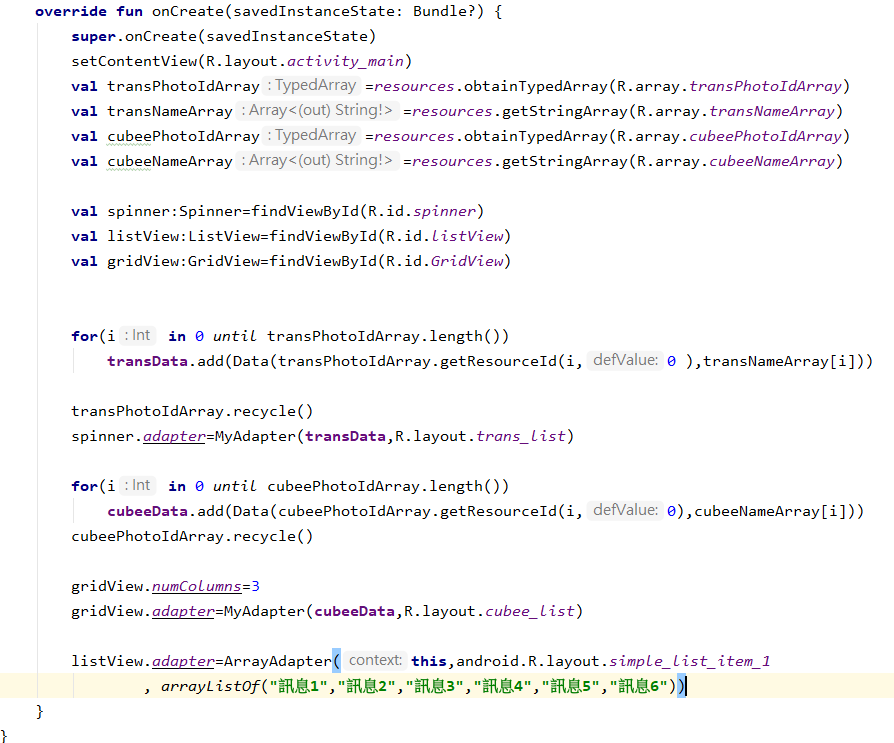
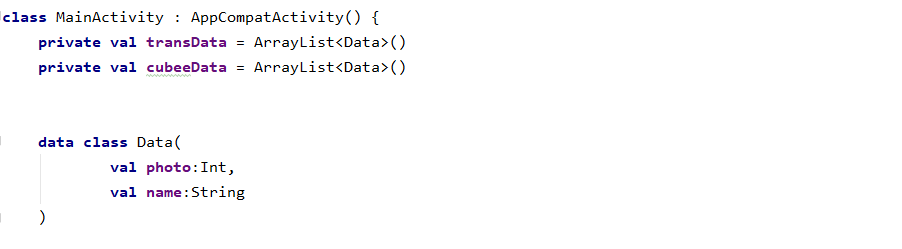
**107360133 電子三甲 蔡東霖**

