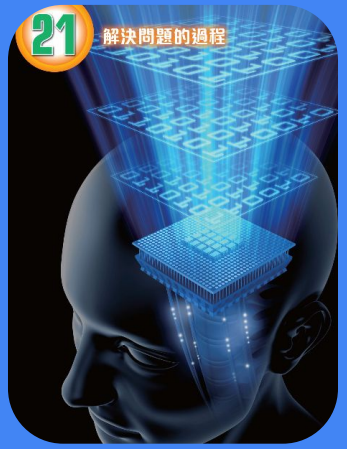
新界喇沙中學
中四級 新高中資訊及通訊科技



必修部分3 單完(D) - 基本程式編寫概念

單元 21 - 解決問題的過程

課堂(二)

任教老師: 陳昌文(主教) , 郭澤坤(助教)

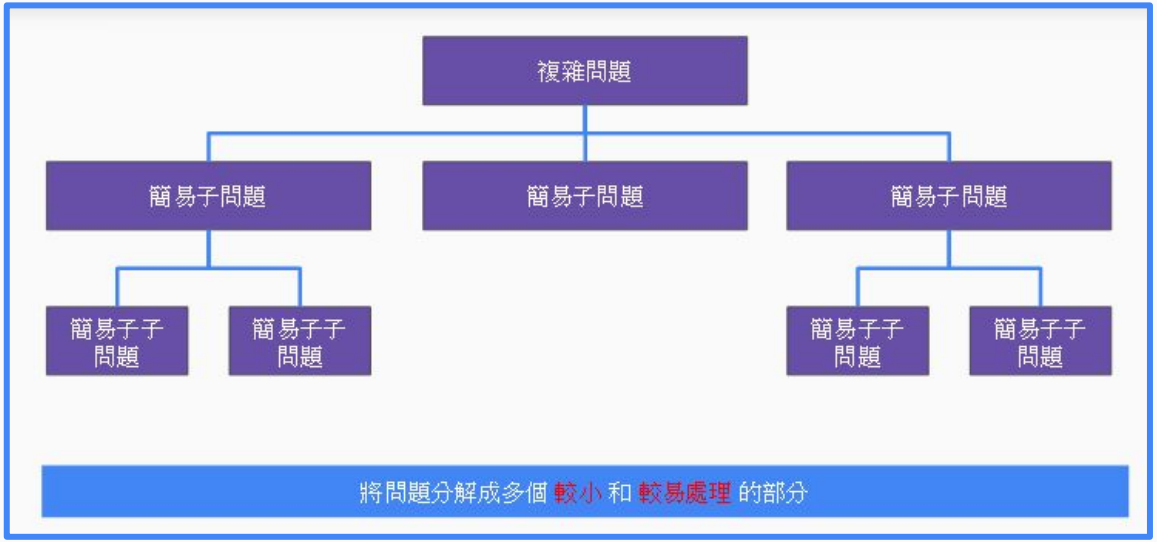
14/3/2017

課堂流程

1. 上堂內容重溫
2. 單元(21.2) - 解決問題的基本過程 (2)
    2.1. 算法設計
    2.2. 構擬解決方法
    2.3. 除錯和測試
    2.4. 文件編製

上堂內容重溫

- 分治法是什麼...?
- 分治法有什麼好處..?
- 釐清問題 - 腦力激盪法
- 問題分析 - IPO圖



21.2 解決問題的基本過程 (2)

六個解決問題的步驟

今天內容:

