CHAP12 電腦解題概論

高中資訊科技概論

松崗圖書公司

本章目錄

12-1 電腦解題簡介
- 一、垂直式思考
- 二、循序漸進的流程

12-2 電腦解題工具介紹
- 一、認識 Scratch
- 二、使用 Scratch 實作遊戲
- 三、Scratch 的進階練習

12-3 電腦解題程序
- 一、問題分析
- 二、解題方法設計
- 三、測試與修正

12-1 電腦解題簡介

- 解決問題是一個過程，這個過程會對問題進行分析，找出解決的方案。由於資訊科技的進步，電腦已成爲解決問題的重要工具

12-1 電腦解題簡介

為什麼要運用電腦來解題？

1. 運算速度快

   可應用於解決需要大量運算的問題

2. 記憶容量大

   可記錄與解決問題相關的大量資料

3. 正確性高

   相同的條件下，反覆執行，結果都是一樣的

4. 可重複不停地作業

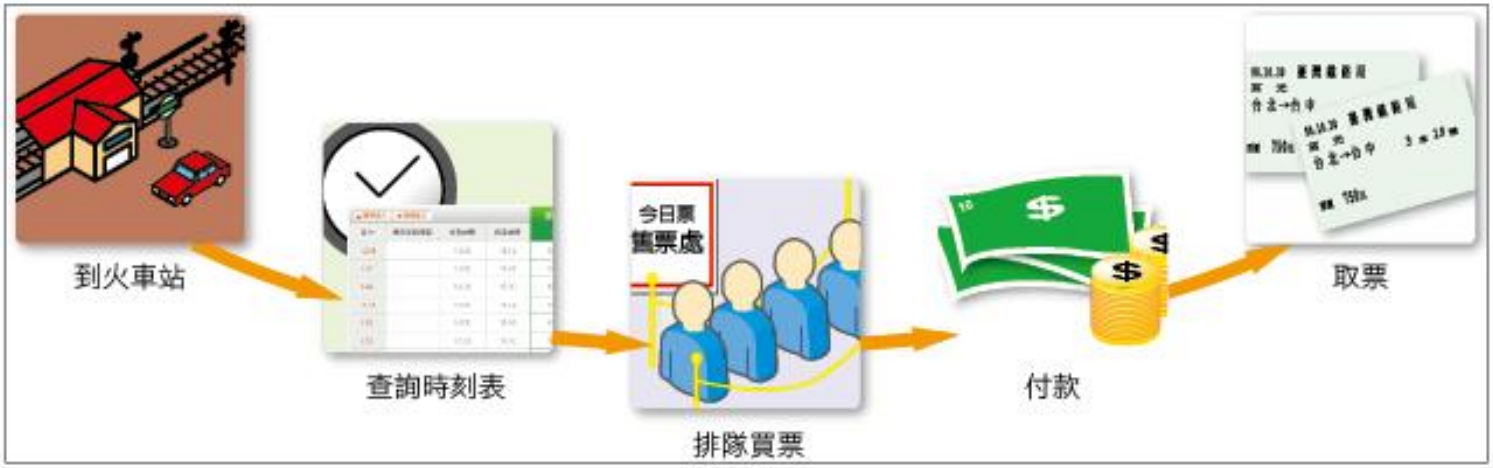
   可以不間斷地運作

5. 具備通訊能力

   透過網路，可以隨時和遠方的電腦進行通訊

12-1 電腦解題簡介

- 以火車購票爲例
  - 傳統排隊購買的方式中，旅客出發前需要事先到火車站，查詢好時刻表後，排隊付款買票；預訂他日的車票，更需要事先專程到火車站購票，十分不方便



(A) 傳統排隊買票

12-1 電腦解題簡介

- 以火車購票爲例
  - 電話訂票只能前進或後退的操作模式，與輸入相關資料的方式，都不方便，且旅客仍需提前到車站或郵局付款取票



(B) 電話語音訂票

12-1 電腦解題簡介

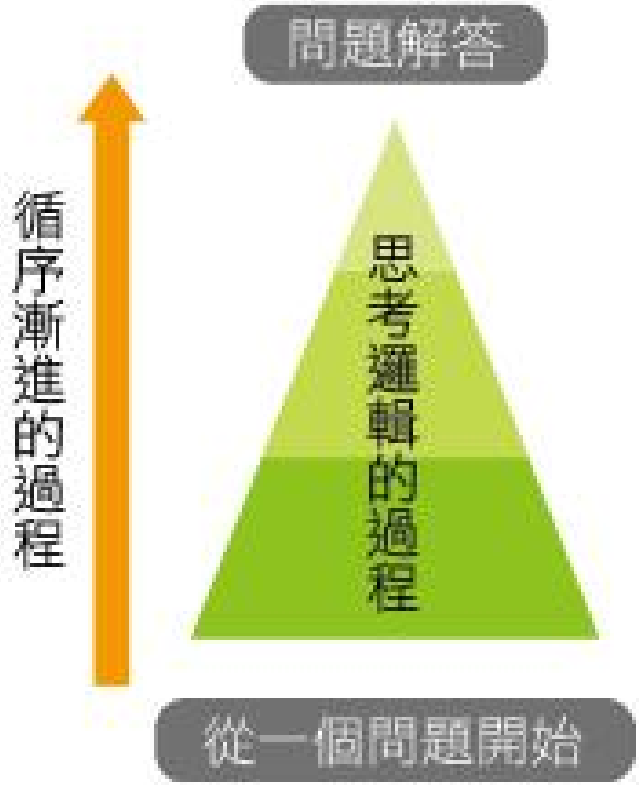
- 以火車購票爲例
  - 架設火車購票的網路系統，讓使用者隨時隨地透過網頁，直接線上完成訂票與取消訂票等作業。由於電腦速度快，可以縮短作業的時間；也可以全年無休一天二十四小時爲旅客服務



