

1228課程大綱

- 使用 Alert 交談窗
 - 一個簡單的交談窗範例
 - 問卷調查APP
- 日期、時間交談窗
 - 日期、時間選擇APP
- 程式作業

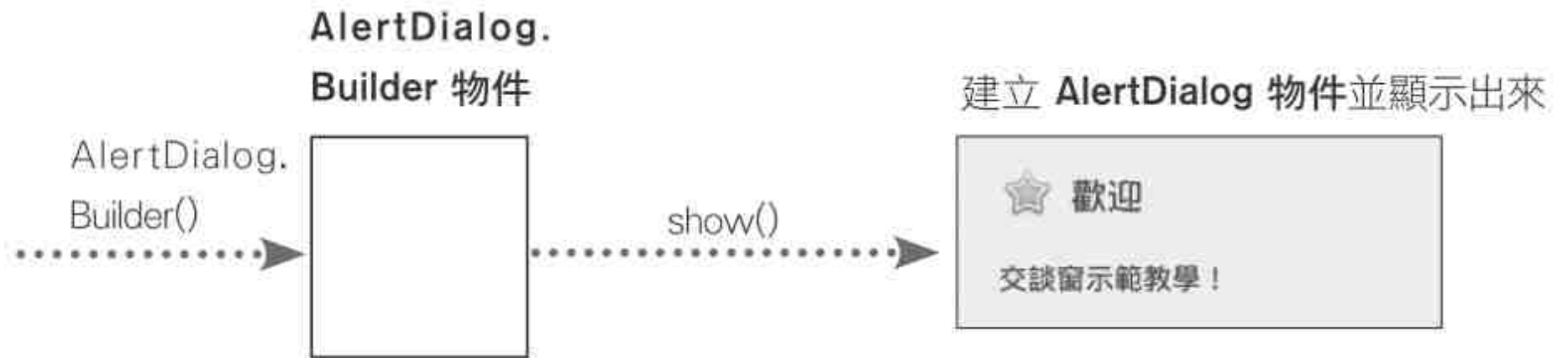
使用 Alert 交談窗(1)

- 要顯示 Alert 交談窗，必須先建立 AlertDialog.Builder 物件並顯示出來：

```
AlertDialog.Builder bdr = new AlertDialog.Builder(this);  
bdr.setMessage("交談窗示範教學！");  
bdr.setTitle("歡迎");  
bdr.setIcon(android.R.drawable.presence_away);
```



使用 Alert 交談窗(2)



使用 Alert 交談窗(3)

- 由於 AlertDialog 中的方法都會傳回 AlertDialog.Builder 物件自己，因此可以直接將多個方法用 “.” 串接起來。

```
AlertDialog.Builder bdr = new AlertDialog.Builder(this);  
bdr.setMessage("交談窗示範教學！")  
    .setTitle("歡迎");  
    .setIcon(android.R.drawable.presence_away)  
    .show();
```