- 算法設計
- 除錯和測試
- 構擬解決方法
- 文件編製

算法設計

- **算法**: 一組有明確邏輯次序, 用來解決問題的步驟
- 兩種設計工具來設計和表示算法:
  - 偽代碼
  - 流程圖

偽代碼

以文字敍述的形式表示算法

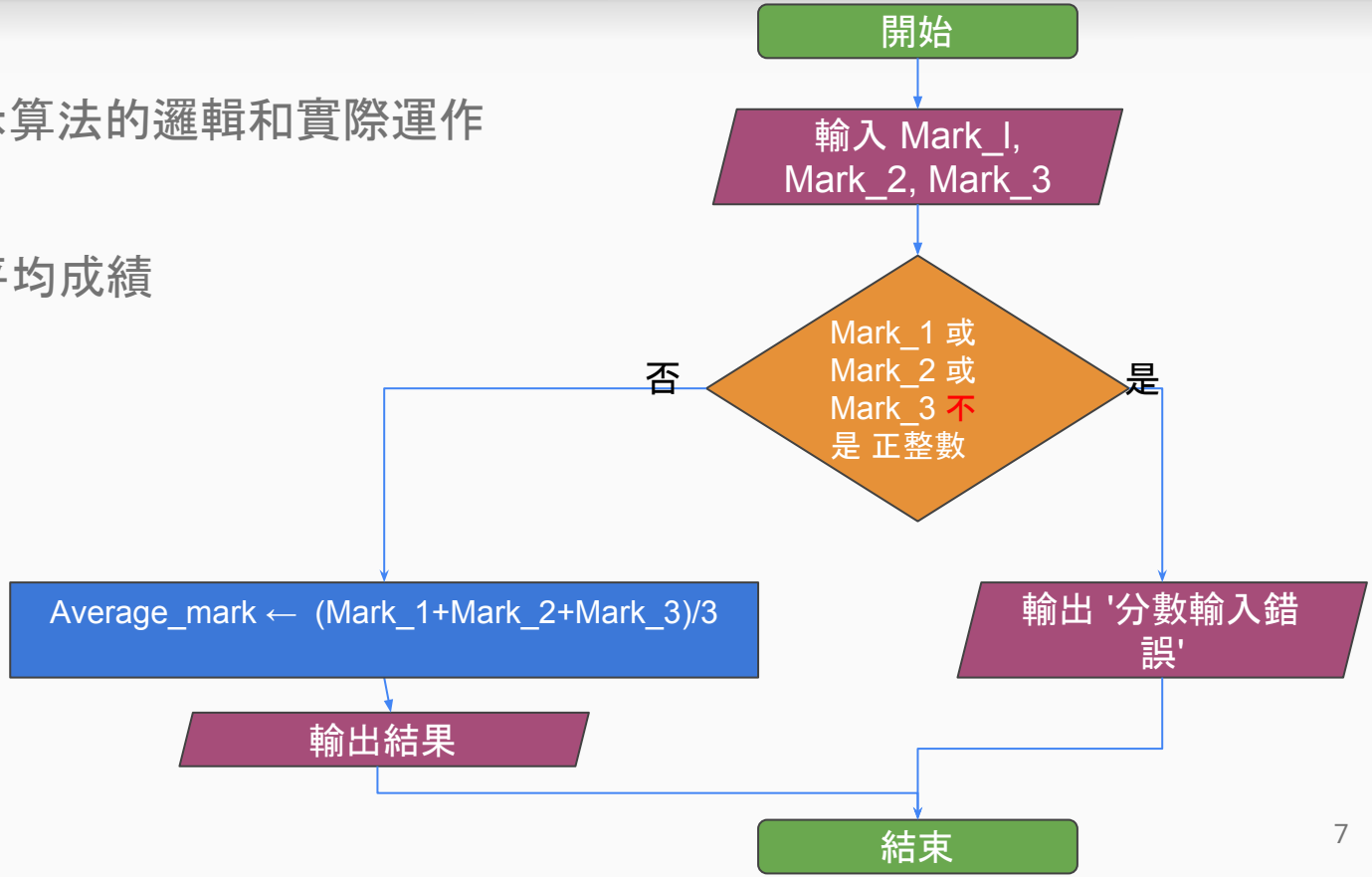
例子: 計算3位學生平均成績

```
輸入 Mark_l, Mark_2, Mark_3

IF Mark_1 或 Mark_2 或 Mark_3 不是 正整數 THEN

        輸出 '分數輸入錯誤'

ELSE

        Average_mark ← (Mark_1+Mark_2+Mark_3)/3

找出平均成績

輸出結果

END IF
```

