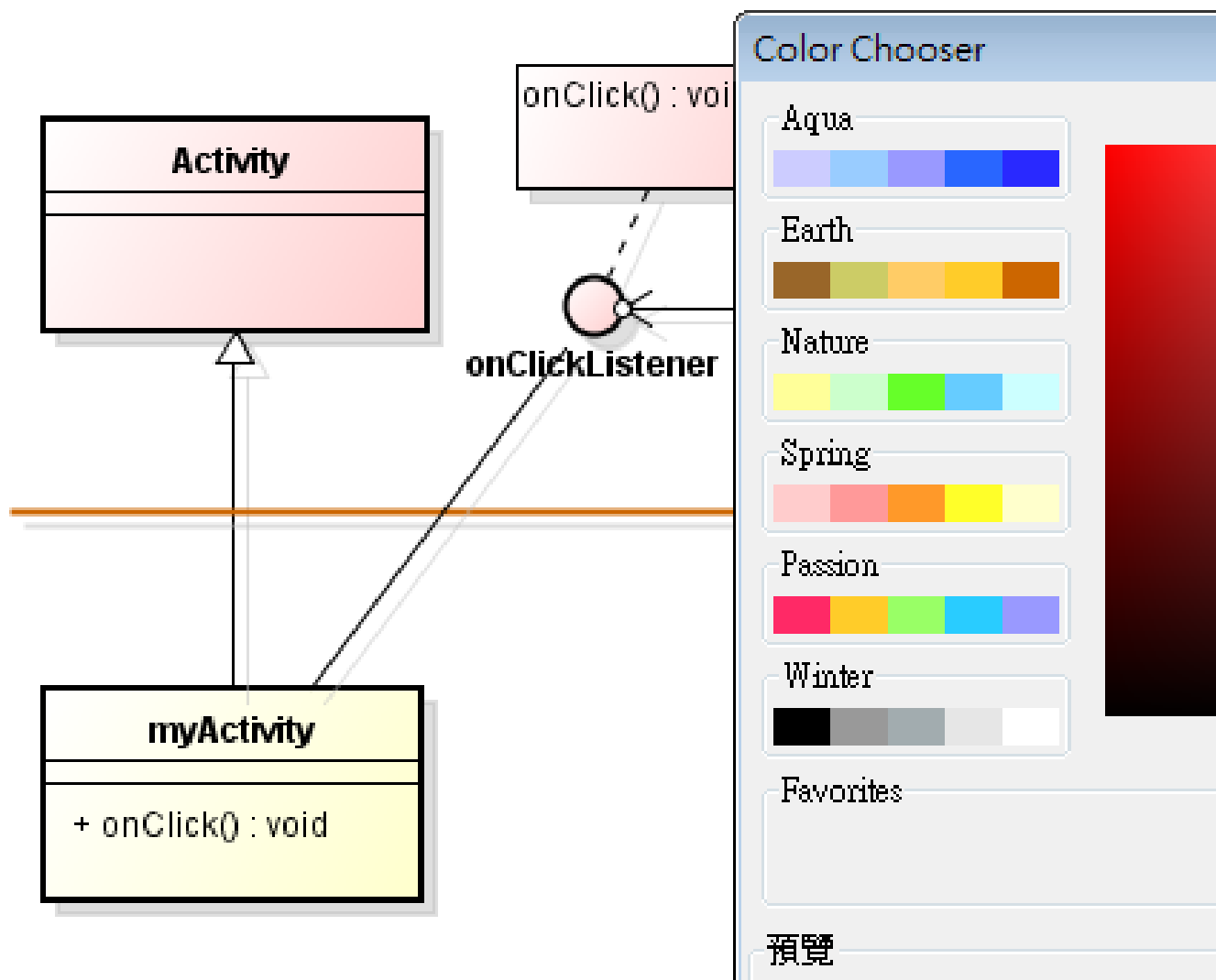


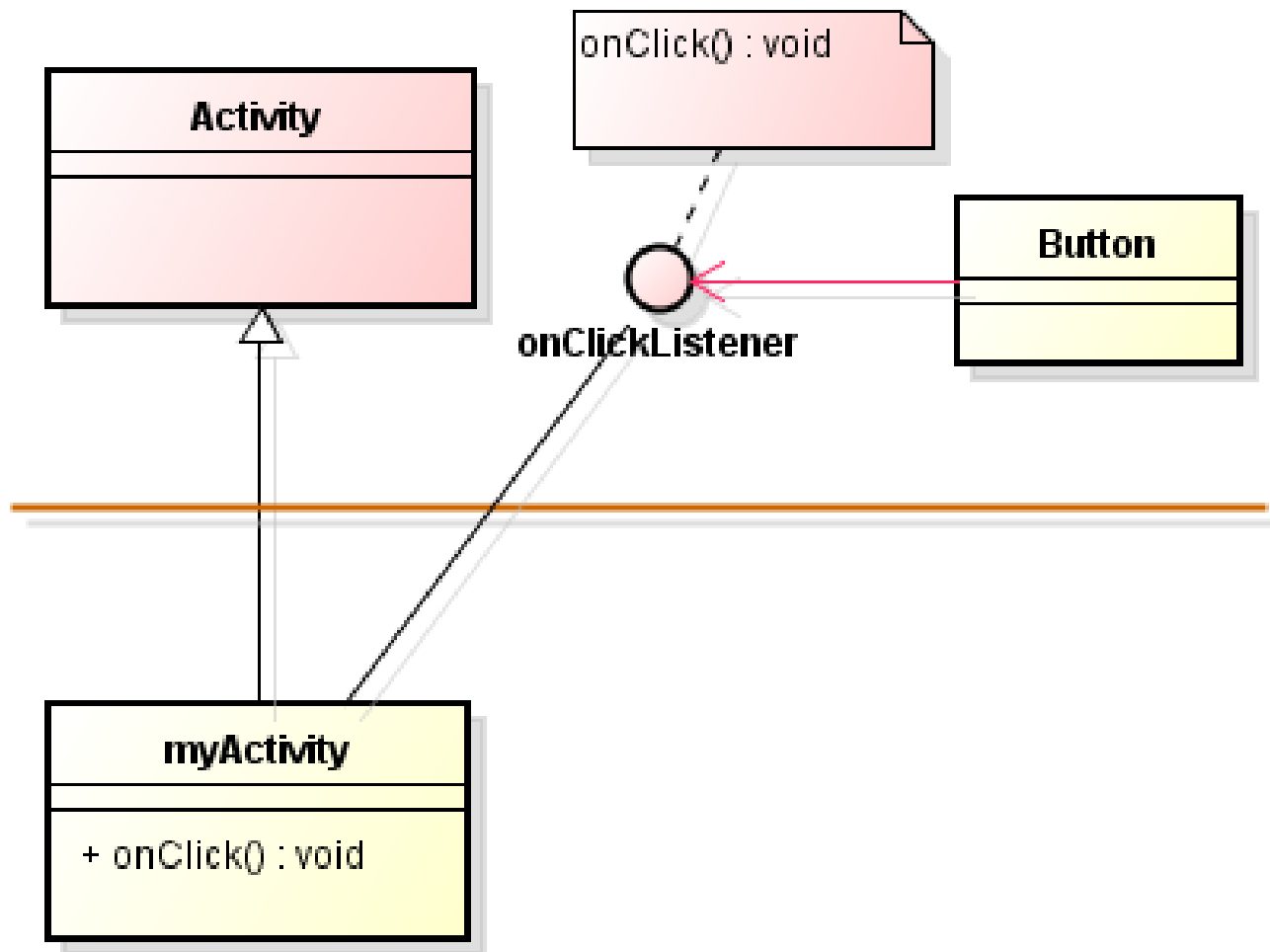


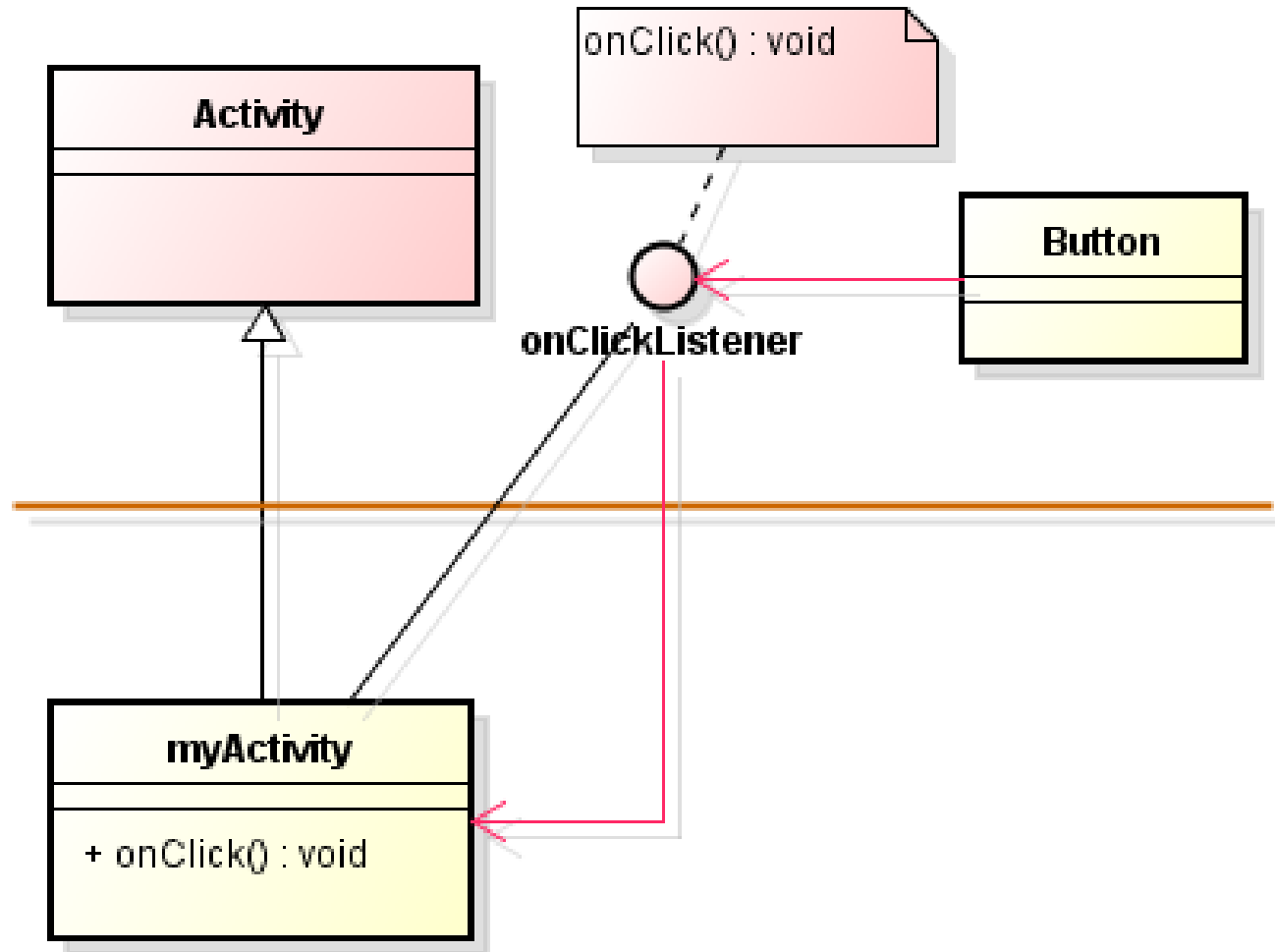