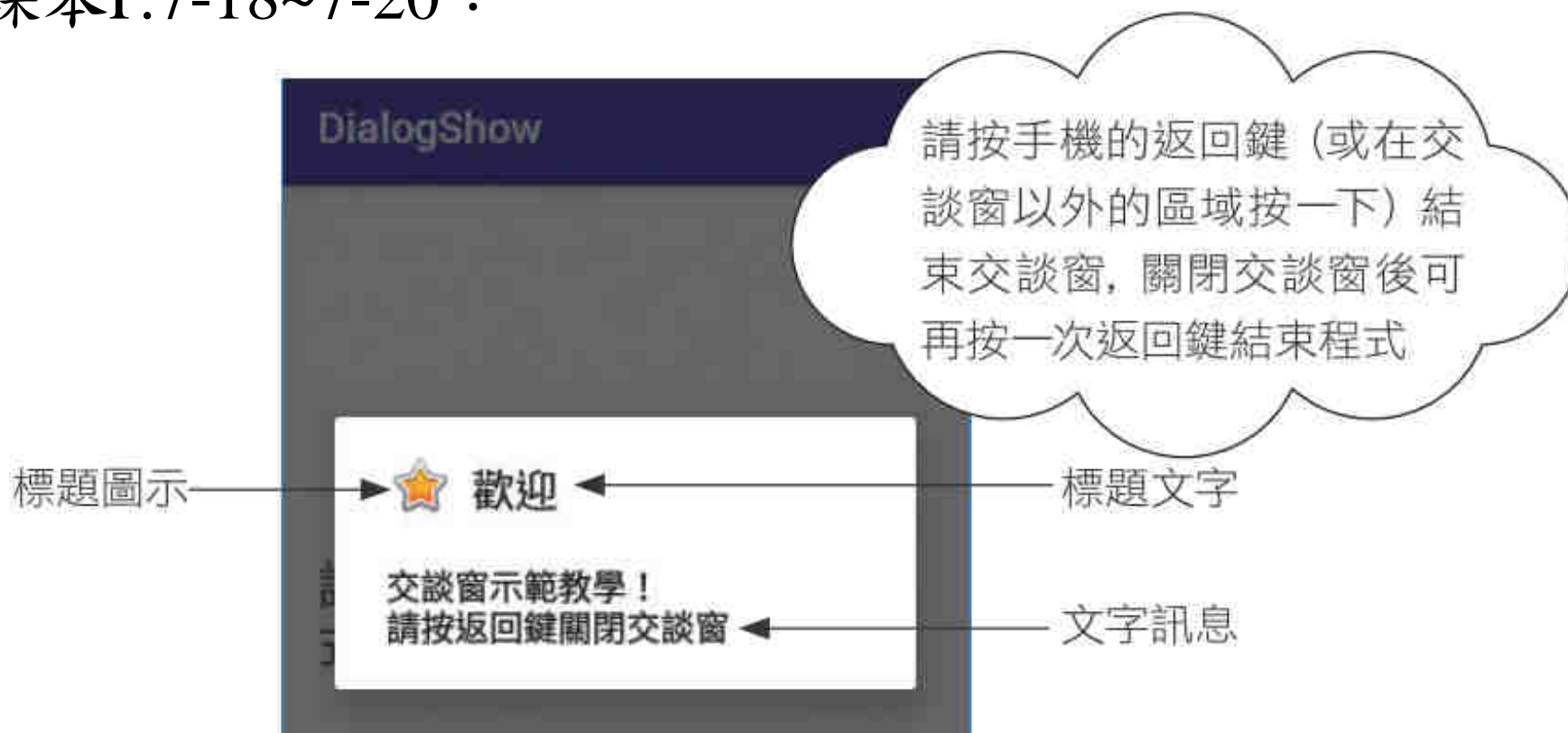
使用 Alert 交談窗(4)

- 甚至還可直接串接 new 運算的結果物件，省掉宣告變數的寫法，變成：

```
new AlertDialog.Builder(this)
    .setMessage("交談窗示範教學！")
    .setTitle("歡迎");
.setIcon(android.R.drawable.presence_away)
.show();
```

一個簡單的交談窗範例(1)

請參考課本P.7-18~7-20：



一個簡單的交談窗範例(2)

```
01 protected void onCreate(Bundle savedInstanceState) {  
02     super.onCreate(savedInstanceState);  
03     setContentView(R.layout.activity_main);  
04  
05     AlertDialog.Builder bdr = new AlertDialog.Builder(this);  
06     bdr.setMessage("交談窗示範教學！\n"    ← 加入文字訊息  
07         + "請按返回鍵關閉交談窗");  
08     bdr.setTitle("歡迎");    ← 加入標題  
09     bdr.setIcon(android.R.drawable.btn_star_big_on); ← 加入圖示  
10     bdr.setCancelable(true); ← 允許按返回鍵關閉交談窗  
11     bdr.show();  
12 }
```

使用 Alert 交談窗(5)

- 在Alert交談窗中最多可以加入3個按鈕，分別代表是(Positive)，否(Negative)，中性(Neutral)：

```
new AlertDialog.Builder(this)
    .setMessage("你好嗎？")
    .setNegativeButton("不好", null)
    .setPositiveButton("很好", null);
```



使用 Alert 交談窗(6)

- 第1個參數為按鈕中要顯示的字串，第二個參數必須傳入實作 `DialogInterface.OnClickListener` 介面的物件，以作為按下按鈕時的 `onClick` 監聽器。
 - 若設為 `null` 表示不處理這個按鈕事件。

使用 Alert 交談窗(7)

- 要讓 MainActivity 監聽交談窗中按鈕的事件，就要先讓 MainActivity 實作 DialogInterface.OnClickListener 介面：

```
public class MainActivity extends AppCompatActivity  
    implements DialogInterface.OnClickListener {
```