(C) 網路訂票

一、垂直式思考 (Vertical Thinking)

- 思考問題的答案時，從問題本身出發，依按部就班、循序漸進的邏輯思考方式，不斷往下垂直探索問題答案，推論過程中的每一個步驟，都須嚴格定義及推論正確，直到獲得問題解答爲止



一、垂直式思考 (Vertical Thinking)

- 垂直式思考的特性
  1. 尋找正確的方法，並且要求每一個執行步驟都是正確的
  2. 會排除不可能的解題方法；也會排除與解題無關的因素
  3. 重複解題時，每次得到的答案都是一樣的
  4. 每次都會探索最可能的路徑
  5. 解題會在有限的步驟內完成

一、垂直式思考 (Vertical Thinking)

- 垂直式思考是電腦解題的精神之一，電腦解題大都採用垂直式思考的方式
- 例如使用應用軟體製作班級成績單，思考如何計算排名時：

1. 先計算每個人的總分
   可使用試算表函數或直接計算的方式
2. 再依總分進行排序
   繼續往下垂直思考排序的方法，會想到使用試算表函數或排序功能，計算出排名

一、垂直式思考 (Vertical Thinking)

旅遊路線規畫與費用估算

1. 規畫路線時，先要決定旅遊的地點與交通工具，然後再決定路線
2. 若要估算搭乘高鐵的交通費，要
   (1) 知道需要購票的大人、小孩的人數
   (2) 選擇班次，決定折扣，最後才能計算出搭乘的費用
   (3) 檢核是否符合預算的要求



旅遊路線的規劃是垂直式思考的一種

一、垂直式思考 (Vertical Thinking)

- 與垂直式思考相對應的是水平式思考 (Lateral Thinking)，它是將思考解決問題的思路，從問題本身向各個方向發散，各指向不同的答案。
- 常用的方法有腦力激盪、自由聯想法等，一般常應用於創造力思考

