心得報告

讀書會:

組員: 109360136洪暉凱、109360117饒浩薰、109360143陳予謙

時間: 2022/12/29的晚上6:30

地點: google meeting

心得:

在課堂上中教過執行續是同時執行不同程式的方法，但這次作業特別的要求是讓我們使用Coroutine代替Thread完成執行續的工作，所以這次的作業也是花好多時間，幾乎都花在網路的爬文上。在Kotlin 官網中提到 Coroutine 是輕量化的 Thread，是一種協同式的分工，而Java語言中的Thread則屬於搶占式分工。

**Thread**

當 Thread 數量過多時就容易增加作業系統切換 Thread (上下文切換 Context Switch) 的負擔，影響整體效能。

**Coroutine**

因為 Coroutine 的切換是在上層，不需要由底層的作業系統來處理，所以 Coroutine 交替時所產生的上下文切換負擔比 Thread 小。

因為Coroutine間要做切換時可以由自己主動把執行權交給其他Coroutine，但

Thread間要做切換時必須透過作業系統安排執行權，所以Coroutine比Thread來的快許多。做完這次的作業後，我覺得Kotlin比Java輕巧的原因除了語法較簡潔外，可以使用Corountine也是一大重點。

語法差別:

Chapter12

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

Chapter13

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

照片:



Github連結: https://github.com/huikai67/HW9.git

Github截圖: