責任鏈模式 / 2019.12.29

DESIGN PATTERN 讀書會 / BOB CHANG

CHAIN-OF-RESPONSIBILITY PATTERN

自我介紹

- 張家銘 Bob Chang
- 國立宜蘭大學電子系學士畢業
- iOS app 開發自學者
- 2015 July 開始學習 `Swift`
- twitter @bob910078
- bob910078@gmail.com

https://cdn2.ettoday.net/images/1408/1408485.jpg

自己不做事，交給下一位做事的 PATTERN

也像是在教室裡傳紙條給某人

1. 最勇敢同學:
明明就坐在第一排,
卻堅信”最危險的地方
就是最安全的地方”,
堅持不認真上課。(勇者)

2. 假認真同學:
因為太晚來,所以坐在
很前排,但根本聽不懂老
師在說什麼,只好微笑點頭。

3. 超用功同學:
這種同學喜歡坐在中間
偏前的座位,抬頭聽老
師說話最舒服的位置。

4. 做自己事同學:
來了喜歡做自己的事情,
沒在聽課,喜歡坐在高大
的同學後面當掩護。

5. 他好累同學:
很可能整晚沒睡覺,
然後就來上課了,
老師通常也懶得理他。

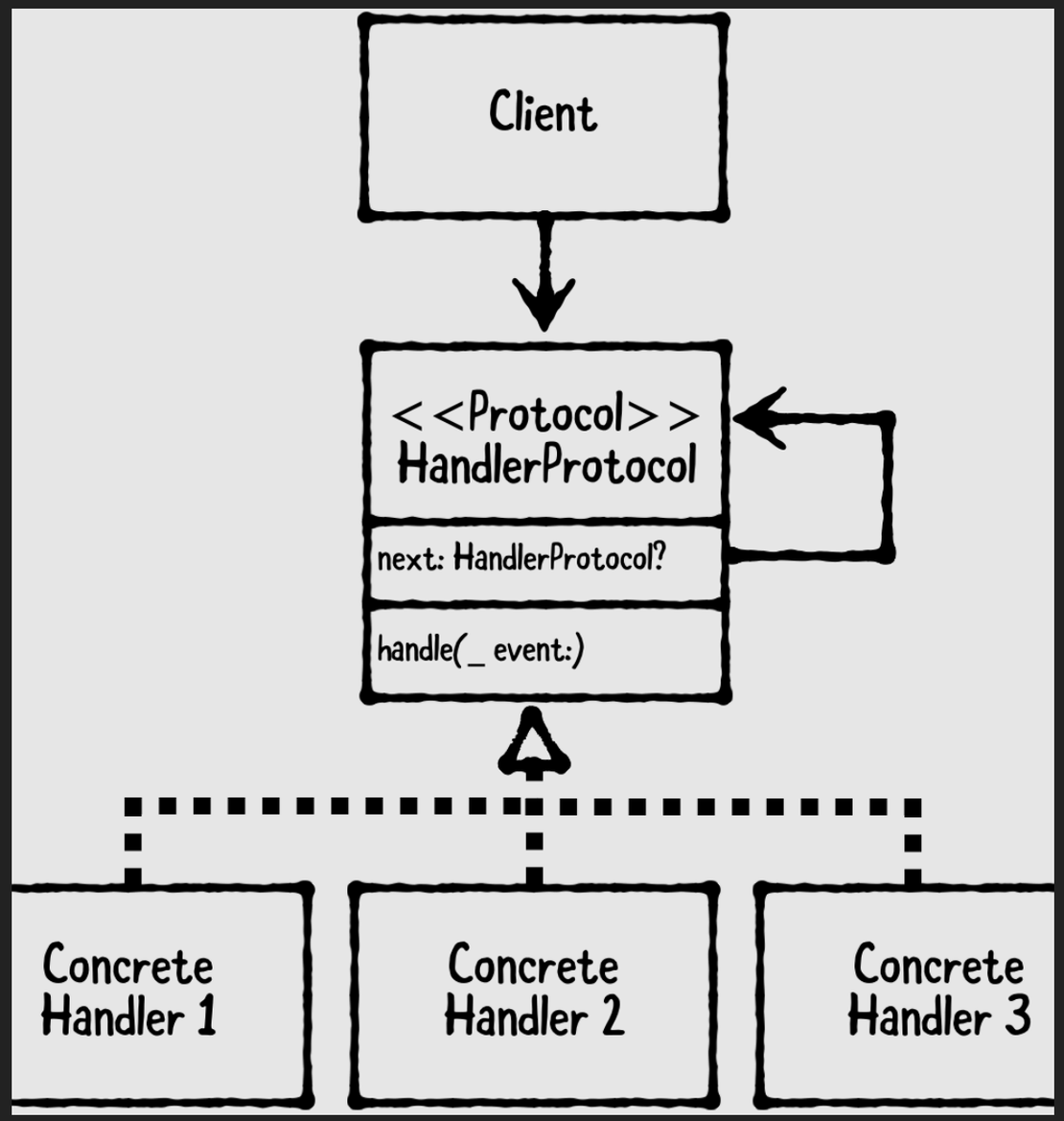
6. 有這個同學?
很少來,除非考試,
有的時候還請別人幫
忙佔位子,最後還是
沒來...