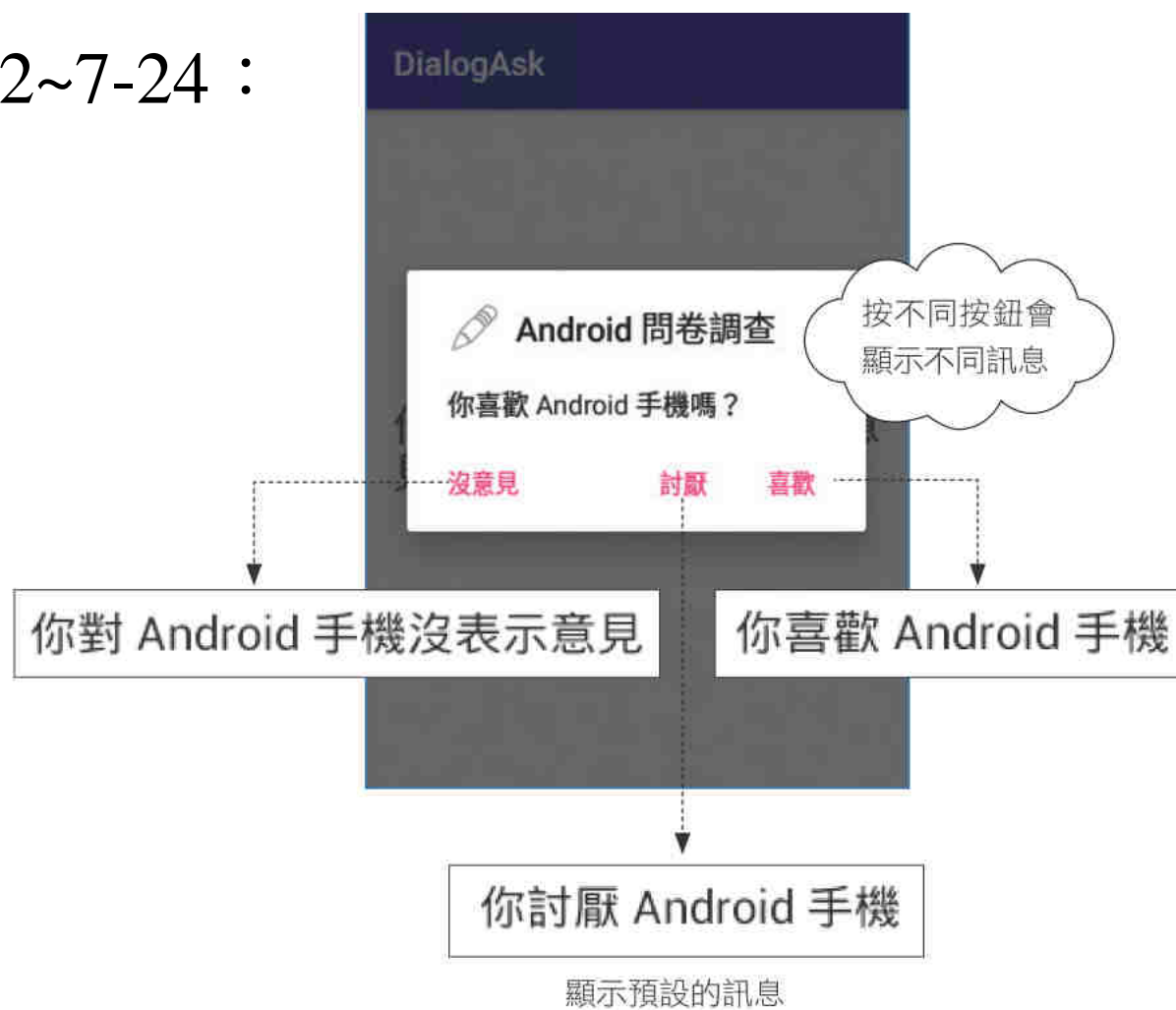
使用 Alert 交談窗(8)

- DialogInterface.OnClickListener 介面中只定義有一個 onClick() 方法：

```
public void onClick(DialogInterface dialog, int which) {  
    if (which == DialogInterface.BUTTON_POSITIVE) {  
        ... // 按下『是』按鈕時的處理  
    }  
    else if (which == DialogInterface.BUTTON_NEGATIVE) {  
        ... // 按下『否』按鈕時的處理  
    }  
}
```

問卷調查APP(1)

