應用軟體實習——心得报告

測試环境：

1）Android Studio

2) java&Kotlin

電子三甲

裘翀皓

108360150

lab07：

使用Adapter和ListView元件去實現清單功能
圖片使用ImageView、文字使用TextView

```
<ImageView
    android:id="@+id/img_photo"
    android:layout_width="wrap_content"
    android:layout_height="80dp"
    android:adjustViewBounds="true"
    app:srcCompat="@android:drawable/ic_menu_gallery"
    />
<TextView
    android:id="@+id/tv_msg"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginStart="16dp"
    android:text="Textview"
    android:textSize="18sp"
    />
```

資料類別
```
data class Item(
 val photo : Int,
 val name : String,
 val price : Int
)
```
分別是圖片、名稱、價格。

最終實現結果，可以切換列表，列表下方會顯示隨機生成的水果價格

lab08：

使用到了ViewHolder，可以讓Adapter在更新畫面的時候，不必建立新的View實體。
RecylerView是進階版的ListView，強制開發者使用ViewHolder類別。
使用了SDK提供的圖片作為清單顯示素材。

資料類別：
```
data class Contact(
    val name: String, //姓名
    val phone: String //電話
)
```

lab09-1:

這是我們課堂上所做的龜兔賽跑的kotlin版本，下面是兩種程式的對比：

java：

```
try {
    Thread.sleep(100);
}catch (InterruptedException e){
```

kotlin:

```
delay(100)
```

java:

```
boolean[] sleepProbability = {true,true,false};
```

kotlin:

```
val sleepProbability = arrayOf(true , true ,false)
```

java:

```
}catch (InterruptedException e){
    e.printStackTrace();
}
```

kotlin:

```
} catch (e: InterruptedException){
    e.printStackTrace()
}
```

java:

```
Toast.makeText(MainActivity.this,
        "兔子勝利", Toast.LENGTH_SHORT).show();
btn_start.setEnabled(true);
```

kotlin:

```
showToast("兔子勝利")
btn_start.isEnabled = true
```

lab09-2:

利用Coroutiness實現模擬耗時的檢測過程，藉此機會了解同步執行的方法以其使用時機

使用了建立showToast方法顯示Toast訊息，提示資料未輸入完整

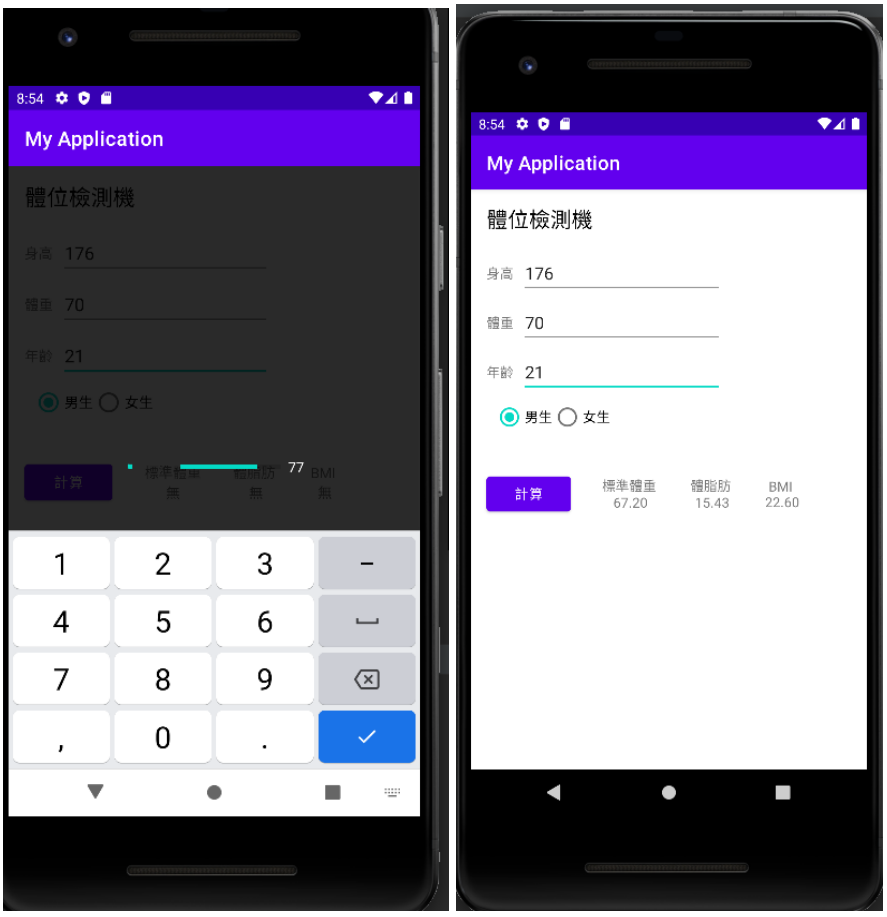
```
private fun showToast(msg: String) {
    Toast.makeText( context: this,msg,Toast.LENGTH_SHORT).show()
}
```

模擬進度條實現的程式，其中循環次數和延時可以隨意調整

```
//初始化進度條
progressBar2.progress = 0
tv_progress.text = "0%"
//顯示進度條
ll_progress.visibility = View.VISIBLE
GlobalScope.launch(Dispatchers.Main) {  this: CoroutineScope
    var progress = 0

    while (progress < 100) {
        delay( timeMillis: 50)
        progressBar2.progress = progress
        tv_progress.text = "$progress"
        progress++
    }
    ll_progress.visibility = View.GONE
```

實現結果如圖所示：



參考資料:

Android Kotlin實作開發-黃士嘉

Github:

lab07:

layout:
https://github.com/108360150-Qiuchonghao/Android_project_108360150/tree/master/kotlin---3/lab7/app/src/main/res/layout

kotlin程式碼:
https://github.com/108360150-Qiuchonghao/Android_project_108360150/tree/master/kotlin---3/lab7/app/src/main/java/com/example/lab7

lab08:

layout:
https://github.com/108360150-Qiuchonghao/Android_project_108360150/tree/master/kotlin---3/lab8/app/src/main/res/layout

kotlin程式碼:
https://github.com/108360150-Qiuchonghao/Android_project_108360150/tree/master/kotlin---3/lab8/app/src/main/java/com/example/ch8

lab09-1:

layout:
https://github.com/108360150-Qiuchonghao/Android_project_108360150/tree/master/kotlin---3/lab9--1/app/src/main/res/layout

kotlin程式碼:
https://github.com/108360150-Qiuchonghao/Android_project_108360150/tree/master/kotlin---3/lab9--1/app/src/main/java/com/example/myapplication

lab09-2:

layout:
https://github.com/108360150-Qiuchonghao/Android_project_108360150/tree/master/kotlin---3/lab9--2/app/src/main/res/layout

kotlin程式碼:
https://github.com/108360150-Qiuchonghao/Android_project_108360150/tree/master/kotlin---3/lab9--2/app/src/main/java/com/example/myapplication