

Test-Driven Development Workshop – Feedback

NM6121030 余振揚

在這次學習 TDD 的過程中，我對它的流程和理念有了新的體會。TDD 的核心是「Red-Green-Refactor」，也就是先寫測試(Red)，再寫程式讓測試通過(Green)，最後整理和優化程式碼(Refactor)。這樣的流程看似簡單，但實際操作時卻需要非常清楚地理解需求，並保持專注。

透過課堂上的 FizzBuzz 練習，我發現 TDD 其實是一種很有條理的開發方式。它強調每一步都以測試為基礎，這不僅減少了出錯的機會，也讓開發過程更有方向感。我也意識到，這樣的方法能夠幫助我們面對突然的需求變更，例如增加新的規則或功能時，因為我們有測試保護，改動起來更有信心。

身為一名研究生，我的研究時常需要撰寫實驗程式或處理數據，而這些工作常常會遇到程式出錯或結果不如預期的情況。如果能用 TDD 來設計程式，就能先定義出目標，然後一步一步實現，避免到最後才發現整個方向錯了。

例如，我最近在進行一項與數據分析相關的研究，處理過程中常需要整理大量資料。如果能在每個步驟前設計好測試，例如確認資料格式正確或計算結果符合預期，那麼即使中途需要修改程式，也不用擔心會影響之前的工作。

這次學習讓我對 TDD 的實用性有了新的認識。我希望未來可以更多地在自己的研究中嘗試 TDD，特別是在開發一些實驗工具或模擬系統時，利用測試來確保每一部分功能都穩定可用。另外，我也會嘗試將這種方法分享給同學或學術團隊，讓大家的研究開發更高效、更可靠。

TDD 讓我看到了一種全新的工作方式：它重視計畫、執行與反思，每一步都清晰有序。對於像我這樣的學生來說，TDD 不僅是一種寫程式技巧，更是一種讓研究更有條理的好工具。