

目錄

序

準備篇

- 一、使用說明 0-2
- 二、標示說明 0-6
- 三、注意事項 0-7

第一類 Android 基礎知識-UI 設計及語法應用

- 101. 整存整付計算機 1-2
- 102. 電費計算機 1-8
- 103. 點餐系統 1-18
- 104. 計算 BMI 值 1-23
- 105. MENU 功能選單 1-36
- 106. 畫廊展示 1-42
- 107. 驗證 Activity 1-48
- 108. 選擇手機型號 1-57
- 109. AlertDialog 顯示結果 1-61
- 110. 動態密碼顯示 1-69

第二類 資料儲存應用

- 201. 猜數字遊戲 2-2
- 202. 公尺與英呎轉換 2-9
- 203. 匯率換算 2-15
- 204. SQLite 與 MENU 作業 2-22
- 205. 圖檔讀取 2-34
- 206. 資料庫讀取 2-40
- 207. 密碼鎖 2-51
- 208. 我是誰? 2-57

- 209. 姓名清單 2-64
- 210. 簡單記事 2-69

第三類 基礎服務應用

- 301. 通訊錄搜尋 3-2
- 302. 接收 SMS 3-9
- 303. MP3 播放器 3-17
- 304. 接收簡訊 3-28
- 305. 接聽來電 3-35
- 306. 查詢地圖 3-40
- 307. 判斷發送簡訊狀態 3-51
- 308. 判斷地標所屬區域 3-60
- 309. 畫布程式 3-69
- 310. 程式背景音樂 3-76

第一類

Android 基礎知識-UI 設計及語法應用

- 101. 整存整付計算機
- 102. 電費計算機
- 103. 點餐系統
- 104. 計算 BMI 值
- 105. MENU 功能選單
- 106. 畫廊展示
- 107. 驗證 Activity
- 108. 選擇手機型號
- 109. AlertDialog 顯示結果
- 110. 動態密碼顯示

Android 2

101. 整存整付計算機

題解說明

A. 解題要項

- 螢幕畫面 Layout 設計
- strings.xml 檔字串設定
- 在 Java 檔裡，取得使用者輸入資料，經過運算處理後，將結果顯示在螢幕上

B. 重點描述

依照題目畫面的要求，在 strings.xml 檔裡加入所要顯示的字串，並給予每個字串 ID，其指令格式如：`<string name="ID">顯示的字串</string>`。

此題目是以 `LinearLayout` 來設計，螢幕畫面 `LinearLayout` 的特性是所加入的元件依序由上而下（或由左至右）排列。文字顯示使用 `<TextView>` 元件，設定使用者輸入的欄位使用 `<EditText>` 元件，按鈕使用 `<Button>` 元件。

程式實作

strings.xml

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <resources>
3      <string name="app_name">整存整付計算機</string>
4      <string name="title">整存整付試算：</string>
5      <string name="lend">本金(NT$):</string>
6      <string name="rate">年利率(%):</string>
7      <string name="number">存款期數:</string>
8      <string name="btn">計算本利和</string>
9  </resources>

```

程式說明

行	數	說	明
3~8		新增 string 並命名與宣告值	

【101. 整存整付計算機】

main.xml