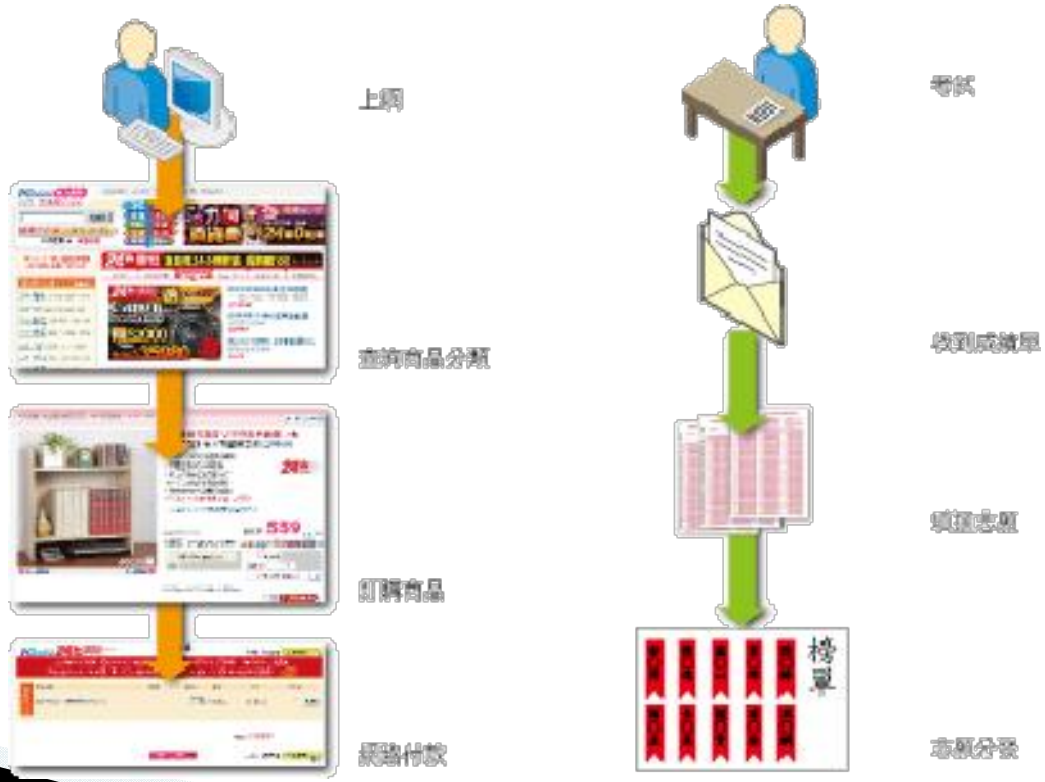


練習

- T 形火車思考問題

二、循序漸進的流程

- 每個步驟依照一定的順序逐步執行，當所有步驟執行完後，就一定會獲得正確的解答，問題也會獲得解決



二、循序漸進的流程

- 實例 1 農夫渡河

  一位農夫帶著一匹狼、一包米、一隻雞要過河，小船一次只能載一個物品。請問農夫應該以何種次序運送？才能將三者安全渡河？
- 限制

  如果農夫不在，狼會吃雞，雞會吃米。

二、循序漸進的流程

- 農夫渡河-解題方法



二、循序漸進的流程

- 實例 2　士兵渡河

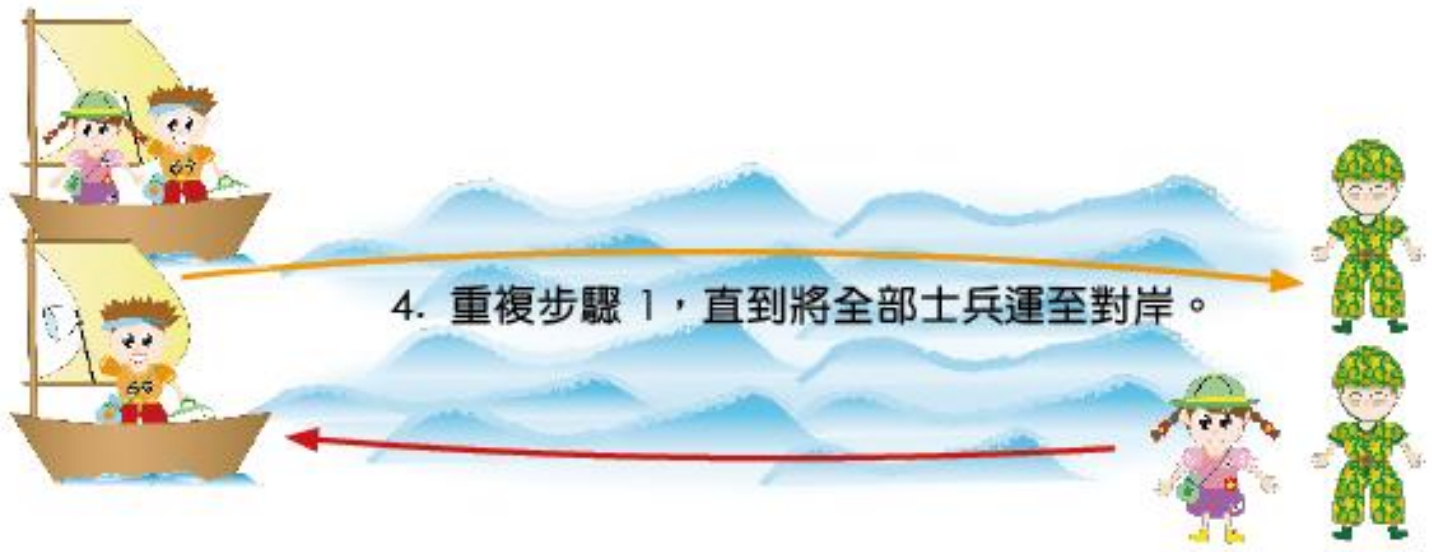
  一隊士兵想要渡河，河邊有一小舟，舟上有兩位小孩，小舟載運量有限制。請問該如何運送，才能讓整隊士兵全部安全渡河？

- 限制

  小舟每次只能載兩個小孩或一個士兵。

二、循序漸進的流程

- 士兵渡河-解題方法



二、循序漸進的流程

- 電腦解題過程強調按部就班、循序漸進的邏輯思考，在流程中，常常有許多 Yes/No 的邏輯決策需要判斷，唯有依照邏輯思考正確判斷，才能正確解決問題
- 電腦解題的決策判斷主要包含 AND, OR, NOT 三種

邏輯判斷	意義	例子	結果	說明
NOT	非	NOT True NOT False	False True	
AND	且	True AND True True AND False False AND True False AND False	True False False False	必須兩者皆為 True，結果才為 True。
OR	或	True OR True True OR False False OR True False OR False	True True True False	只要有一者為 True，結果便為 True。

二、循序漸進的流程

- 例如判斷 a = 8 是否符合判斷式 5 < a < 20 的邏輯思考，可依下列步驟進行

1. 將 5 < a < 20 轉換成電腦的邏輯表示法：a > 5 AND a < 20
2. 將 a = 8 代入 8 > 5 AND 8 < 20 ，所以符合判斷式。