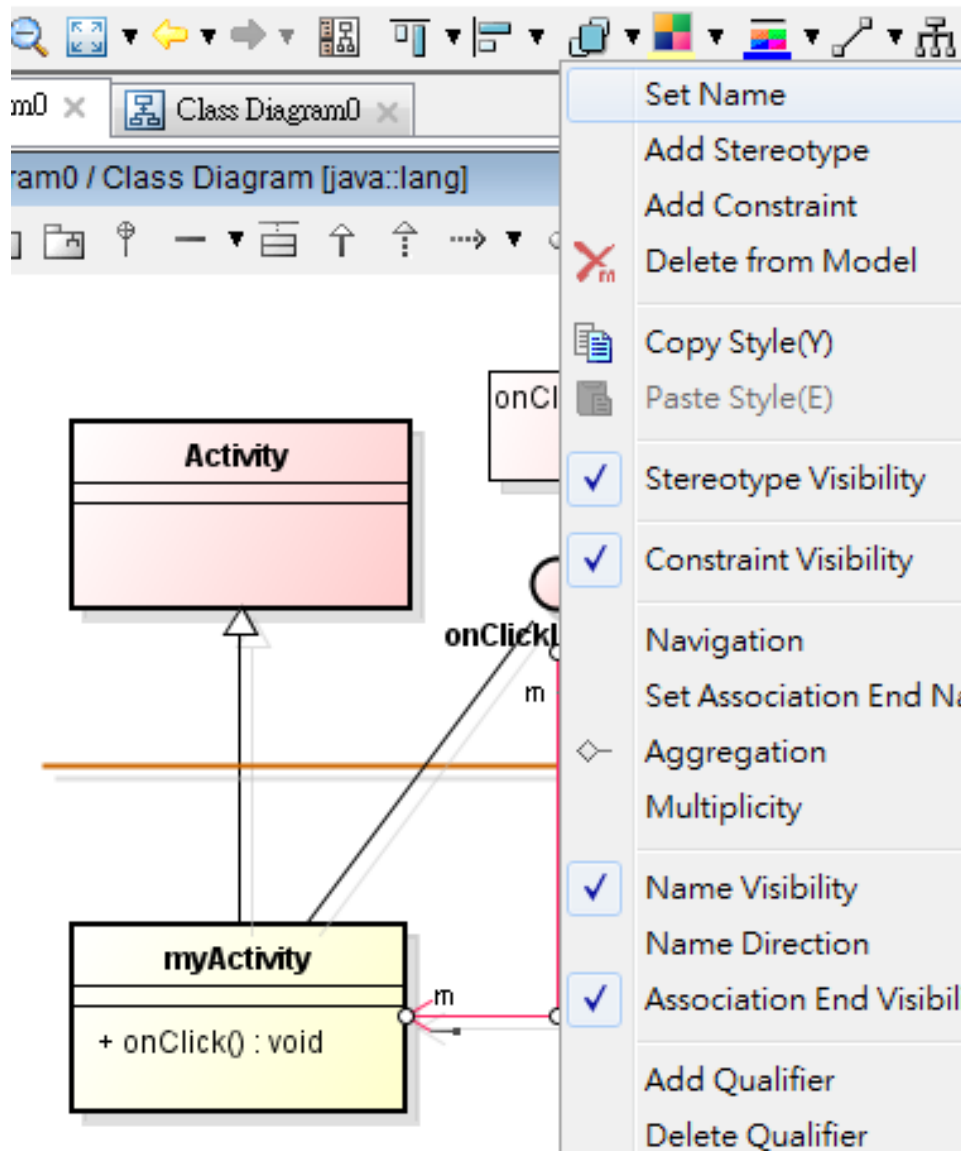


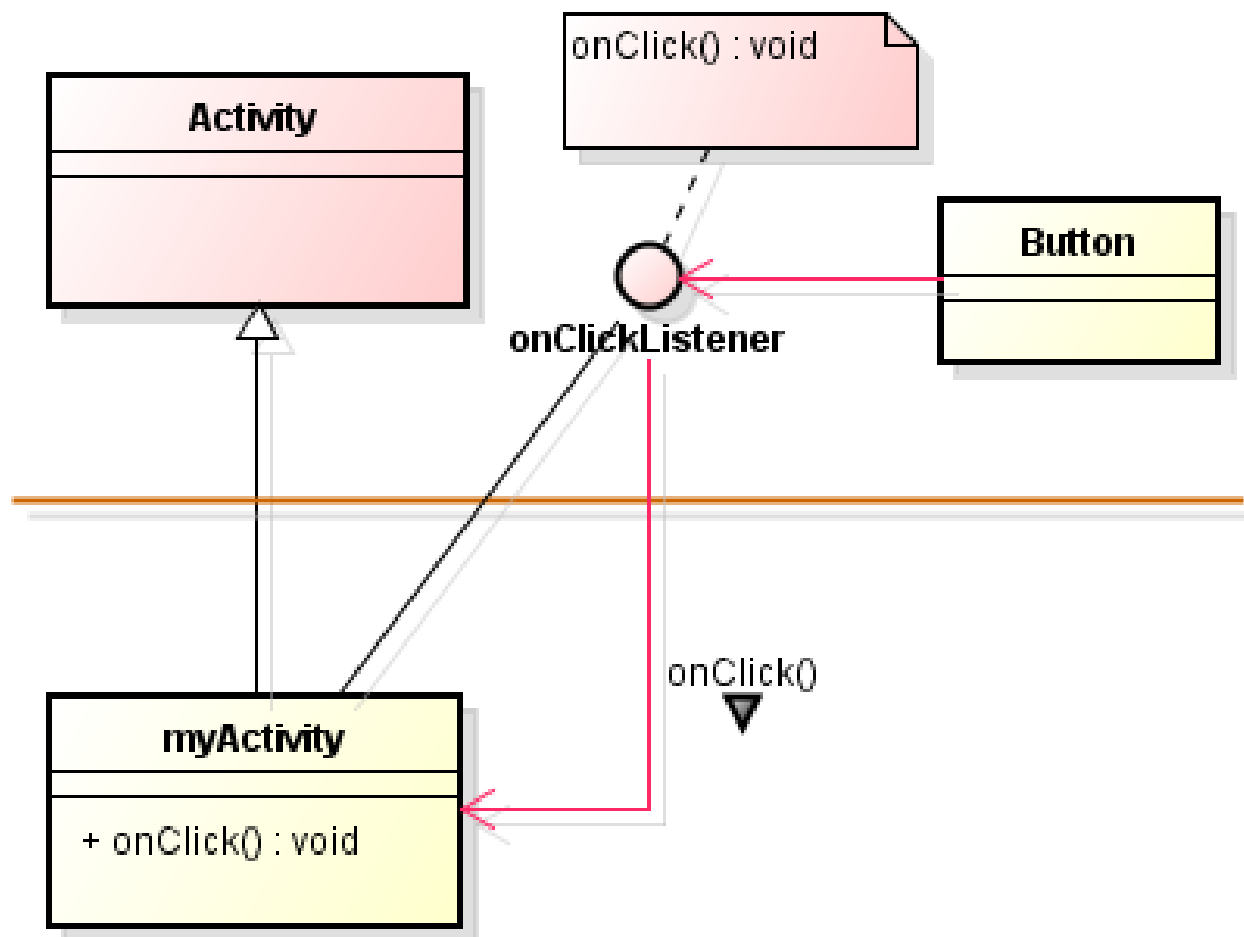




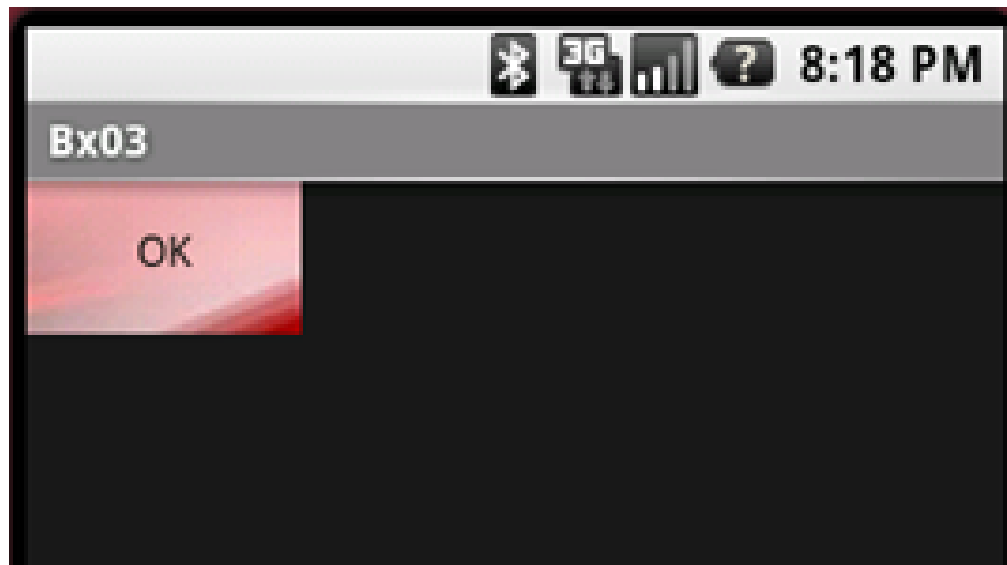








- ◎ 这表达出：当人们按下屏幕画面上的按钮时，Android框架(及其幕后的系统服务)会透过OnClickListener接口而调用了myActivity类别里的onClick()函数。



5、演练： UML的类别與接口

接口的表示

- ◎ 在OOP里，将接口定义为一种特殊的类别(Class)。
- ◎ 如果一个类别的某些函数是抽象函数的话，就称为「抽象函数」(Abstract Class)。如果一个抽象类别，它的所有函数全部都是抽象函数的话，就称为「纯粹抽象类别」(Pure Abstract Class)；这种类别又称为「接口」(Interface)。

◎ 在C++裡，類別包括3種：

1. 一般(具象)類別

-- 所有函數都是具象(內有指令)