流程圖

以**圖像的形式**來表示算法的邏輯和實際運作

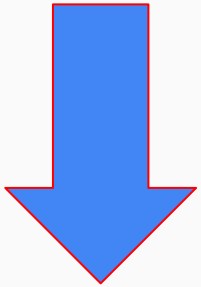
例子: 計算3位學生平均成績



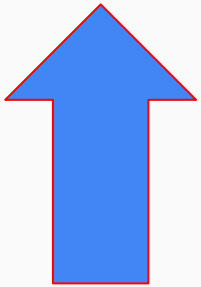
構擬解決方法

利用不同的技巧來構擬解決問題的方案

- 由上而下式

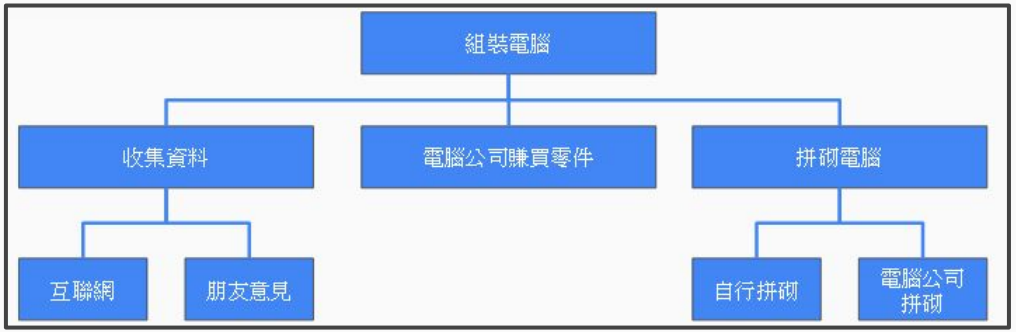


- 由下而上式



由上而下式

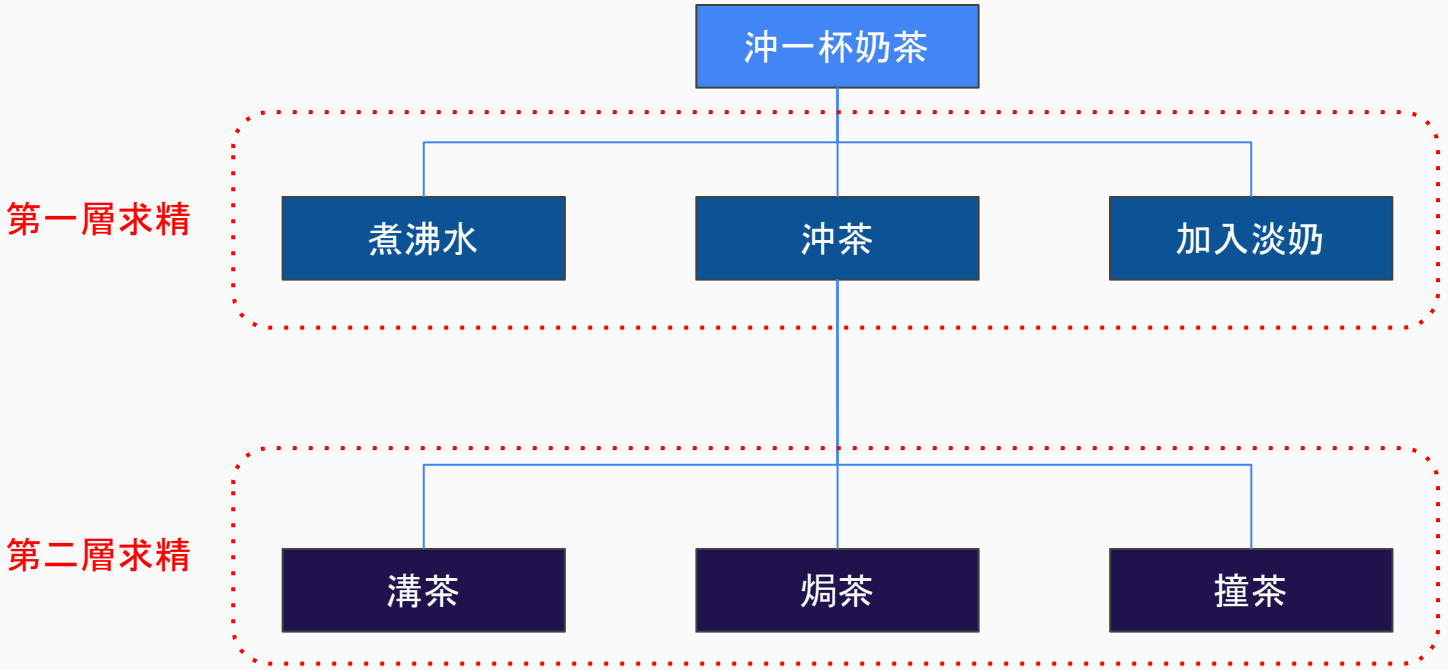
- 將問題分解成若干較易處理的子問題
- 解決子問題, 並把各子問題的解決方案組織起來, 以解決整個問題
- 利用由上而下式的技巧來構擬解決問題的方案
- 可避免在過程中可能出現的錯漏