```
1 <?xml version="1.0" encoding="utf-8" ?>
2
3 <!--在考題程式碼中，使用LinearLayout，並且android:orientation屬性設定為"vertical"
4 ，所加入的元件依序由上而下排列 -->
5 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
6     android:orientation="vertical"
7     android:layout_width="fill_parent"
8     android:layout_height="fill_parent">
9
10 <!-- 加入一個TextView元件，並設定顯示內容 -->
11 <TextView android:text="@string/title"
12     android:layout_width="fill_parent"
13     android:layout_height="wrap_content" />
14
15 <!-- 加入一個TextView元件，並設定顯示內容 -->
16 <TextView android:text="@string/lend"
17     android:layout_width="fill_parent"
18     android:layout_height="wrap_content"/>
19
20 <!-- 加入一個EditText元件，並設定輸入資料型態為integer -->
21 <EditText android:id="@+id/input_lend"
22     android:layout_width="fill_parent"
23     android:layout_height="wrap_content"
24     android:numeric="integer"/>
25
26 <!-- 加入一個TextView元件，並設定顯示內容 -->
27 <TextView android:text="@string/rate"
28     android:layout_width="fill_parent"
29     android:layout_height="wrap_content"/>
30
31 <!-- 加入一個EditText元件，並設定輸入資料型態為integer -->
32 <EditText android:id="@+id/input_rate"
33     android:layout_width="fill_parent"
34     android:layout_height="wrap_content"
35     android:numeric="integer"/>
36
37 <!-- 加入一個TextView元件，並設定顯示內容 -->
```

【101. 整存整付計算機】

```

38 <TextView android:text="@string/number"
39 android:layout_width="fill_parent"
40 android:layout_height="wrap_content"/>
41
42 <!-- 加入一個EditText元件，並設定輸入資料型態為integer -->
43 <EditText android:id="@+id/input_number"
44 android:layout_width="fill_parent"
45 android:layout_height="wrap_content"
46 android:numeric="integer"/>
47
48 <!-- 加入一個Button元件，設定按鈕上的顯示文字，按鈕的動作需要在Java程式中處
49 理，所以需要為此按鈕設定ID -->
50 <Button android:text="@string/btn"
51 android:id="@+id/submit"
52 android:layout_width="fill_parent"
53 android:layout_height="wrap_content"/>
54
55 <!-- 加入一個TextView元件，此TextView元件的顯示內容需要在Java程式中處理，所
56 以需要為此元件設定ID -->
57 <TextView android:text=""
58 android:id="@+id/amount"
59 android:layout_width="fill_parent"
60 android:layout_height="wrap_content"/>
61 </LinearLayout>

```

程式說明

行 數	說 明
12~14	加入一個 TextView 元件，字串的內容是從 ID 為 title 的 string 物件取得， TextView 元件的寬度填滿一整層， TextView 元件的高度與所顯示的內容同高
17~19	加入一個 TextView 元件，字串的內容是從 ID 為 lend 的 string 物件取得， TextView 元件的寬度填滿一整層， TextView 元件的高度與所顯示的內容同高
22~25	加入一個 EditText 元件，設定此 EditText 元件的 ID(input_lend)， EditText 元件的寬度填滿一整層， EditText 元件的高度與所顯示的內容同高，設定此欄位讓使用者輸入整數資料，鍵盤會出現數字鍵盤
28~30	加入一個 TextView 元件，字串的內容是從 ID 為 rate 的 string 物件取得， TextView 元件的寬度填滿一整層， TextView 元件的高

【101. 整存整付計算機】

	度與所顯示的內容同高
33~36	加入一個 EditText 元件，設定此 EditText 元件的 ID(input_rate)， EditText 元件的寬度填滿一整層， EditText 元件的高度與所顯示的內容同高，設定此欄位讓使用者輸入整數資料，鍵盤會出現數字鍵盤
39~41	加入一個 TextView 元件，字串的內容是從 ID 為 number 的 string 物件取得， TextView 元件的寬度填滿一整層， TextView 元件的高度與所顯示的內容同高
44~47	加入一個 EditText 元件，設定此 EditText 元件的 ID(input_number)， EditText 元件的寬度填滿一整層， EditText 元件的高度與所顯示的內容同高，設定此欄位讓使用者輸入整數資料，鍵盤會出現數字鍵盤
51~54	加入一個 Button 元件，按鈕上顯示的文字從 ID 為 btn 的 string 物件取得，設定此元件的 ID(submit)， Button 元件的寬度填滿一整層， Button 元件的高度與所顯示的內容同高
57~60	加入一個 TextView 元件，字串的內容空白，設定此元件的 ID(amount)， TextView 元件的寬度填滿一整層， TextView 元件的高度與所顯示的內容同高

GDD01.java