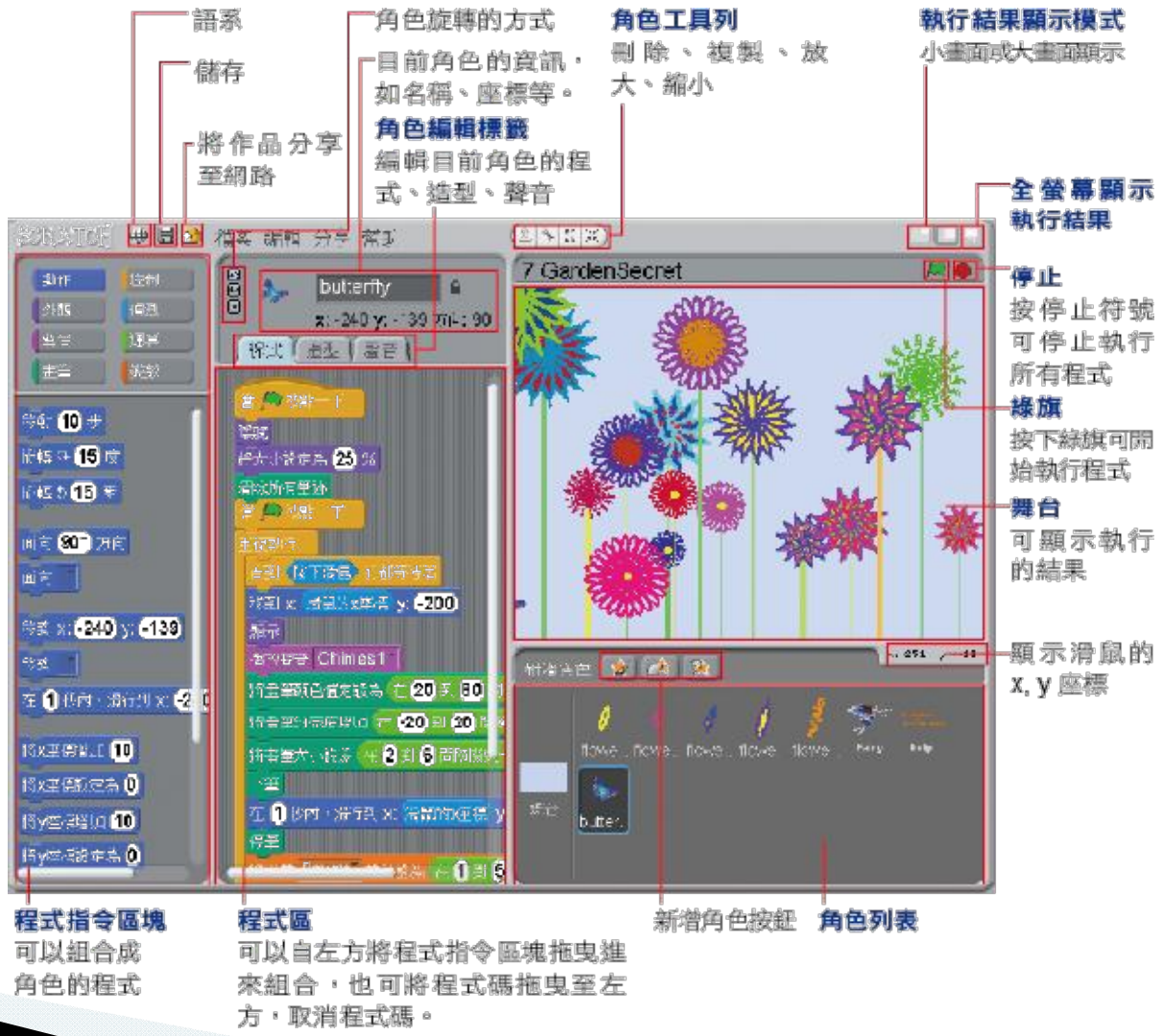
True AND True
True

12-2 電腦解題工具介紹

- Scratch 系統是由美國麻省理工學院研發的免費軟體。使用者不需背誦程式語言的指令，使用滑鼠點選、拖曳的方式就可以組合程式

12-2 電腦解題工具介紹

- Scratch 的介面



二、使用 Scratch 實作遊戲

觀看遊戲設計的軟體操作影片

使用 Scratch 軟體製作動畫—讓小貓咪走動

一、認識 Scratch

- 執行步驟

1 至 http://scratch.mit.edu 下載並安裝程式。



2 安裝完成後，按下 Scratch 圖示，即可執行。





一、認識 Scratch

- 開啓與執行範例

按「綠旗」按鈕 ，就可以執行開啟的專案。



一、認識 Scratch

- 角色介紹

5 為了要讓貓咪產生能往前走動的效果，將



拖曳至程式區。

6 加入聲音：

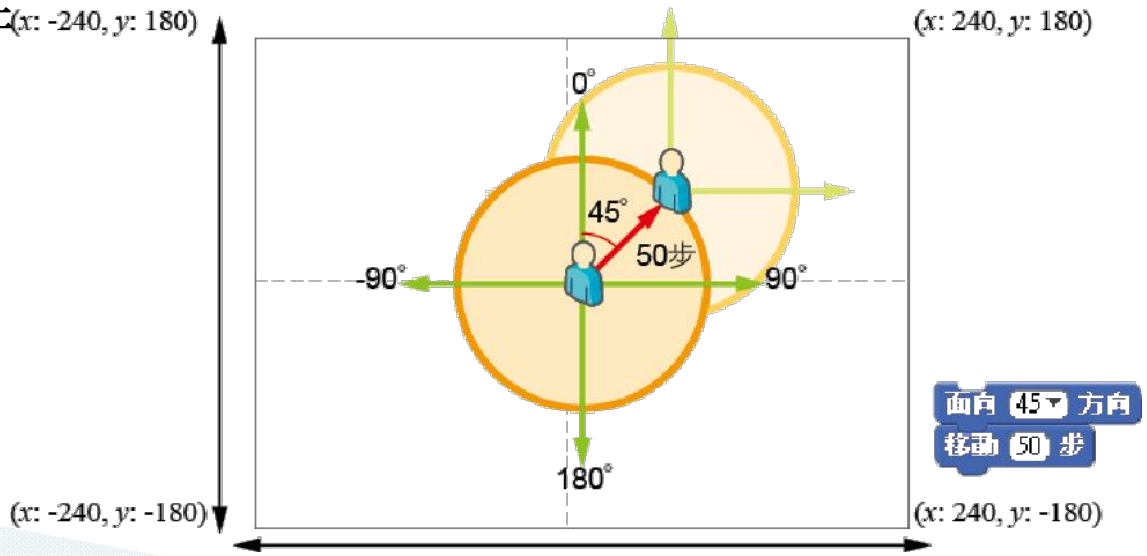


按綠旗 執行程式，儲存檔案即可。



一、認識 Scratch

- 在 Scratch 中，舞台是讓所設計的故事、遊戲、以及動畫表演的場所，角色可以在舞台上移動，或和其他角色互動。舞台的寬爲 480，高爲 360 單位，並以 x-y 的格線分割，中央位置的 x, y 座標爲 0, 0。另外控制角色時，也常使用角度和距離



二、使用 Scratch 實作遊戲

- 使用 Scratch 設計一個乒乓球遊戲

  建立一顆球及一支球拍，球可自由跳動，玩家可使用滑鼠操作球拍反擊球，讓球不斷反彈。若球掉落至舞台下方紅色區域，結束遊戲

二、使用 Scratch 實作遊戲

觀看遊戲設計的軟體操作影片

使用 Scratch 軟體設計遊戲—乒乓球遊戲 (一)

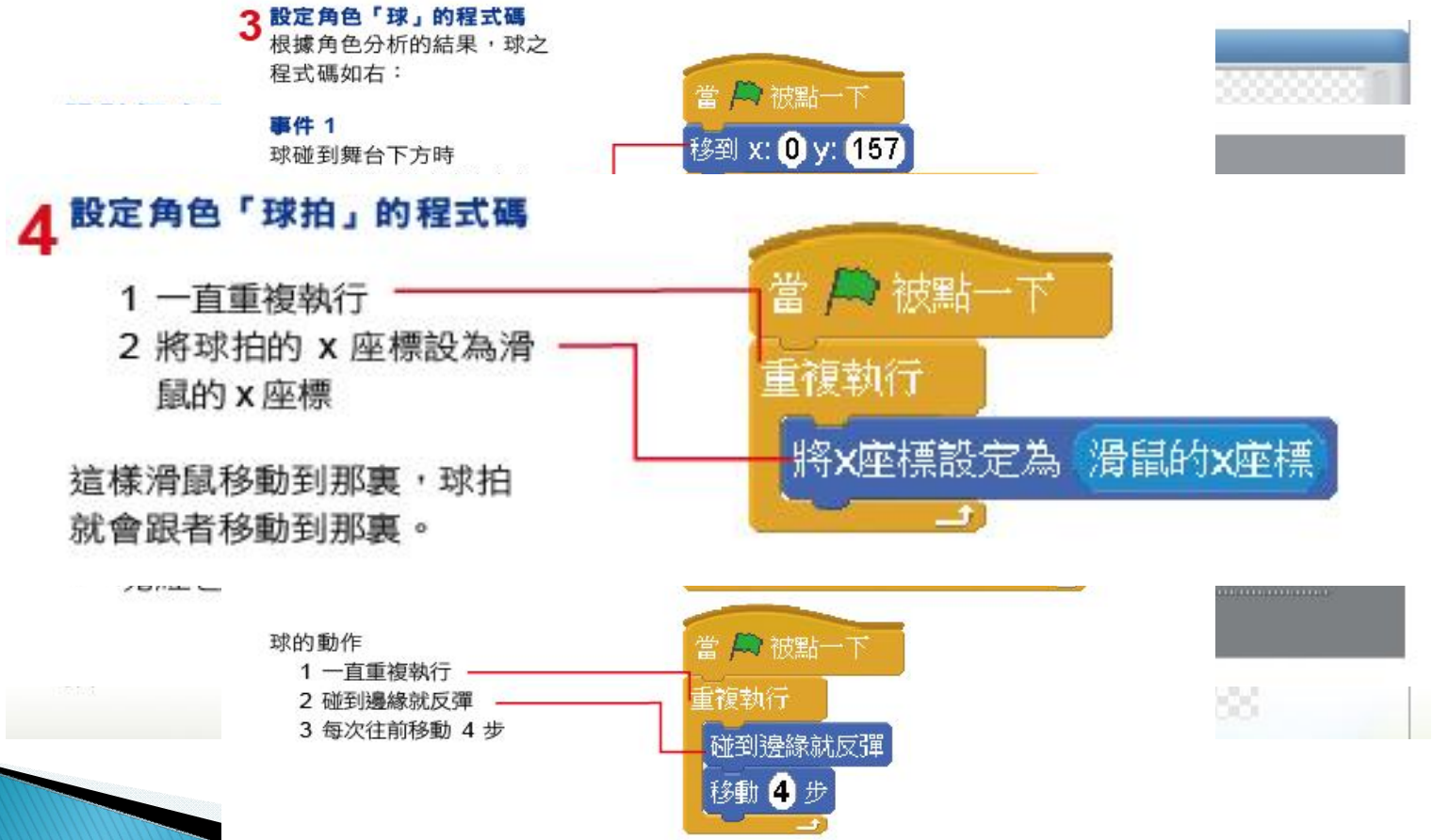
二、使用 Scratch 實作遊戲

- 角色分析
  - 在開始設計遊戲之前，先分析此遊戲的各個角色的動作，及其在不同事件下所採取的不同動作

角 色	事 件	動 作
球		重複移動
	1. 當球碰到舞台下方紅色區域	遊戲結束
	2. 球碰到球拍時	將球反彈回去 (需決定反彈的方向與位置，隨機改變方向)
球 拍		玩家可以使用滑鼠移動球拍

二、使用 Scratch 實作遊戲

- 設計步驟



三、Scratch 的進階練習

- 接續前面的實作練習，將乒乓球的遊戲增加以下功能：
  1. 計分：球拍每擊到一次球，便得 1 分，並在畫面顯示累積的分數
  2. 加速：得分超過 5 分時，加快球的速度

三、Scratch 的進階練習

觀看遊戲設計的軟體操作影片

使用 Scratch 軟體設計遊戲

乒乓球遊戲計分與加速 (二)

三、Scratch 的進階練習

- 乒乓球實作進階練習

2 點選



讓每次遊戲
都從 0 開始

3 點選



因為只有球
分，所以將
至適當的位

6 完整的程式碼如右圖



12-3 電腦解題程序

- 電腦解題的程序包含問題分析、解題方法設計、測試與修正三個步驟。

一、問題分析

- 輸入規範 (Specifications of Input)

  輸入的內容、格式、控制條件、範圍等
- 輸出規範 (Specifications of Output)

  輸出的資料規範，通常和輸入值有關
- 輸入與輸出對應關係

  將輸入轉化爲輸出的處理方式，常使用 IPO (Input Process Output) 圖進行分析說明，IPO 圖常用以描述輸入資料、對資料進行的處理、和輸出資料之間的關係
- 邊界條件

  探討問題在特殊的輸入條件時，可能產生特殊輸出的例子

一、問題分析

問 題

設計一個電腦程序，讓使用者輸入身高和體重後，可依右表自動判定身體質量的等級。

其中，$BMI = \frac{體重\ (kg)}{身高^2\ (m^2)}$

分級	身體質量指數
體重過輕	BMI < 18.5
正常範圍	18.5 ≤ BMI < 24
體重過重	24 ≤ BMI < 27
輕度肥胖	27 ≤ BMI < 30
中度肥胖	30 ≤ BMI < 35
重度肥胖	35 ≤ BMI

分析問題

輸入規範

輸入的內容	格式	控制條件	範　圍
體重	整數	單位 (kg)	20 ≤ 體重 ≤ 120
身高	整數	單位 (cm)	60 ≤ 身高 ≤ 200

輸出規範

輸出的內容	格式
身體質量的等級	文字

輸入與輸出對應關係

身高和體重 → BMI計算
決定BMI分級 → 身體質量的等級

邊界條件

身高輸入 0 時，BMI 的計算會產生錯誤，應在輸入規範中就要排除。

二、解題方法設計

- 電腦解題時，需針對待解決之問題，提出可能的解決方法，並經由實際執行，驗證解題方法

解題方法

可使用循序漸進、按步就班的方式解決此問題，構思其解題的程序，應依序包含下列步驟：

1. 根據輸入資料，計算出 BMI 值。

2. 再根據 BMI 值，判斷其身體質量等級。

例如若某位同學身高 170 cm、體重 70 kg，則

1. BMI 計算

$$\text{BMI} = \frac{\text{體重 (kg)}}{\text{身高}^2\ (\text{m}^2)} = \frac{70}{1.7^2} = 24$$

2. 決定 BMI 分級

根據 BMI = 24，查詢對應的分級，其中 $24 \le \text{BMI} < 27$ 屬於體重過重類型，所以輸出為「體重過重」。

二、解題方法設計

- Scratch解題步驟



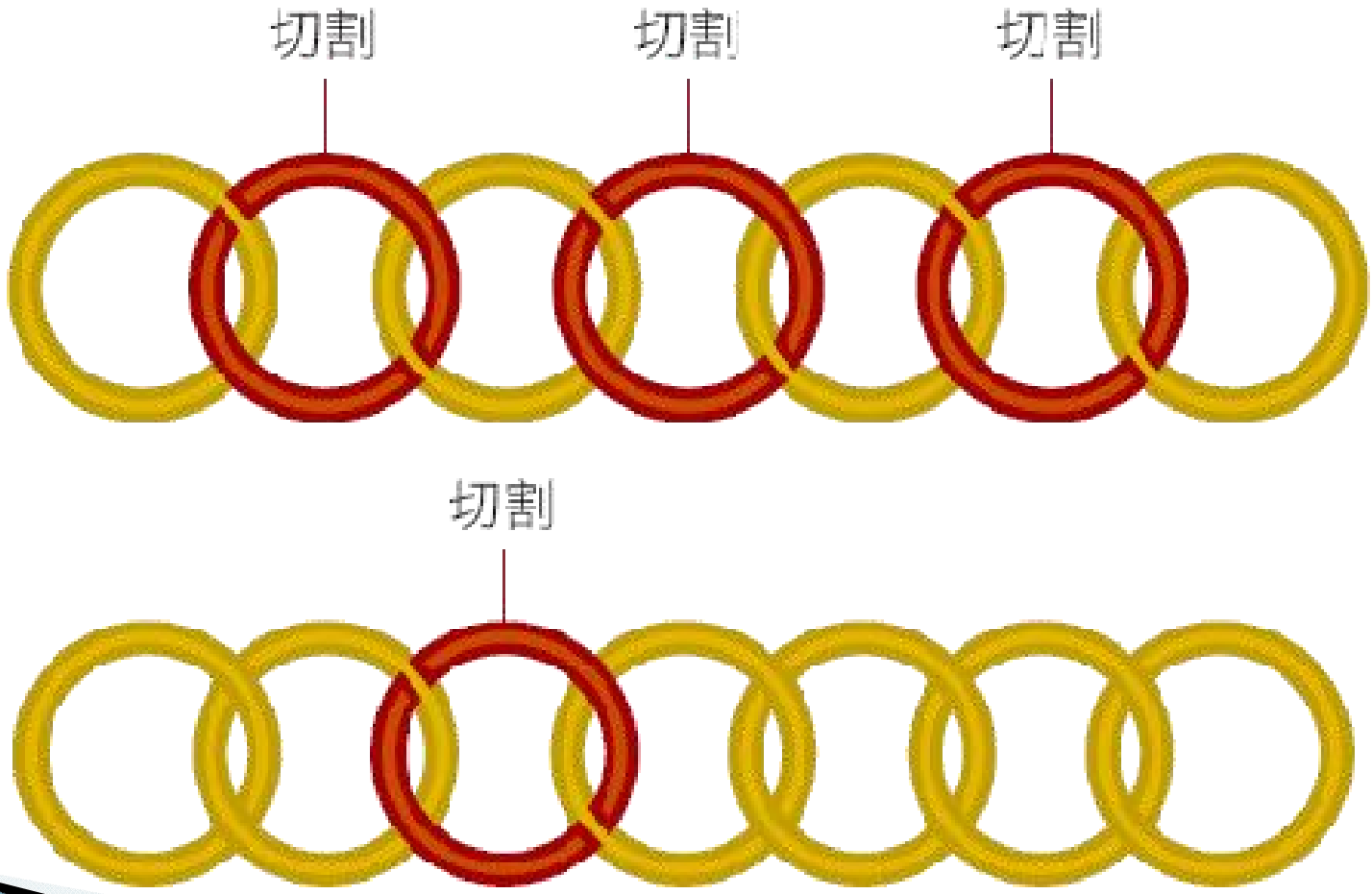
三、測試與修正

- 程式上的錯誤常被稱爲 Bug，除錯 (Debug) 則是找出並清除錯誤的過程
- 實例

  有位富商帶 7 個相連的金環，到旅店住宿 7 天，一天住宿的費用是一個金環。在切割最少金環的條件下，每天付一個金環給旅店。該如何切割金環，才能完成需求？



三、測試與修正



天數	旅店的金環	富商的金環
1		
2		
3		
4		
5		
6		
7		

練習

- 軟體測試模擬
- 撲克牌排序方法

三、測試與修正

錯誤的類型

- 語法錯誤 (Syntax Error)

  指程式含有不符合語法的敘述，無法讓編譯程式正確翻譯。

  例如程式正確指令爲 if，輸入成 If

- 語意錯誤 (Semantic Error)

  又稱邏輯錯誤 (Logical Error)，發生在程式的語法正確，程式仍會順利執行，但執行結果卻不正確

三、測試與修正

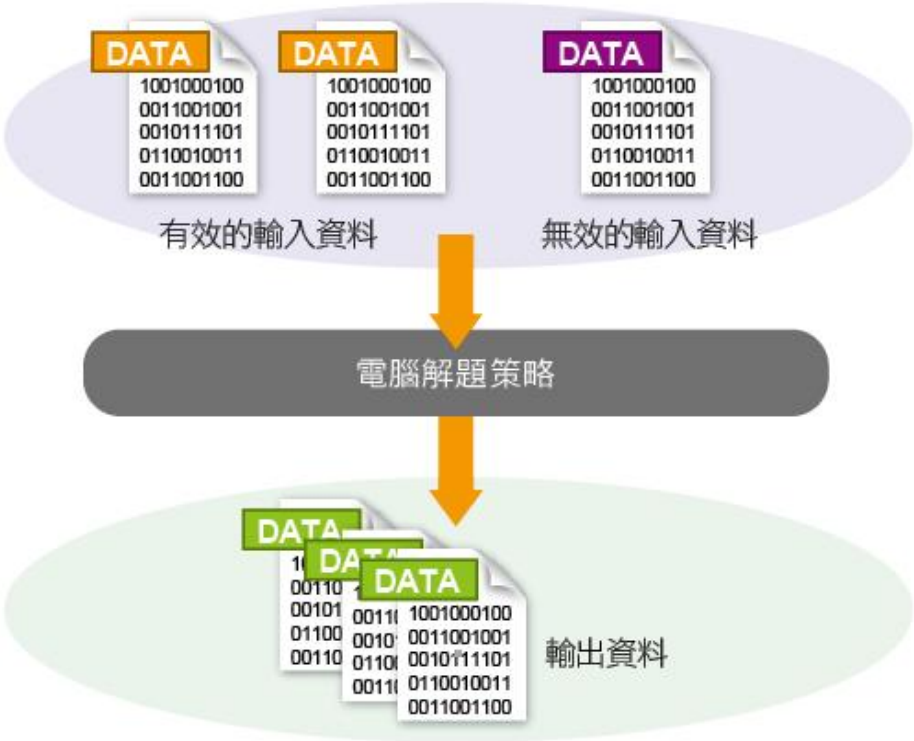
- 實例

解題方法	正 確	語意錯誤
攝氏與華氏溫度轉換 (華氏 = 100º)	攝氏 = (華氏 – 32)×5 / 9 (100–32)×5/9 = 37.78	攝氏 = 華氏 – 32×5 / 9 100 – 32×5 / 9 = 82.22
80 < x < 90 (若 x = 75 則 80<75<90 爲假)	80 < x AND x < 90 80 < 75 AND 75 < 90 假	80 < x OR x < 90 80 < 75 OR 75 < 90 真