*Lab6清單元件*

**Java**

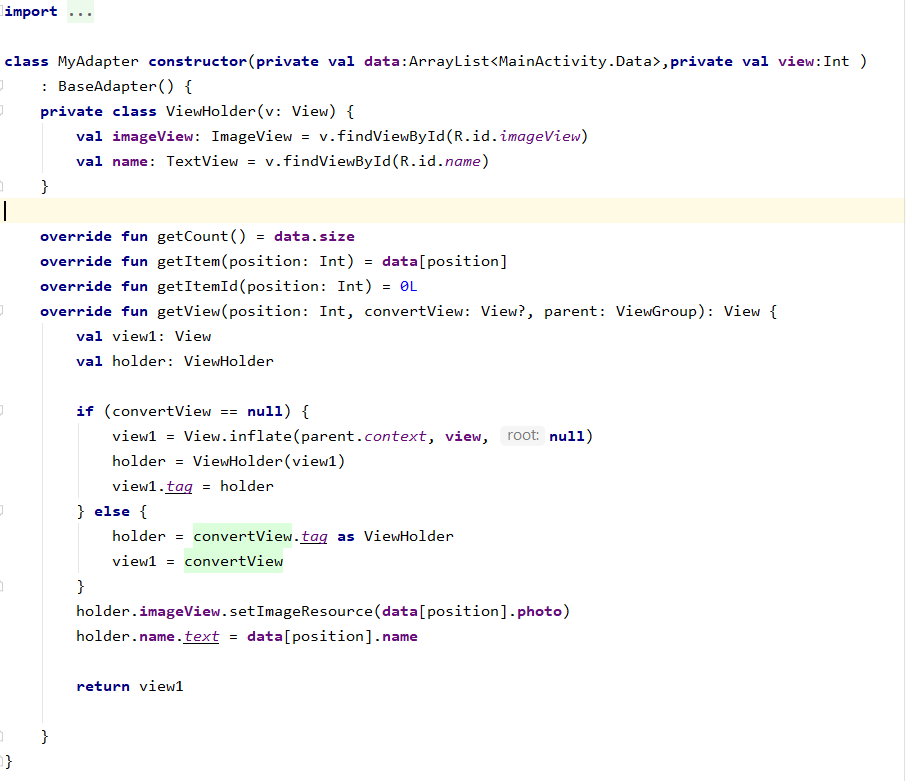


**Kotlin**

MainActivity



MyAdapter



上課時我們實作的清單元件(Java語法)是將Adapter寫在MainActivity裡面，但回家作業(Kotlin語法)是將Adapter和MainActivity分開來寫；除此之外，Kotlin的語法讓我們不用把每一個cubeelist和translist中的圖都一一做連接，我們使用陣列的方式將資料都存入，然後再用getstring的方式直接取出資料來顯示出來。

*Lab7非同步執行緒 7-1龜兔賽跑*

**Java**

宣告&監聽事件



runThread() & Handler

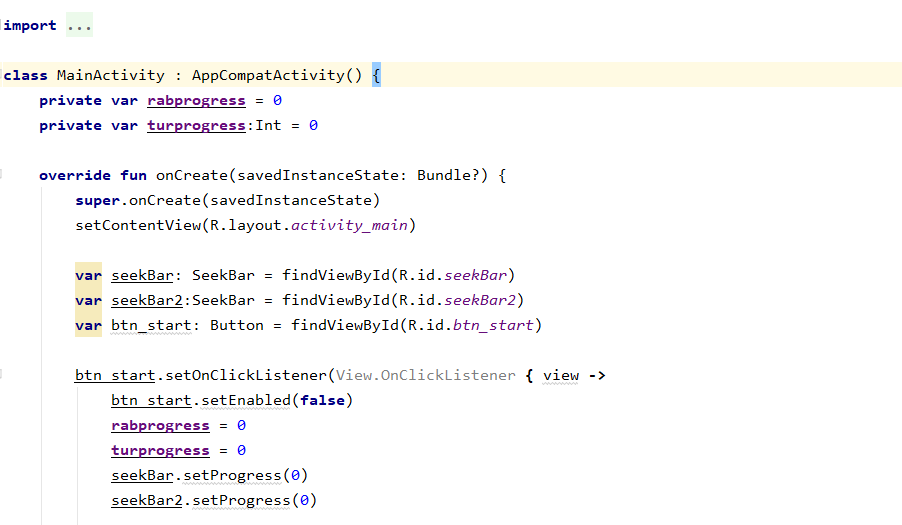


AsyncTask()



**Kotlin**

宣告&監聽事件



GloablScope.launch()