請參考課本P.7-22~7-24：



問卷調查APP(2)

```
01 package tw.com.flag.ch07_dialogask;
02
03 import android.os.Bundle;
04 import android.app.Activity;
05 import android.app.AlertDialog;
06 import android.content.DialogInterface;
07 import android.view.Menu;
08 import android.widget.TextView;
09
10 protected class MainActivity extends AppCompatActivity
11     implements DialogInterface.OnClickListener { ← 實作監聽介面
12
13     TextView txv; ← 記錄預設的 TextView 元件
14
15     @Override
16     public void onCreate(Bundle savedInstanceState) {
```

問卷調查APP(3)

```
17      super.onCreate(savedInstanceState);
18      setContentView(R.layout.activity_main);
19
20      txv = (TextView)findViewById(R.id.answer); ← 找出預設的 TextView
21      new AlertDialog.Builder(this) ← 建立 Builder 物件
22          .setMessage("你喜歡 Android 手機嗎?") ← 設定顯示訊息
23          .setCancelable(false) ← 禁用返回鍵關閉交談窗
24          .setIcon(android.R.drawable.ic_menu_edit) ← 採用內建的圖示
25          .setTitle("Android 問卷調查") ← 設定交談窗的標題
26          .setPositiveButton("喜歡", this) ← 加入肯定按鈕並監聽事件
27          .setNegativeButton("討厭", this) ← 加入否定按鈕並監聽事件
28          .setNeutralButton("沒意見", null) ← 不監聽中性按鈕
29          .show(); ← 顯示交談窗
30    }
31
32    @Override
```

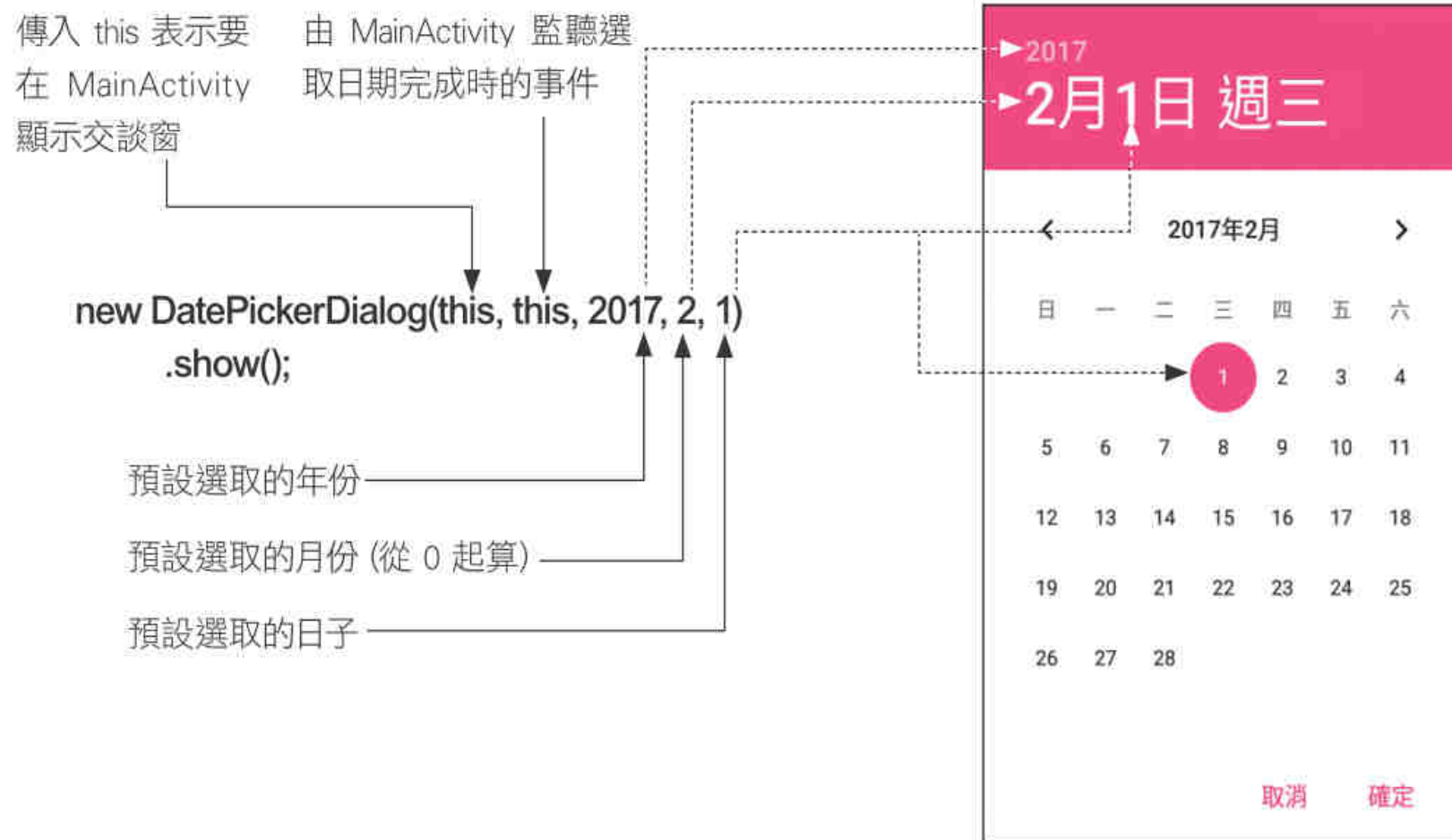
問卷調查APP(4)

```
33     public void onClick(DialogInterface dialog, int which) { ← 實作監聽介面定義的方法
34         if(which == DialogInterface.BUTTON_POSITIVE) { ← 如果按下肯定的『喜歡』
35             txv.setText("你喜歡 Android 手機");
36         } ← 如果按下否定的『討厭』
37         else if(which == DialogInterface.BUTTON_NEGATIVE) { ←
38             txv.setText("你討厭 Android 手機");
39         }
40     }
```