由上而下式

- 應用**分治法**, 把一個複雜、抽象的問題分解成若干具體的細節問題
- 通過 *分解問題* 以及進一步分解子問題, 我們可得出一個**結構圖**
- 結構圖由不同的**模組**組成, 每個模組代表不同的大小問題
- 以**逐步求精法**的技巧來構擬解決方案

結構圖例子 -「沖一杯奶茶」模組的結構圖



構擬解決方案 - 由上而下式

- 由上而下式 - 各模組都有其**規格說明**
- 模組規格說明讓我們理解模組與模組之間的**數據傳輸**。
- 包括:

1. 輸入　　(所接收的數據)
2. 處理步驟　　(所用的邏輯)
3. 輸出　　(傳回的資訊)

構擬解決方案 - 由上而下式

- 「沖奶茶」例子的模組規格說明:

輸入	處理步驟:	輸出:
沸水, 茶葉	1.) 溝茶, 焗茶, 撞茶 2.) 加入淡奶, 沸水 3.) 把茶倒入杯內。	一杯奶茶

除錯和測試

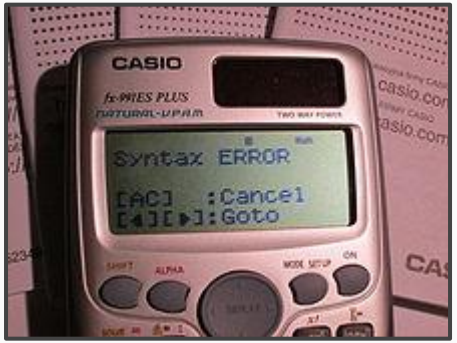
- **除錯**: 查找和清除錯誤的過程
- **測試**: 以確保程序能正常運作, 不受錯誤干擾

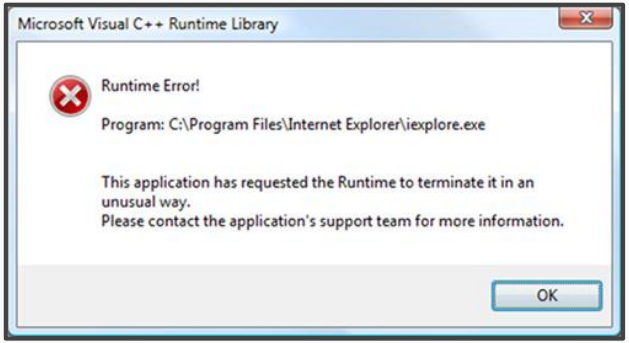
檢查電腦程序的過程, 用於確定某程序是否**合乎設計的要求**

除錯和測試

三種類型的錯誤：

1. 語法錯誤
2. 運行時錯誤
3. 邏輯錯誤






```
Enter 1st value: 5
Enter 2nd value: 3
The sum is: 0
```

三種錯誤類型

	語法錯誤	運行時錯誤	邏輯錯誤
定義	當程序指令不符合程序語言的語法規則時，會出現錯誤訊息。	錯誤只會在程序運行時出現。	錯誤是由不正確的程序邏輯設計所造成。預期結果與實際結果通常不一致。
典型例子	• 程序中的指令字拼法有錯。 • 忽略標點符號。 • 指令字的出現次序有錯。	• 數據異常錯誤，例如除數為零 • 系統資源不足，例如記憶體不足	• 不正確地使用控 制結構 • 不正確地使用指令
識別錯誤的困難程度	容易	一般	困難