```

1 package COM.TQC.GDD01;
2
3 import java.text.DecimalFormat;
4
5 import android.app.Activity;
6 import android.os.Bundle;
7 /*以下這些套件所含的類別會在程式中使用到，所以必須import進來。在Eclipse中，
8 可以使用 ctrl+shift+o，將程式中所用到的類別import進來，程式設計師不用自行編
9 寫import的指令
10 */
11 import android.view.View;
12 import android.widget.Button;
13 import android.widget.EditText;
14 import android.widget.TextView;
15
16
17 public class GDD01 extends Activity {
18     /* Called when the activity is first created. */

```

【101. 整存整付計算機】

```
19     @Override
20     public void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.main);
23         findViews(); //將main檔裡的元件與Java檔裡的物件連結起來
24         setlisteners(); //監聽Button按鈕被按，執行相對應的動作
25     }
26
27     /*先做物件宣告，在main.xml檔裡，有1個<Button>，3個<EditText>，還有1個有ID的
28     <TextView>*/
29     private Button button_calc;
30     private EditText field_lend;
31     private EditText field_rate;
32     private EditText field_number;
33     private TextView view_result;
34
35     /*透過findViewById建構子，建立main.xml檔裡的元件與Java檔裡的物件連結關係 */
36     private void findViews(){
37         button_calc= (Button)findViewById(R.id.submit);
38         field_lend=(EditText)findViewById(R.id.input_lend);
39         field_rate=(EditText)findViewById(R.id.input_rate);
40         field_number=(EditText)findViewById(R.id.input_number);
41         view_result=(TextView)findViewById(R.id.amount);
42     }
43
44     /*監聽button_calc這個按鈕物件是否被按，如果被按，執行calcMoney方法 */
45     private void setlisteners(){
46         button_calc.setOnClickListener(calcMoney);
47     }
48
49     //calcMoney方法的宣告及程式碼
50     private Button.OnClickListener calcMoney=new Button.OnClickListener(){
51         public void onClick(View v){
52             //設定顯示數字的格式，小數點後0位
53             DecimalFormat nf=new DecimalFormat("0");
54             //取得field_lend物件裡的資料，存放在double型態的變數lend裡
55             double lend=Double.parseDouble(field_lend.getText().toString());
56             //取得field_rate物件裡的資料，存放在double型態的變數rate裡
```

【101. 整存整付計算機】


```

57     double rate=Double.parseDouble(field_rate.getText().toString());
58 //取得field_number物件裡的資料，存放在double型態的變數number裡
59     double number =Double.parseDouble(field_number.getText().toString());
60 //本利和=本金*(1+年利率/12)%^存款期數
61     double total=lend*Math.pow(1+rate/12/100, number);
62 //設定view_result物件的顯示字串
63     view_result.setText("本利和為"+nf.format(total));
64 }
65 };
66 }

```

程式說明

行 數	說 明
11~14	程式中所需要使用到的類別
20~25	覆寫 onCreate 方法，加入畫面顯示、選取畫面元件資料，以及按鍵監聽、處理功能
29~33	先物件宣告，在 main.xml 檔裡，有 1 個 <Button>，3 個 <EditText>，還有 1 個有 ID 的<TextView>，以上這些元件的資料和動作，都需要在 Java 檔裡處理，所以 Java 檔裡需要有相對應的元件
36~42	透過 findViewById 建構子，建立 main.xml 檔裡的元件與 Java 檔裡的物件連結關係
45~47	監聽 button_calc 這個按鈕物件是否被按，如果被按，執行 calcMoney 方法
50~65	當 button_calc 按鈕被按下時，執行以下所有的程式。從使用者輸入的欄位裡，將本金、利率、期數取出，計算出本利和，再將結果顯示在 TextView 元件裡