日期、時間交談窗(1)

- 日期、時間交談窗可以讓使用者以選取的方式輸入日期及時間，以確保輸入資料的正確性。
 - DatePickerDialog及TimePickerDialog 類別可用來建立日期、時間交談窗。
 - 再來show()方法顯示出來。

日期、時間交談窗(2)



日期、時間交談窗(3)

傳入 this 表示要在
MainActivity 顯示交談窗

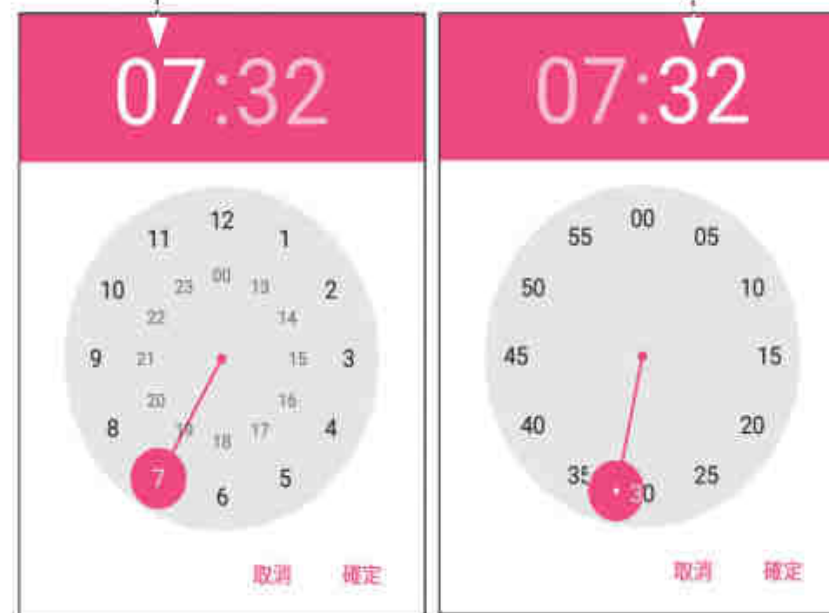
由 MainActivity 監聽選
取時間完成時的事件

```
new TimePickerDialog(this, this, 7, 32, true)  
.show();
```

預設選取的時數

預設選取的分鐘數

使用 24 小時制

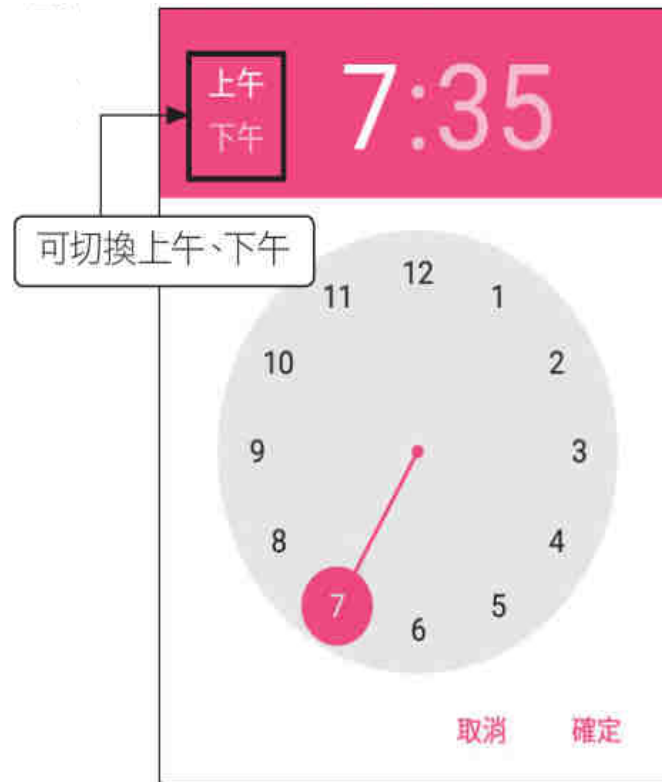


設定時 (幾點)

設定分

日期、時間交談窗(4)

- 設定時間時，若第三個參數設為false：



日期、時間交談窗(5)

- 如果是DatePickerDialog，監聽物件必須實作DatePickerDialog.OnDateSetListener介面定義的onDateSet()方法：

```
public void onDateSet(DatePicker v, int y, int mon, int d) {  
    ...  
}
```

日期、時間交談窗(6)

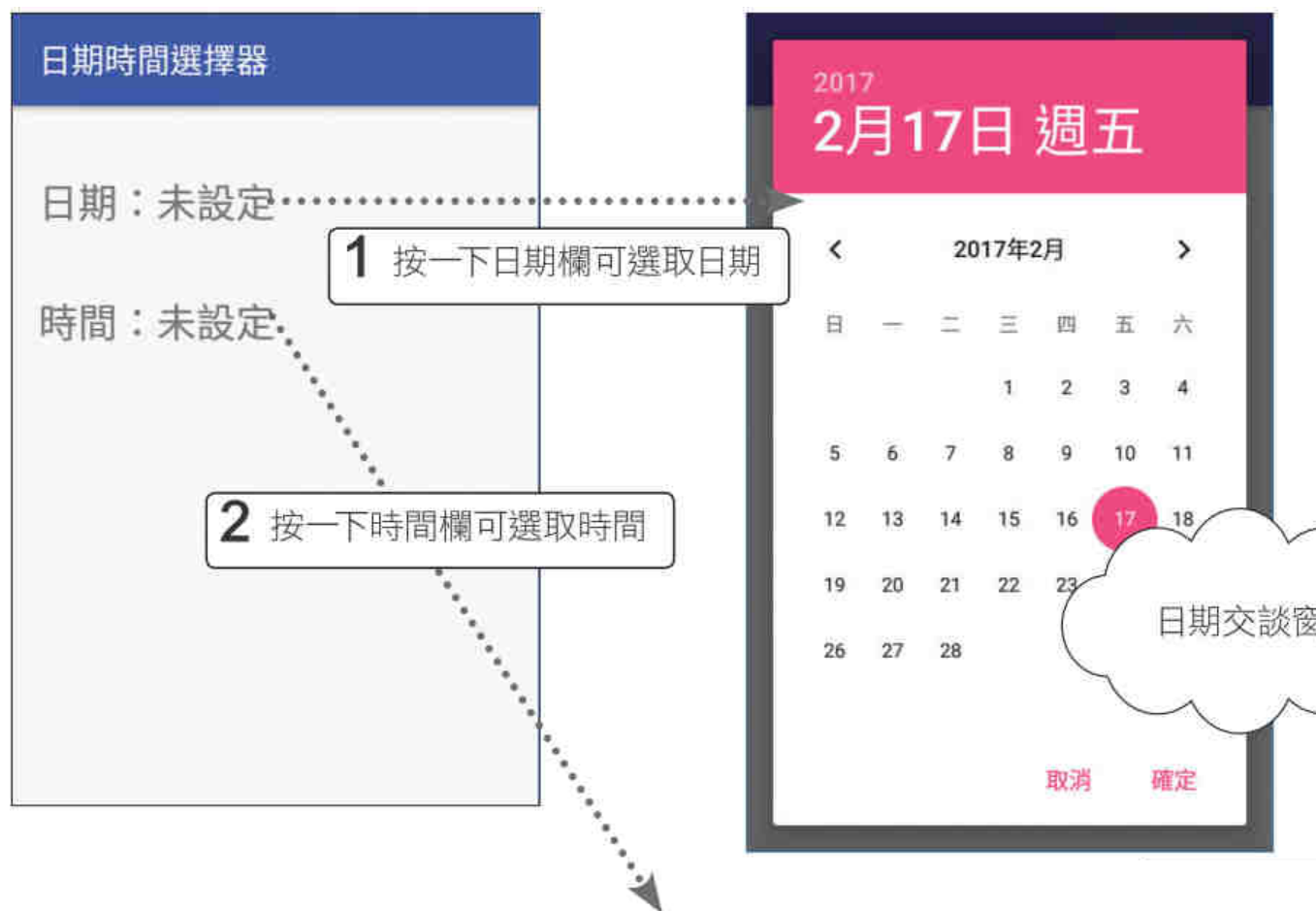
- 如果是TimePickerDialog，監聽物件必須實作TimePickerDialog.OnTimeSetListener介面定義的onTimeSet()方法：

```
public void onTimeSet(TimePicker v, int h, int m) {  
    ...  
}
```