三種錯誤類型

	語法錯誤	運行時錯誤	邏輯錯誤
定義	當程序指令不符合程序語言的語法規則時, 會出現錯誤訊息。	錯誤只會在程序運行時出現	錯誤是由不正確的程序邏輯設計所造成。預期結果與實際結果通常不一致。
典型例子	• 程序中的指令字拼法有錯 • 忽略標點符號 • 指令字的出現次序有錯	• 數據異常錯誤, 例如除數為零 • 系統資源不足, 例如記憶體不足	• 不正確地使用控 制結構 • 不正確地使用指令
識別錯誤的困難程度	容易	一般	困難

三種錯誤類型

	語法錯誤	運行時錯誤	邏輯錯誤
定義	當程序指令不符合程序語言的語法規則時, 會出現錯誤訊息。	錯誤只會在程序運行時出現	錯誤是由不正確的程序邏輯設計所造成。預期結果與實際結果通常不一致。
典型例子	• 程序中的指令字拼法有錯 • 忽略標點符號 • 指令字的出現次序有錯	• 數據異常錯誤, 例如除數為零 • 系統資源不足, 例如記憶體不足	• 不正確地使用控 制結構 • 不正確地使用指令
識別錯誤的困難程度	容易	一般	困難

三種錯誤類型

	語法錯誤	運行時錯誤	邏輯錯誤
定義	當程序指令不符合程序語言的語法規則時, 會出現錯誤訊息。	錯誤只會在程序運行時出現	錯誤是由不正確的程序邏輯設計所造成。預期結果與實際結果通常不一致。
典型例子	• 程序中的指令字拼法有錯 • 忽略標點符號 • 指令字的出現次序有錯	• 數據異常錯誤, 例如除數為零 • 系統資源不足, 例如記憶體不足	• 不正確地使用控制結構 • 不正確地使用指令
識別錯誤的困難程度	容易	一般	困難

相應的除錯方法

	語法錯誤	運行時錯誤	邏輯錯誤
除錯方法	• 如果程序存在語法錯誤, 在編譯程序時會出現錯誤信息 • 一些程序開發工具可用以檢查程序中的語法錯誤	利用測試數據來產生可能的運行時錯誤, 然後設法除錯	研究程序算法的邏輯, 以找出錯誤的原因。
備註	我們可使用除錯工具來找出語法錯誤	在除錯及測試中, 程序編寫員有時會故意製造運行時錯誤, 以測試程序的可行性。	要解決邏輯錯誤是最繁瑣的, 所花的時間亦是最長。除錯器可協助追蹤邏輯錯誤。

