

應用軟體設計實習

HW Kotlin5

班級：電子三甲

姓名：賴德智

學號：110360139

Java 的 Thread 和 Kotlin 的 Coroutines 都是用於處理並行和異步編程的工具，但它們之間有一些重要的區別。以下是 Java 的 Thread 和 Kotlin 的 Coroutines 之間的一些主要差異：

* 語法風格：

Java 的 Thread 使用傳統的基於線程的模型，通常需要使用繼承 Thread 類或實現 Runnable 接口的方式來創建線程。

Kotlin 的 Coroutines 使用更具有表達性的語法，它提供了 suspend 修飾符，可以在非阻塞的情況下進行異步編程，而無需創建新的線程。

* 性能：

在 Java 中，創建和管理線程的成本相對較高，因為每個線程都需要分配內存和其他資源。

Kotlin 的 Coroutines 使用輕量級的協程，這些協程運行在較少的線程上，因此可以更有效地使用系統資源，提高性能。

* 阻塞 vs 非阻塞：

Java 的 Thread 在執行 I/O 操作或其他阻塞操作時，通常會導致整個線程阻塞。

Kotlin 的 Coroutines 使用非阻塞的方式，可以透過 suspend 修飾符將執行權交還給呼叫者，而不是阻塞整個線程。

* 協程的取消：

Kotlin 的 Coroutines 提供了協程取消的概念，可以方便地取消異步操作，這在處理大量的並行任務時非常有用。

在 Java 中，取消一個線程可能需要使用共享的變數或其他手段，相對比較複雜。

* 異常處理：

Java 的 Thread 使用 try-catch 塊來處理異常，而異常通常會中斷整個線程。

Kotlin 的 Coroutines 允許使用擁有更具表達性的異常處理方式，不會中斷整個協程，而是由開發者自己選擇是否要中斷。

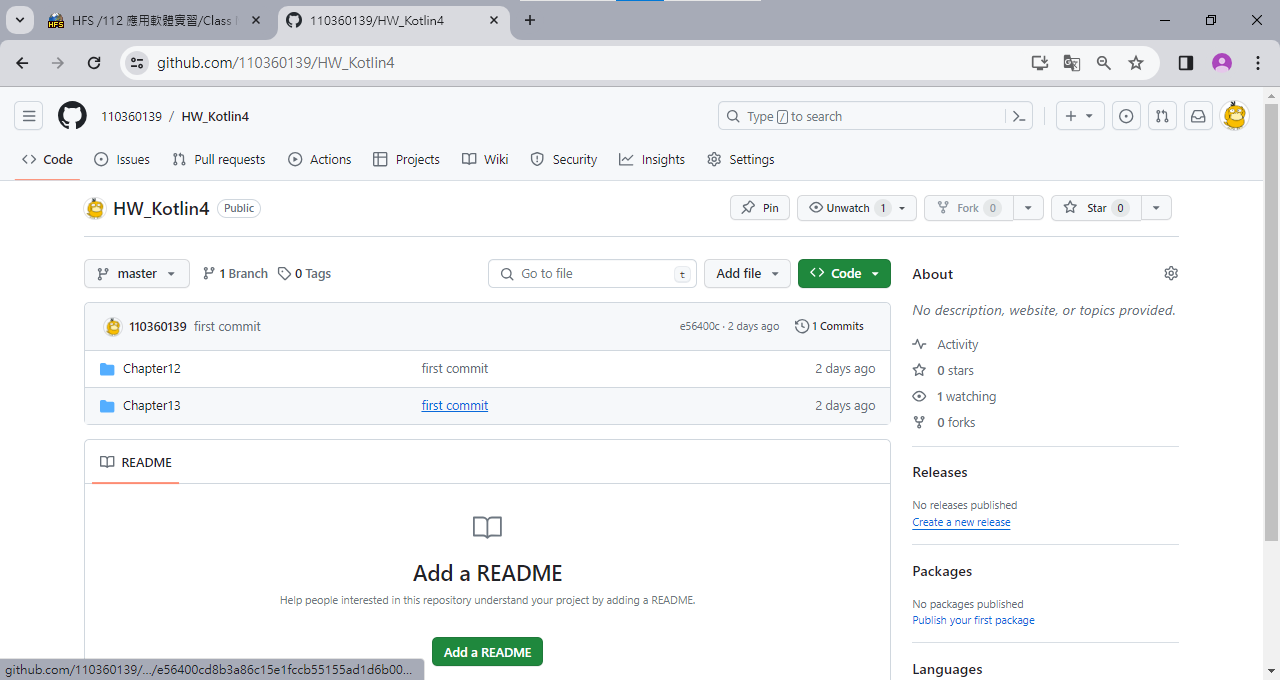
在本章之前的範例程式都是循著一條流程來進行處理的，在Java中可以同

時擁有多個處理流程。也就是說可以同時執行程式碼中不同的部分，這樣的一

個程式處理流程就稱為執行緒(thread)。

Kotlin 的 Coroutines 是一種用於簡化非同步和異步編程的機制。它們提供了一種更具表達性和方便的方式來處理非同步任務，避免了傳統回調風格的編碼，並且相較於傳統的多線程模型，它們更輕量、更具有可擴展性。

Github截圖 ( <https://github.com/110360139/HW_Kotlin4.git> )



心得:

Kotlin的寫法和java的寫法有些不一樣，不只是文法上的不同，演算法都有些差異，這讓我在讀懂程式碼時遇到不少困難，必須問問chatgpt才勉強看出架構。

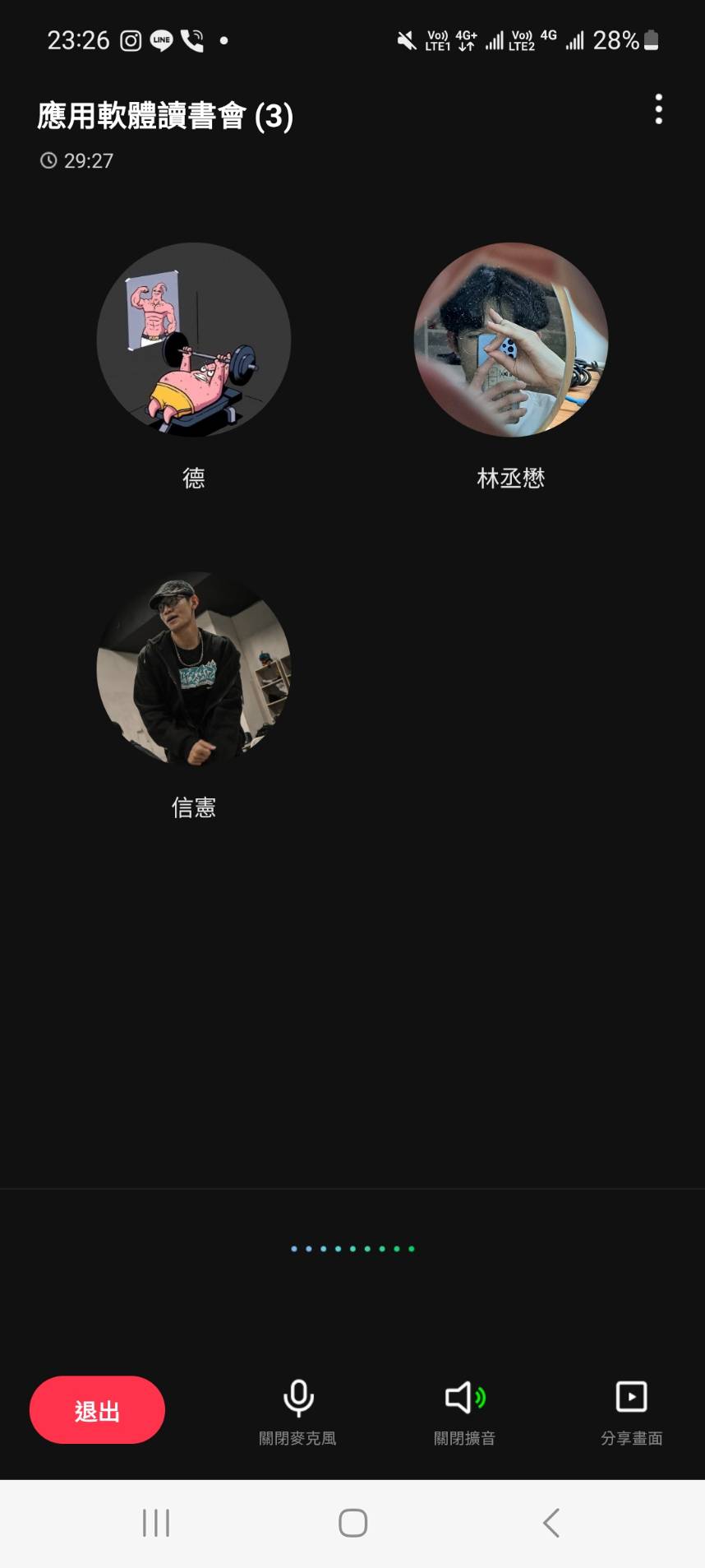
再來是LAB7所遇到的問題，除了要匯入圖片，還要額外改string的程式碼，這是以往沒遇過的情況，還是靠著同學的幫助才完成。

程式碼截圖

Ch14

Ch15

討論截圖



討論時間:12/12 晚上23:00

討論地點:線上

組員:林丞懋,曾信憲,賴德智

ChatGPT