三、測試與修正

- 測試的方法

  爲了要確保解題方法的正確性，常會使用各種資料，對解題方法進行測試，這些測試資料往往是特別挑選或產生出來的，有些是有效的輸入資料，有些是無效的

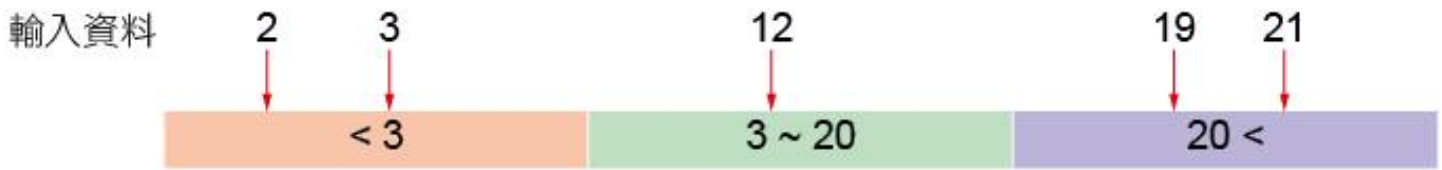


測試資料的分類

三、測試與修正

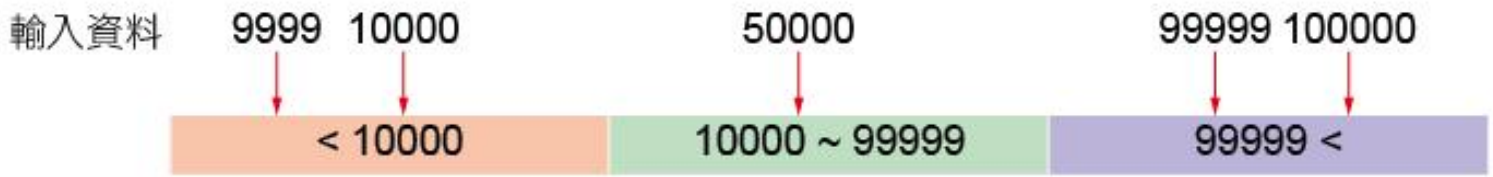
- 測試的方法

  測試資料的選擇應考慮邊界條件，如下圖，輸入資料是 3~20 的整數，有效的測試資料可以選擇邊界值 3 和 20，無效的測試值可以選擇 2 和 21



三、測試與修正

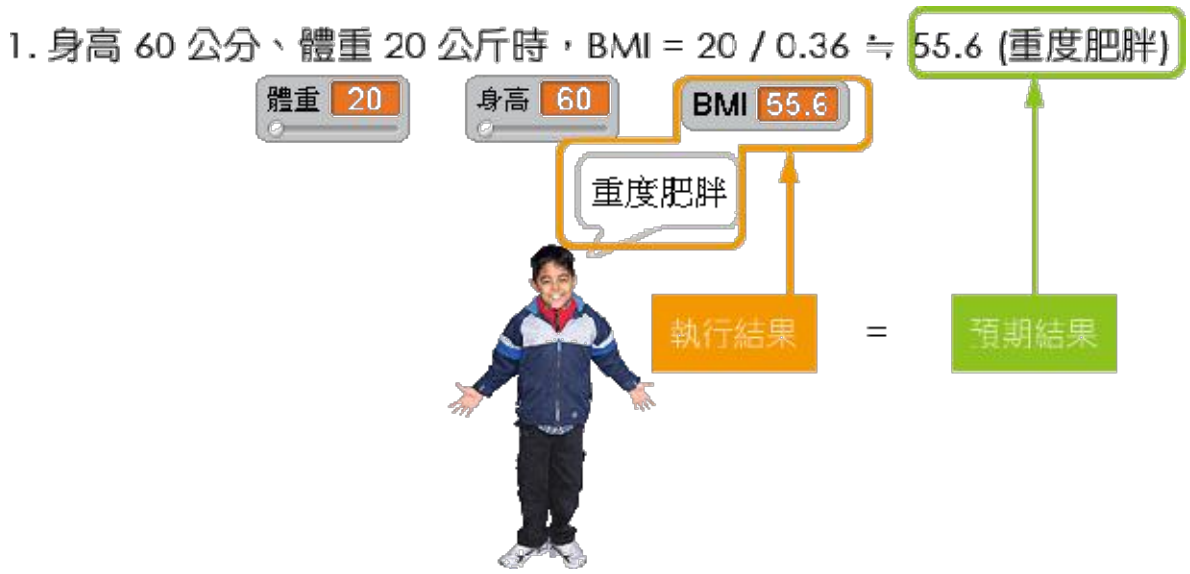
- 測試的方法
  同理，輸入資料是 10000~99999 的整數時，有效的測試資料可以選擇邊界值 10000 和 99999，無效的測試值則可以選擇 9999 和 100000；兩者也可分別選擇有效測試值的中間值 12 和 50000 分別進行測試



三、測試與修正

- 測試的方法

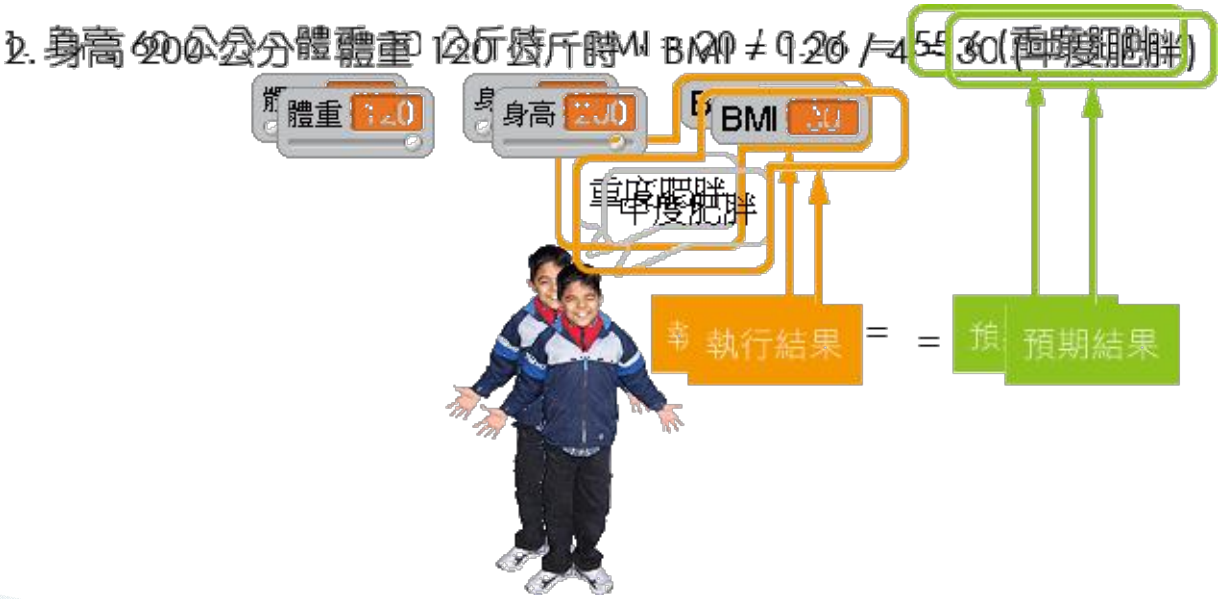
  以測試邊界條件爲例，選擇兩組測試資料，分別計算出其身體質量等級，再檢查解題工具執行結果是否正確



三、測試與修正

- 測試的方法

  以測試邊界條件爲例，選擇兩組測試資料，分別計算出其身體質量等級，再檢查解題工具執行結果是否正確



練習

- 求自己成績在全班的排名
- 溫度單位轉換

本章內容結束