2. 抽象(abstract)類別

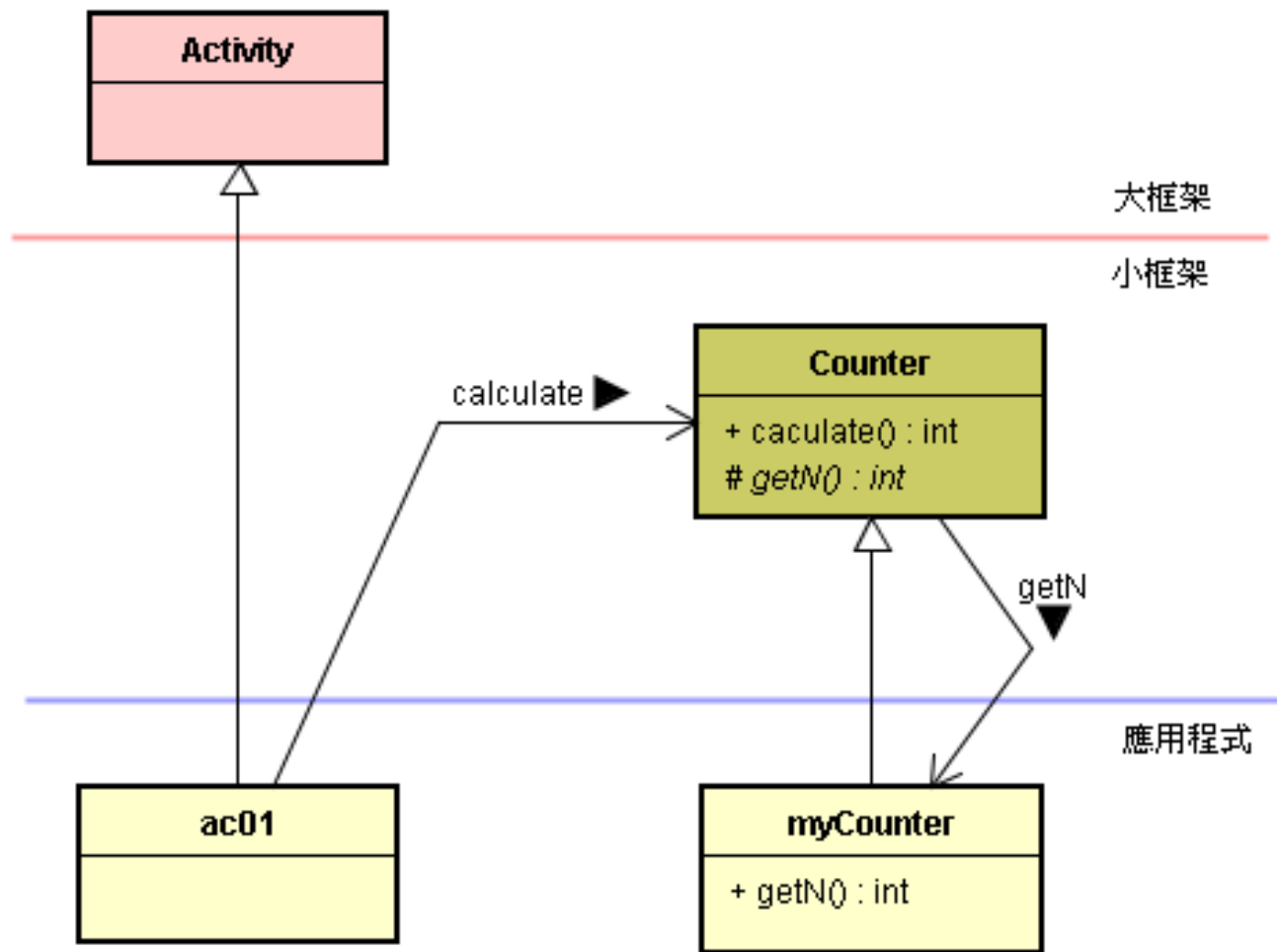
-- 有一個或多個函數是抽象的(內無指令)