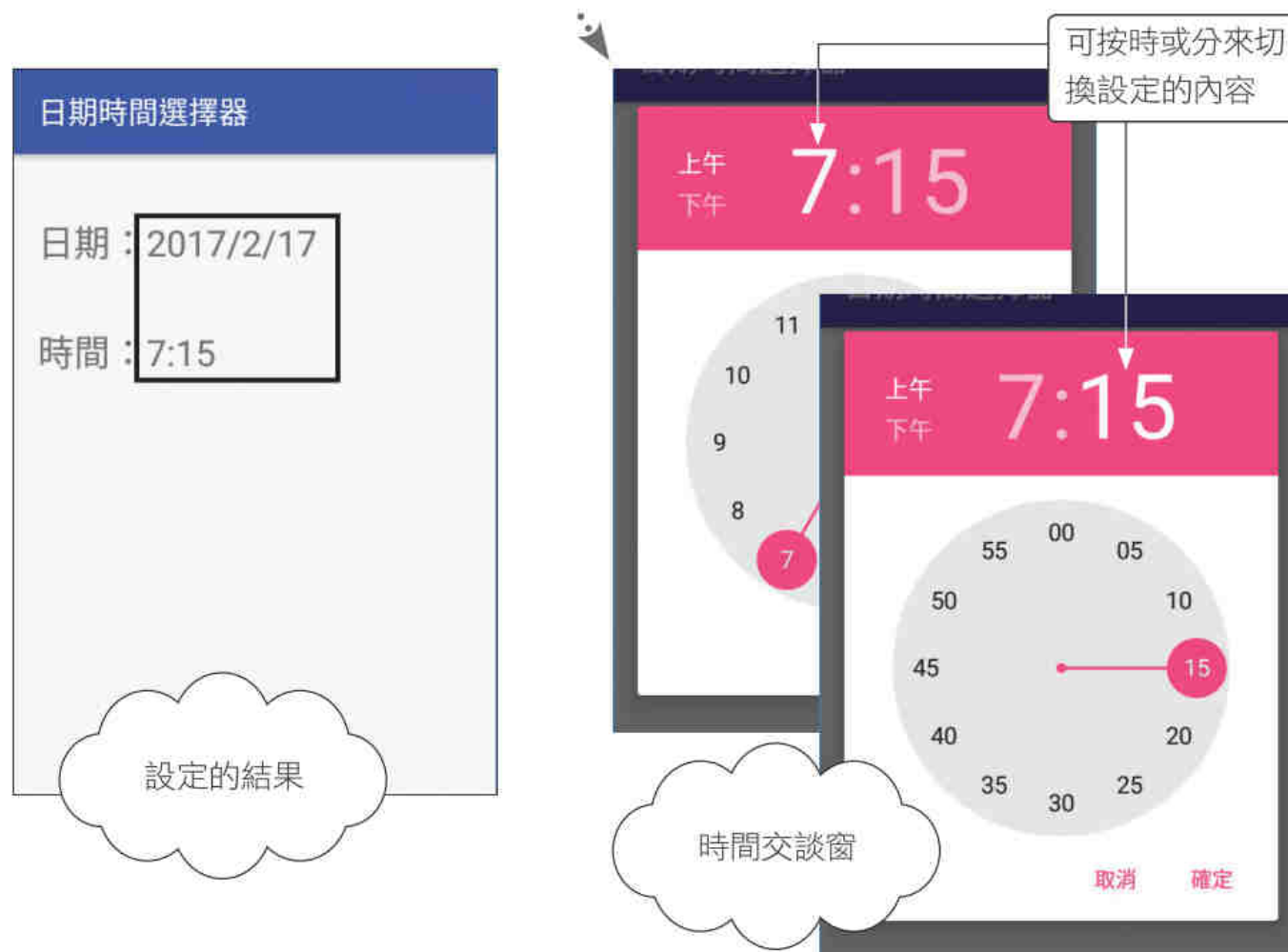
↑
參數 h 與 m 分別是
使用者選取的時與分

日期、時間選擇APP(1)

