

## 應用軟體設計實習 Kotlin Homework4

電子三甲 108360130 鄭秉宸

心得報告：這次的作業 lab12 我想跟上課的 lab8 是有所相關的，

至於作業 lab13 就是一個新東西，那經過編寫逐漸掌握

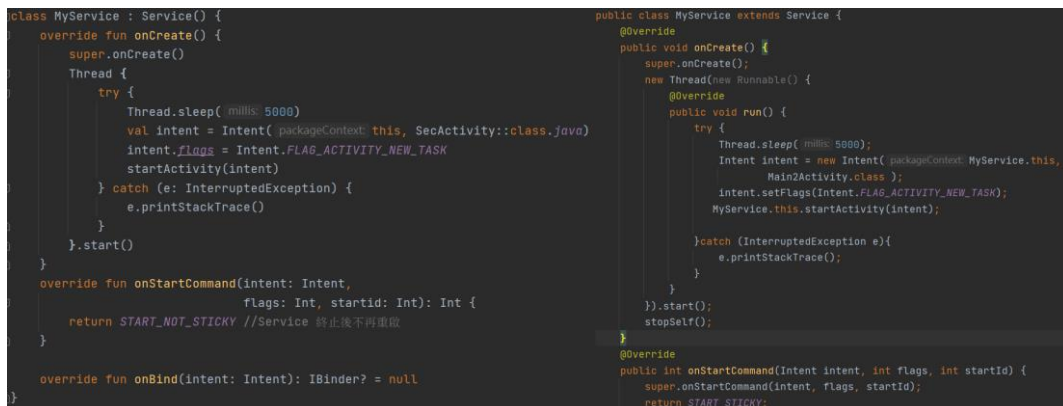
了 lab13 要教給我們的東西，就是 app 上面十分常見的

長按，長按可以做些甚麼呢，像是平常在用的 line，你

對一條訊息長按後就可以有許多新的功能可以給你使

用，我想這對每一個 app 來說，都是能將版面上的功能

給隱藏在長按後所出現的功能框，十分的好用!!



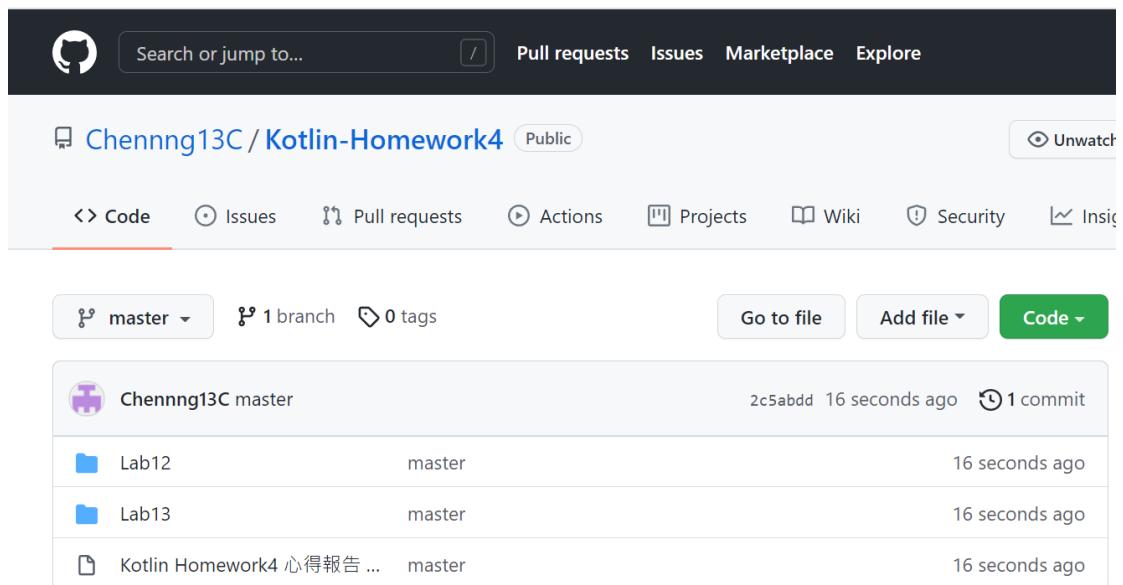
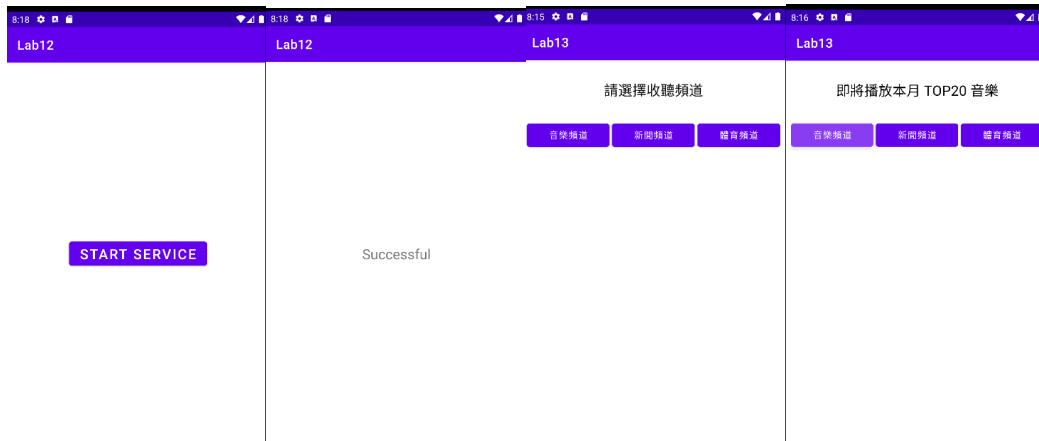
The image shows two side-by-side code snippets comparing Kotlin and Java implementations of a `MyService` class. The Kotlin code on the left uses `Thread` to execute a task in the background, starting another activity. The Java code on the right uses `Runnable` and `Thread` for the same purpose. Both implementations include a 5-second sleep and a catch for `InterruptedException`.

```
class MyService : Service() {
    override fun onCreate() {
        super.onCreate()
        Thread {
            try {
                Thread.sleep( millis: 5000)
                val intent = Intent( packageContext: this, SecActivity::class.java)
                intent.flags = Intent.FLAG_ACTIVITY_NEW_TASK
                startActivity(intent)
            } catch (e: InterruptedException) {
                e.printStackTrace()
            }
        }.start()
    }
    override fun onStartCommand(intent: Intent,
                                flags: Int, startId: Int): Int {
        return START_NOT_STICKY //Service 終止後不再重啟
    }
    override fun onBind(intent: Intent): IBinder? = null
}

public class MyService extends Service {
    @Override
    public void onCreate() {
        super.onCreate();
        new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    Thread.sleep( millis: 5000);
                    Intent intent = new Intent( packageContext: MyService.this,
                                                MainActivity.class );
                    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                    MyService.this.startActivity(intent);
                } catch (InterruptedException e){
                    e.printStackTrace();
                }
            }
        }).start();
        stopSelf();
    }
    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        super.onStartCommand(intent, flags, startId);
        return START_STICKY;
    }
}
```

### ● 使用 Thread 執行耗時任務 Kotlin<=>Java

主體來說並沒有太大的不同，在啟動 `MyService` 裡面不一樣的地方就是變數的使用方式，以及開始終止的副程式簡化。



- <https://github.com/Chenng13C/Kotlin-Homework4.git>