ICT 知識分享

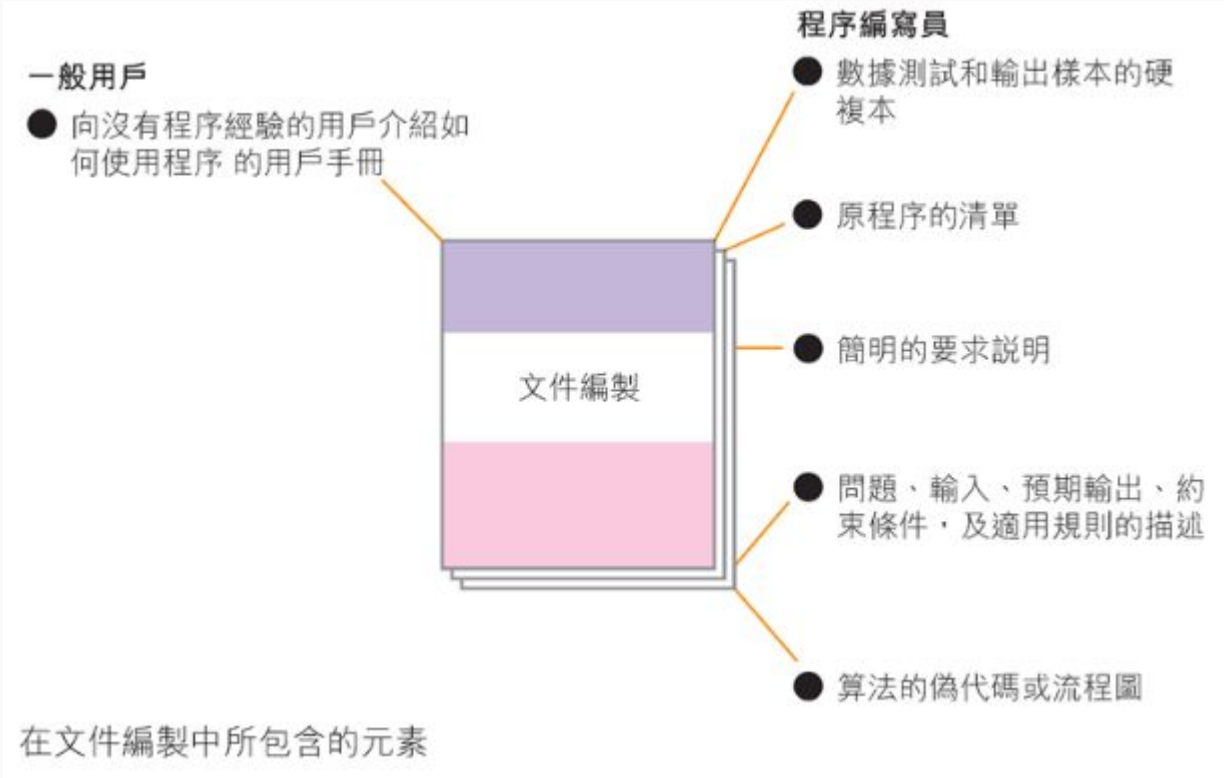
千年蟲問題

是指由於電腦程式設計的一些問題，使得電腦在處理2000年1月1日以後的日期和時間時候，可能會出現不正確的操作，從而可能導致一些敏感的工業部門（比如電力、能源）和銀行、政府等部門在2000年1月1日零點工作停頓甚至是發生災難性的結果。千年蟲問題多數出現在約十年前出產的舊式電子產品。

文件編製

- 描述電腦程序設計, 開發和測試的過程
- 解決問題過程的最後步驟
- 每個步驟中都進行文件編製

文件編製



在文件編製中所包含的元素

文件編製

文件編製主要有...

兩類讀者 : 程序編寫員 一般用戶

兩類文件 : 程序手冊 用戶手冊

文件編製用途

	用戶手冊	**程序手冊**
讀者	一般用戶	程序編寫員
主要內容	→ 怎樣安裝該程序 → 怎樣啟動該程序 → 該程序所提供的功能 → 怎樣使用該程序來完成特定工作 → 怎樣處理一些簡單的錯誤，並解釋錯誤信息的意義。	→ 技術性的資料 → 問題說明 → 程序要求說明 → 算法 → 測試數據和輸出樣本
寫作樣式	包含簡單描述、屏幕捕捉及選單流程的 非技術性文章	包含結構化的文字、偽代碼和流程圖的 技術性文章

文件編製

文件編製**十分重要**, 因為:

→ 可協助程序編寫員在將來**有效**地**維護程序**

→ 可協助**新參與者熟悉**程序的內容

→ 可協助程序編寫員**找出程序中的錯誤**

→ 一般用戶可透過用戶手冊來學會**使用程序**

內容重溫

- 算法設計
  - 偽代碼
  - 流程圖
- 構擬解決方法
  - 由上而下式
  - 模組之間的數據傳輸

內容重溫

- 除錯和測試
  - 三種類型的錯誤: 語法錯誤, 運行時錯誤,邏輯錯誤
- 文件編製
  - 兩類讀者: 一般用戶, 程序編寫員
  - 兩類文件: 用戶手冊, 程序手冊

參考資料

培生朗文 新高中資訊及通訊科技 必修單元3 單元21

https://www.iconexperience.com/g_collection/icons/?icon=debug

https://zh.wikipedia.org/wiki/%E8%AF%AD%E6%B3%95%E9%94%99%E8%AF%AF

http://www.computerhope.com/jargon/r/runtimee.htm

https://www.drupal.org/project/debug_tools

https://zh.wikipedia.org/wiki/2000%E5%B9%B4%E9%97%AE%E9%A2%98