3. 純粹抽象(pure abstract)類別

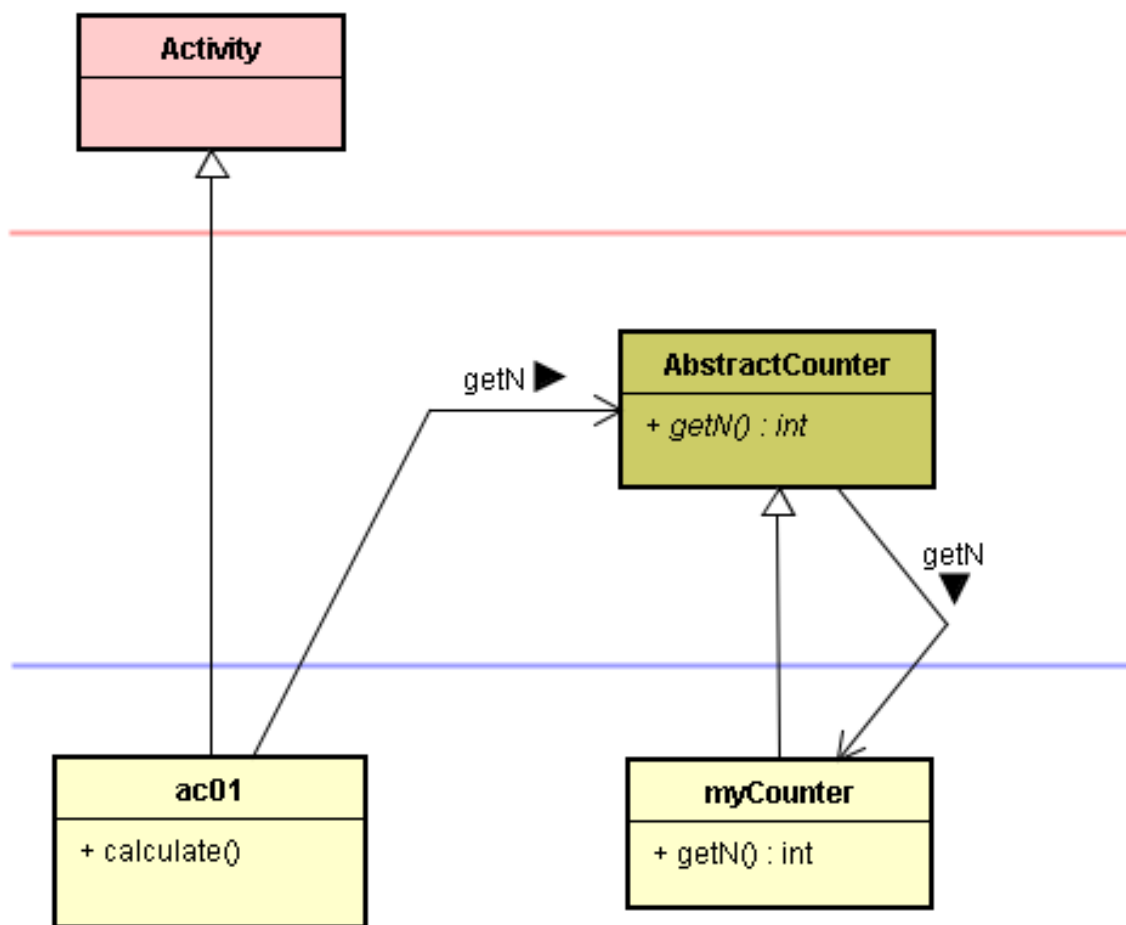
-- 所有函數都是抽象的

- ◎ 在Java里，将上述的「纯粹抽象类别」称为接口(Interface)
- ◎ 在UML里，以圆圈()来表示接口。

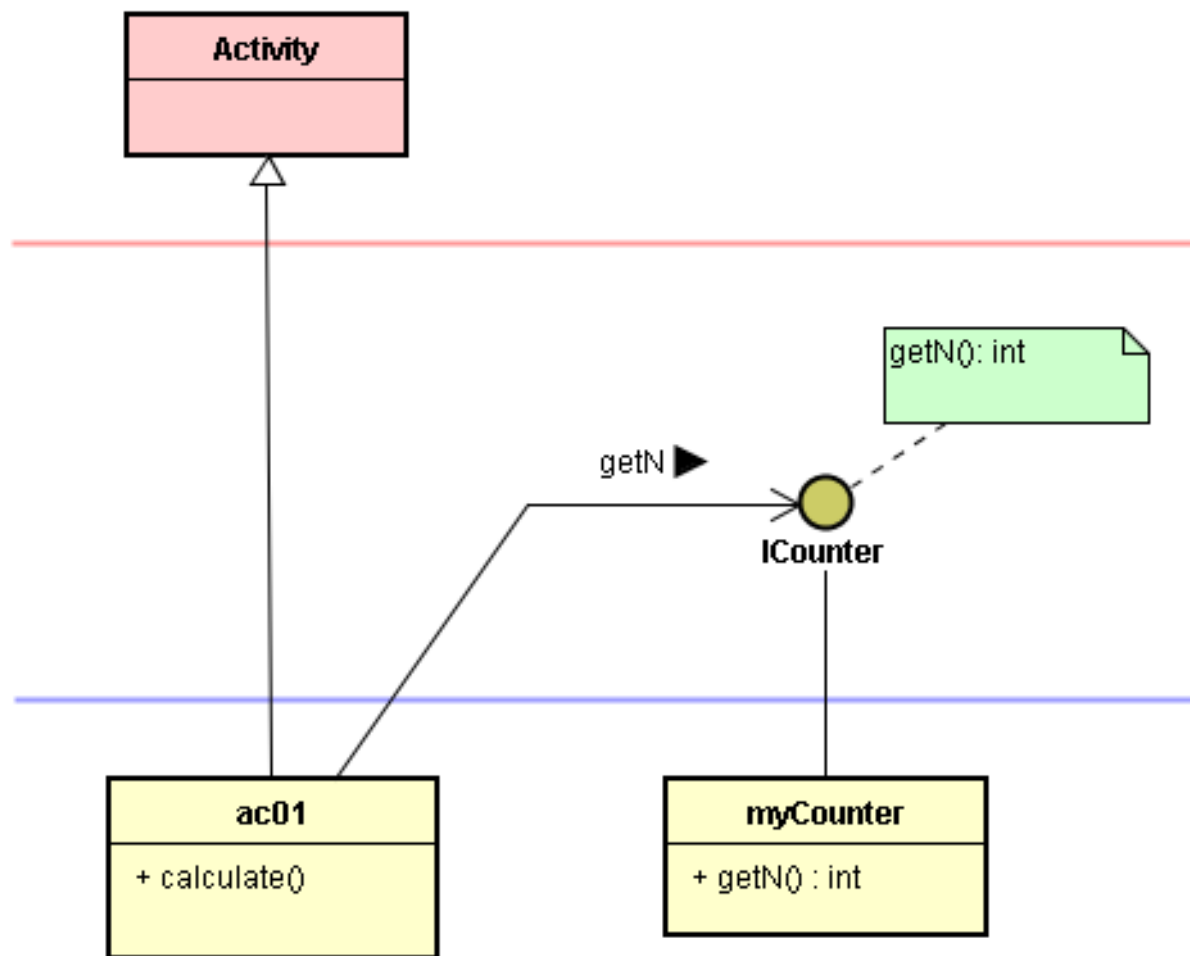