吃到飽

http://fc.life.com.tw/upload_file/6/content/d2c25b55-53c8-d584-5253-93619f0f629f.jpg

TEXT

```
Protocol HandlerProtocol {
    var next: HandlerProtocol?
    func handle(_ event: Int)
}
```



TEXT

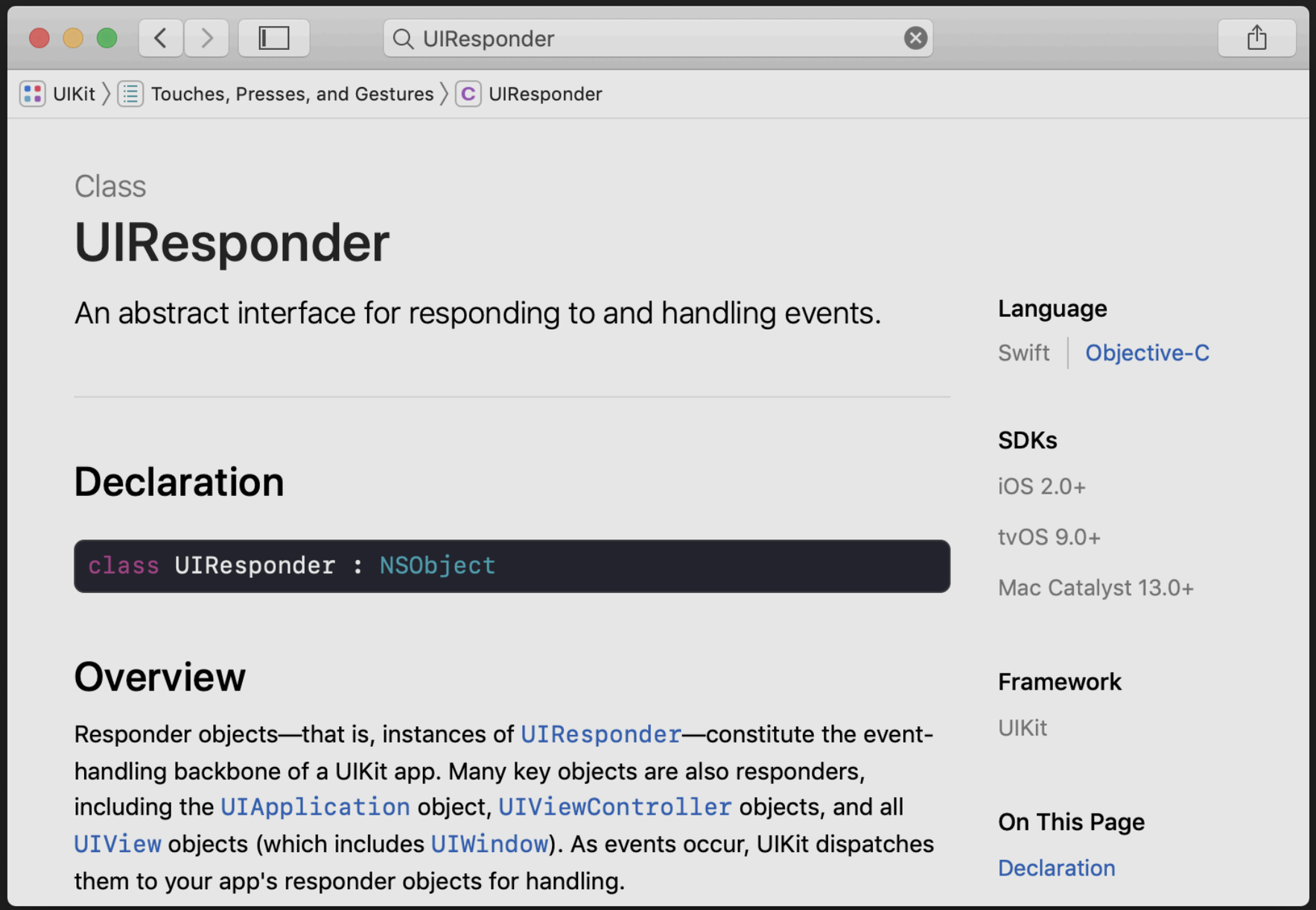
WHEN USE IT? WHAT TO NOTICE? HOW TO USE IT?

- 什麼時候可以用這個 pattern？
  - 需要有一系列的物件去處理相似的事件或是資料
  - (例如自動販賣機的硬幣偵測、觸摸螢幕的事件處理)

- 使用時可能需要注意什麼？
  - 我們這一系列的物件都無法處理輸入事件的狀況
  - (都沒有人要處理該怎麼辦)

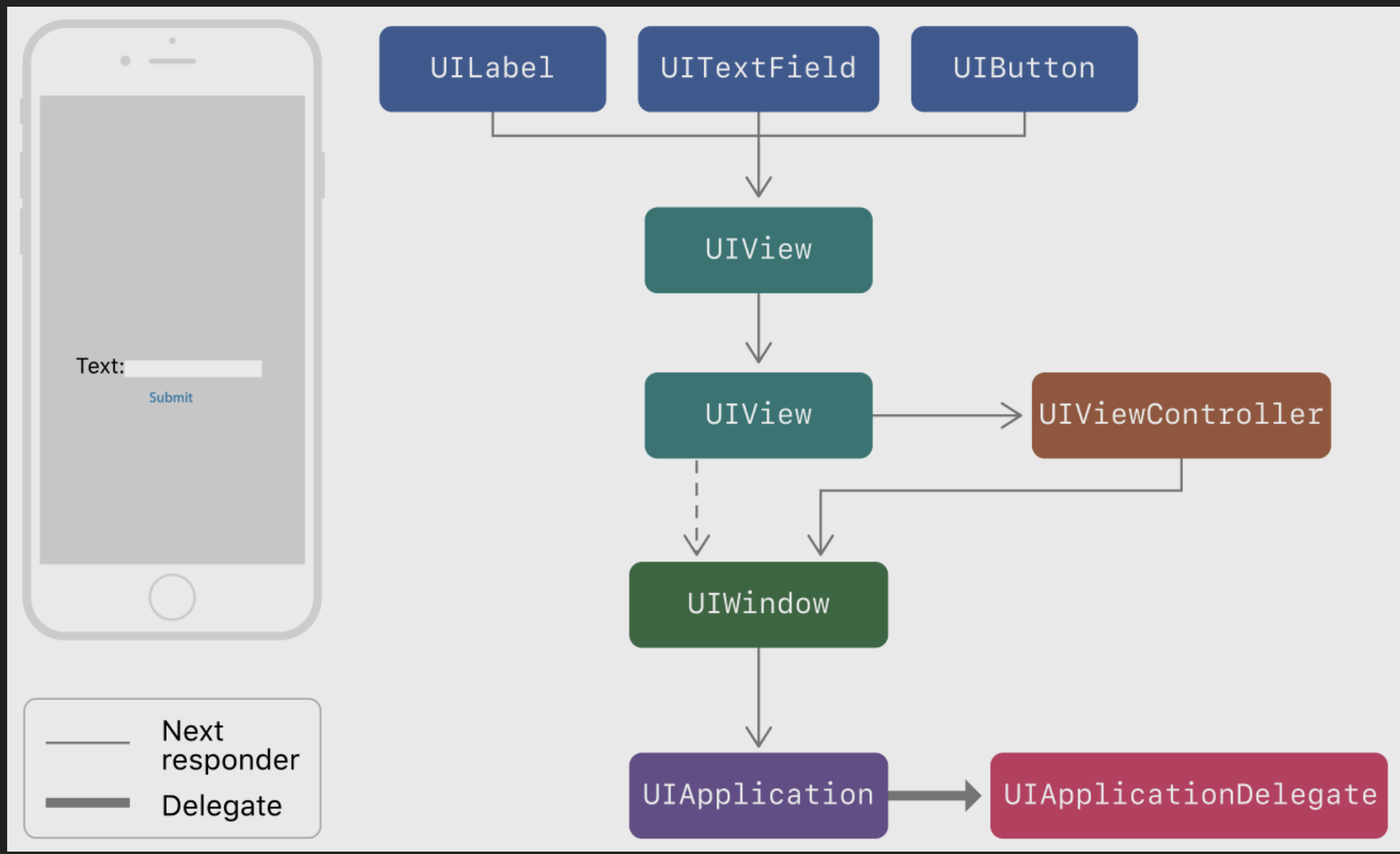
- 如何使用這個 pattern？
  - 責任鏈中的每位響應者都必須要接受某一事件，
  - 並自行決定「是否處理」或是「傳遞給下一位響應者」

典型應用：UIRESPONDER



https://developer.apple.com/documentation/uikit/uiresponder

RESPONDER CHAINS IN AN APP



https://developer.apple.com/documentation/uikit/touches_presses_and_gestures/using_responders_and_the_responder_chain_to_handle_events?source=post_page-----cc836d7c9e61----------

典型應用：UIRESPONDER (續)

- UIResponder 雖然是一個類別，但是其精神不出 chain-of-responder pattern
- 重要的兩個概念：`optional 繼任者` 和 `事務處理區塊`
- var next: UIResponder?
- func touch began(Set<UITouch>, with: UIEvent?)

OVERRIDE UIRESPONDER.TOUCHESBEGAN(_:WITH:)

- UIView
- UIViewController
- UIWindow
- UIApplication
- AppDelegate

```
extension UIView {
    override func touchesBegan(
        _ touches: Set<UITouch>,
        with event: UIEvent?) {
        print("UIView")
        next?.touchesBegan(touches, with: event)
    }
}
```

結束