請參考課本P.7-28~7-33：



日期、時間選擇APP(2)



日期、時間選擇APP(3)

- Calendar物件可用來查詢目前的日期及時間，以便作為日期/時間交談窗的預設值。
 - 此物件為系統資源，必須先用 `Calendar.getInstance()` 方法來取得。
 - 例如，`Calendar c = Calendar.getInstance();`
 - 再用 `c.get(Calendar.YEAR)`、`c.get(Calendar.MONTH)` 等來取得目前的年、月及時間資料。

日期、時間選擇APP(4)

```
01 public class MainActivity extends AppCompatActivity
02     implements OnClickListener, ←出現紅色波浪線時，可按 Alt + Enter 鍵
                                然後選擇 Implement Method 來自動加
                                入實作介面的方法
03         DatePickerDialog.OnDateSetListener, ←
                                實作監聽日期交談窗事件的介面
04         TimePickerDialog.OnTimeSetListener { ←
                                實作監聽時間交談窗事件的介面
05
06     Calendar c = Calendar.getInstance(); ← 建立日曆物件
07     TextView txDate; ← 記錄日期文字的元件
08     TextView txTime; ← 記錄時間文字的元件
```

日期、時間選擇APP(5)

```
01    protected void onCreate(Bundle savedInstanceState) {  
02        super.onCreate(savedInstanceState);  
03        setContentView(R.layout.activity_main);  
04  
05        txDate = (TextView)findViewById(R.id.textView1); ←取得用來顯示日期的 TextView  
06        txTime = (TextView)findViewById(R.id.textView2); ←取得用來顯示時間的 TextView  
07  
08        txDate.setOnClickListener(this); ←設定按下日期文字時的監聽物件  
09        txTime.setOnClickListener(this); ←設定按下時間文字時的監聽物件  
10    }
```

日期、時間選擇APP(6)

```
01     public void onClick(View v) {  
02         if(v == txDate) { ← 按的是日期文字  
03             //建立日期選擇交談窗，並傳入設定完成時的監聽物件  
04             new DatePickerDialog(this, this, ← 由 MainActivity  
05                 c.get(Calendar.YEAR), ← 由 Calendar 物件取得目前的西元年  
06                 c.get(Calendar.MONTH), ← 取得目前月 (由 0 算起)  
07                 c.get(Calendar.DAY_OF_MONTH)) ← 取得目前日  
08             .show(); ← 顯示交談窗  
09         }  
10         else if(v == txTime) { ← 按的是時間文字  
11             //建立時間選擇交談窗，並傳入設定完成時的監聽物件  
12             new TimePickerDialog(this, this, ← 由 MainActivity 監聽事件  
13                 c.get(Calendar.HOUR_OF_DAY), ← 取得目前的時(24小時制)  
14                 c.get(Calendar.MINUTE), ← 取得目前的分  
15                 true) ← 使用 24 小時制  
16             .show(); ← 顯示交談窗  
17         }  
18     }
```

日期、時間選擇APP(7)

```
01    public void onDateSet(DatePicker v, int y, int m, int d) { ←┐
                                           選定日期後的處理方法
02        txDate.setText("日期：" + y + "/" + (m+1) + "/" + d); ←┐
03    }                                     將選定日期顯示在螢幕上
04
05    @Override                             選定時間後的處理方法
06    public void onTimeSet(TimePicker v, int h, int m) { ←┐
07        txTime.setText("時間：" + h + ":" + m); ←┐
08    }                                     將選定的時間顯示在螢幕上
```

程式作業

- 完成一個簡單的Alert交談窗。
 - 將交談窗圖示改為綠色機器人。
- 完成問卷調查APP。
 - 若使用者按了沒意見按鈕時，改在在文字框上顯示“要不要試用看看 Android手機呢?”
- 完成日期、時間選擇APP。