◎请问，下图的Counter是一个什么类别呢？



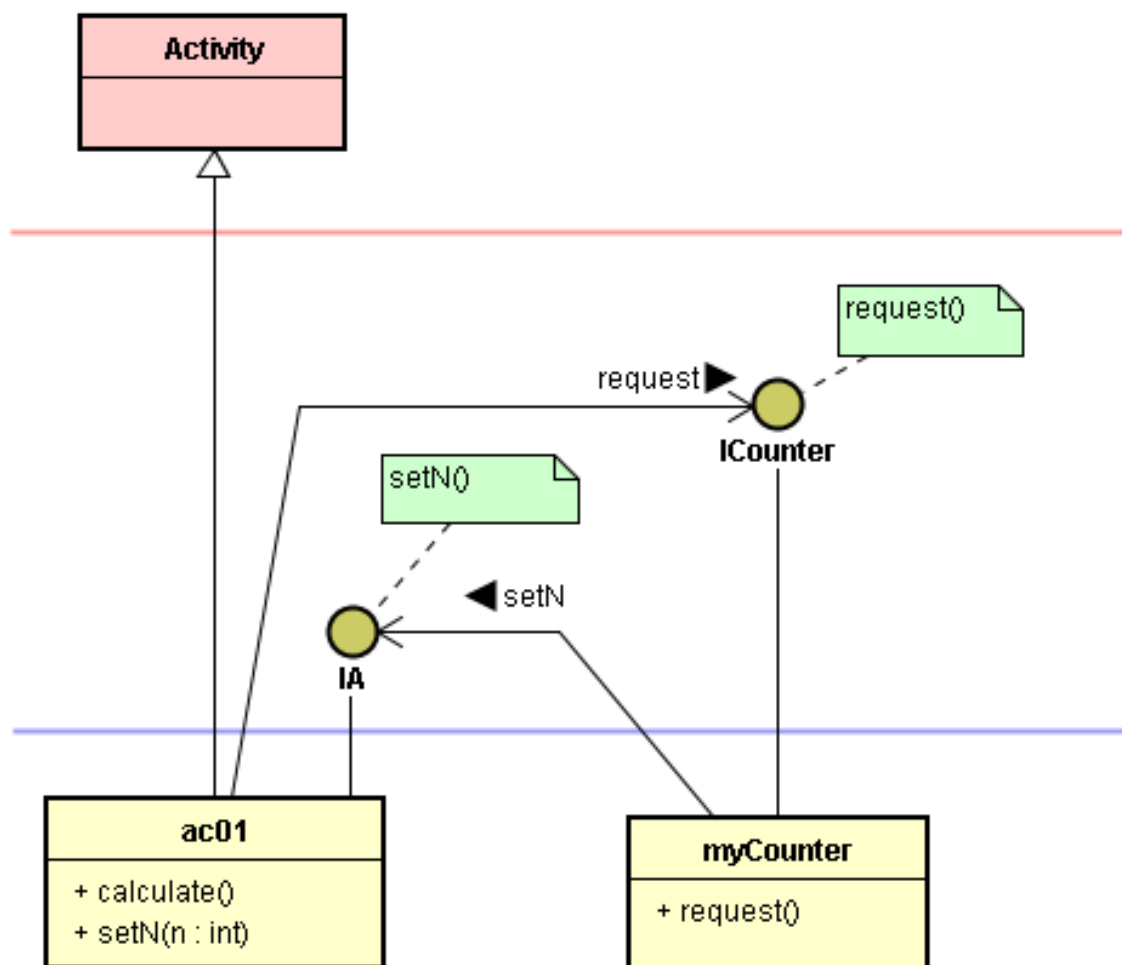
- ◎ 请问，下图里的Counter是一个什么类别呢？



◎ 上图相当于(也能表示为)下图，理由是什么？



◎请说明下图的涵意：



Q&A



高煥堂