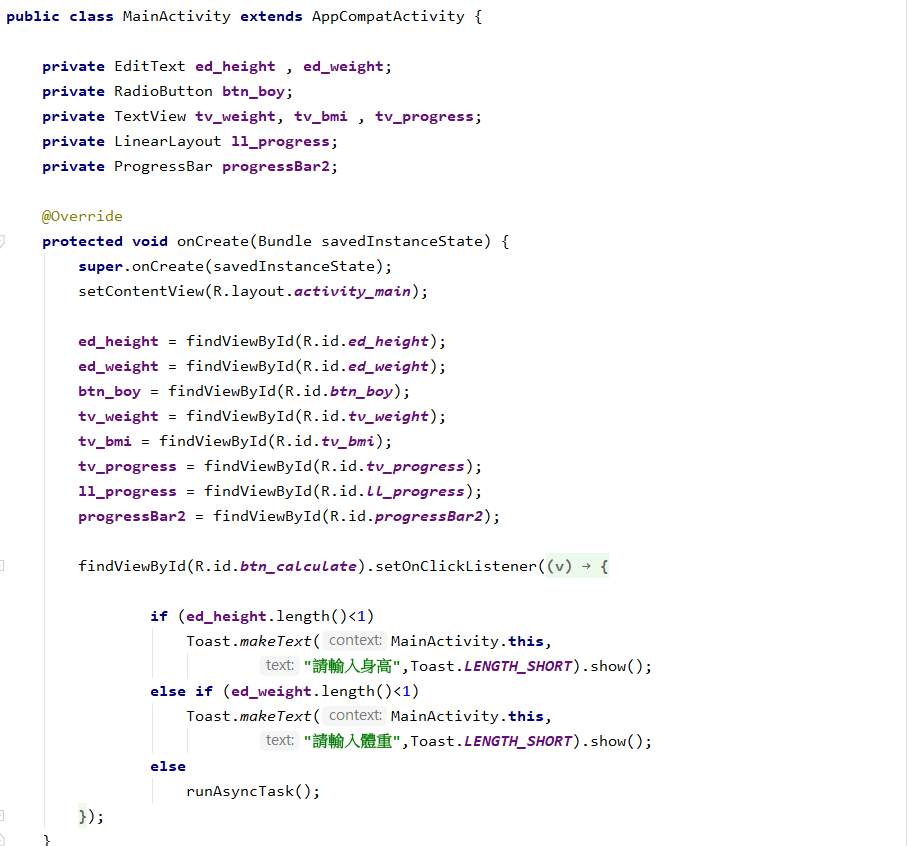


這次功課的要求是要我們將AsyncTask換成coroutine來執行相同的背景執行功能，AsyncTask和Coroutine最大的差異就在於，AsyncTask這個函式需要額外寫四個function依序來執行，分別是onPreExecute(初始化)、doInBackground(背景執行)、onProgressUpdate(進度更新)、onPostExecute(結果處理)，而最一開始的宣告 new AsyncTask<Integer , Void , String>這三個用於定義輸入、進度更新、輸出的資料型態，也會分別對應到doInBackGround、onProgressUpdate、onPostExecute的輸入值。

而coroutine的好處就是不需要再額外宣告出這麼長的程式碼，只需要宣告在try{}catch{}中不同的位置，就能夠達到上述的那四個function所產生的功能。

*Lab7非同步執行緒 7-2 BMI計算機*

**Java**





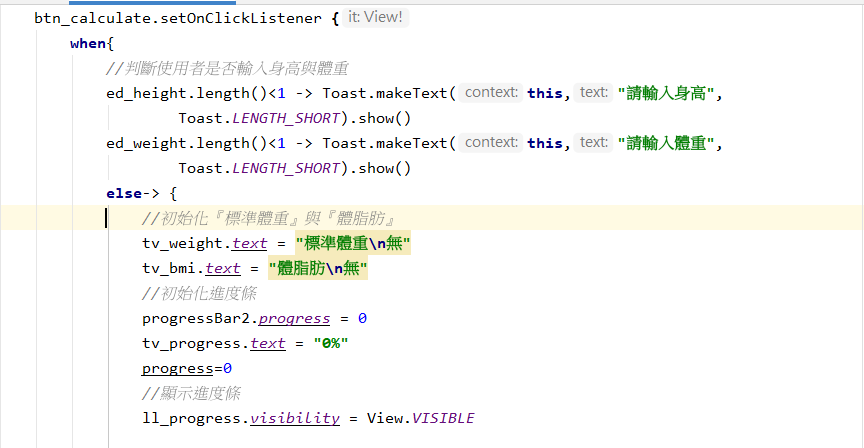


**Kotlin**

宣告



監聽事件



GlobalScope.launch()



心得：

這次的功課和以往都不太一樣，要自己思考的部分變多了，但這樣也是一件好事，這樣才能讓學到的東西印象更深刻，雖然目前還是沒有把整個概念理解得很清楚，不過相信多練習就可以搞懂了。