數學思維與運算思維的火花：多種進路的教學設計

文／吳宛柔

前言

因應科技發展，數位時代為學生帶來新的學習方式。PISA 國際評量計畫（The Programme for International Student Assessment, 簡稱 PISA）將數位時代的學習（Leaning in the Digital World, 簡稱 LDW）定義為：「the capacity to engage in an iterative and self-regulated process of knowledge building and problem solving using computational tools and practices（OECD, 2023, p.10）」，即利用電腦運算工具並透過迭代和自我調節歷程以進行知識建構與問題解決的能力，此能力亦可與運算思維（Computational thinking）相呼應。若我們將數學概念或知識應用於解決問題時，亦可視為一種數學思維（Mathematical thinking），這說明數學思維與運算思維共同觀點皆是問題解決的能力。

壹、數學思維與運算思維間的關係

數學思維與運算思維間存在何種關係呢？ Pérez (2018) 指出數學思維和運算思維的關係，可視為兩種取向：一是仍以數學為主的內導取向（inward orientation），二是藉由運算思維連結數學的外導取向（outward orientation）。前者仍以理解數學結構與本質為主，後者則著重數學與其它領域的連結。由於前者以運算思維的應用加強學生數學概念的理解與發展，培養學生對數學結構的掌握，又基於後者強調數學與其他領域的應用與整合，使學生將數學知識應用到其他領域中，Pérez（2018）進一步主張數學思維與運算思維具有互補關係。除了 Pérez（2018）主張的兩個取向外，Rambally（2016）提出數學思維和運算思維之間還存在著互惠關係（synergistic）。

貳、多種進路的教學設計

一、外導取向的教學設計–以 Scratch 繪製加號符號為例

外導取向的教學設計，以「應用數學思維」的學習為主。例如：學生應用他們的數學概念做為先備知識來撰寫並完成 Scratch 程式。學生在 Scratch 任務中不僅學會如何進行程式撰寫，亦運用數學概念培養運算思維技能，有助於學生更全面地理解和應用所學的知識。

(一) Scratch 任務說明

請用 Scratch 小貓咪動態畫出一個由五個大小相同正方形組成的加號形狀，如圖 1 所示。

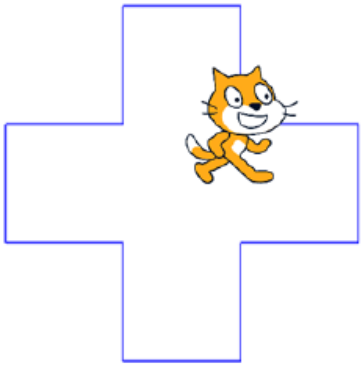

圖 1、由五個大小相同正方形組成的加號形狀（Milicic, Wetzel, and Ludwig, 2020）

(二) 數學概念做為先備知識

為完成任務，學生需先應用正方形的定義，即加號形狀中每個角度都是 90 度，每邊的邊長都等長。當學生分析完加號形狀時，可考慮使用對稱性、平行與垂直的性質確保每個正方形的位置，並使用 Scratch 的「定位積木」將小貓咪移到畫面中指定的位置。

運算思維技能

1. **拆解**（Decomposition）：為了實現由五個相同大小的正方形組成的加號形狀，學生需要拆解問題，將整體分解為更小的部分，包含如何創建單個正方形的程式碼、設定它們的位置、控制它們的旋轉等。

2. **規律辨識**（Pattern Recognition）：通過規律辨識，學生注意到加號形狀是由五個相同大小的正方形按照一定的模式排列而成。這種規律性使得他們能夠更容易地設計一個能夠重複應用的演算法，而不僅是處理單個正方形。

3. **抽象化**（Abstraction）：學生將具體正方形的創建和排列過程抽象化成通用的步驟和模式。

4. **演算法**（Algorithm Design）：當學生理解並分析完上述步驟，即可設計一個演算法（圖 2 為其中一種寫法）來完成任務。

二、內導取向的教學設計–以 Scratch 繪製正方形為例

內導取向的教學設計，以「發展與深化數學思維」的學習為主。例如：學生應用運算思維技能（拆解、規律辨識、抽象化、演算法）以解決數學問題。程式撰寫的過程為學生的數學知識和認知提供了學習機會，包括問題解決、測量、幾何與空間概念（Savard and Highfield, 2015）。下列任務與外導取向任務不同之處在於，任務對象適用於對正方形概念尚需釐清的學生，學生繪製正方形時，討論正方形的性質，並利用操作程式測試和量測長度、角度，且進一步辨識重複圖形重複的模式。

圖 2、參考的 Scratch 程式碼–以 Scratch 繪製加號符號為例

（一）Scratch 任務說明

請用 Scratch 小貓咪畫一個正方形，如圖 3 所示。

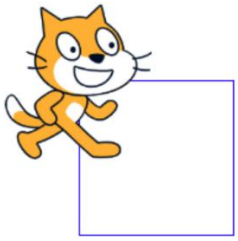

圖 3、畫一個正方形（Miller, 2019）

（二）運算思維技能

1. 拆解（Decomposition）：為了完成任務，學生可先從設置起點、繪製一條邊、旋轉等步驟依序處理每個步驟，降低任務的複雜度。

2. 規律辨識（Pattern Recognition）：規律辨識幫助學生理解正方形的特性，例如，每條邊的長度相等，每次旋轉 90 度。辨識數學中的模式對於發現規則、定義變數和建立數學模型是重要的。

3. 抽象化（Abstraction）：學生將繪製正方形的具體步驟抽象為通用的程式碼，使用 Scratch 積木來表示畫一條邊、旋轉的操作。抽象思維有助於將數學問題轉換為一組操作和規則，使問題更易於理解和解決。

4. 演算法（Algorithm Design）：設計一個演算法（圖 4 為

圖 4、參考的 Scratch 程式碼–以 Scratch 繪製正方形為例

其中一種寫法）將上述步驟組合起來。訓練演算法思維有助於學生在解決數學問題時有組織、有序地進行推理和操作。

（三）探索中學習

1.　無法畫出一個正方形

學生弄錯每次的旋轉角度（例如：20 度）或旋轉方向，如圖 5 所示。

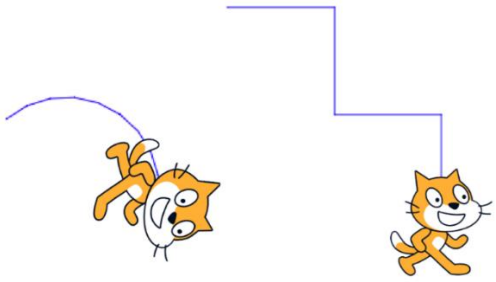

圖 5、學生弄錯每次的旋轉角度或旋轉方向（Miller, 2019）

2.　這樣算是正方形嗎？

學生正確寫出程式碼並畫出圖形，但因 Scratch 小貓咪一開始不是面朝 90 度，而是面朝 45 度，故繪製出旋轉的正方形，學生不確定該圖形是否為正方形，如圖 6 所示。

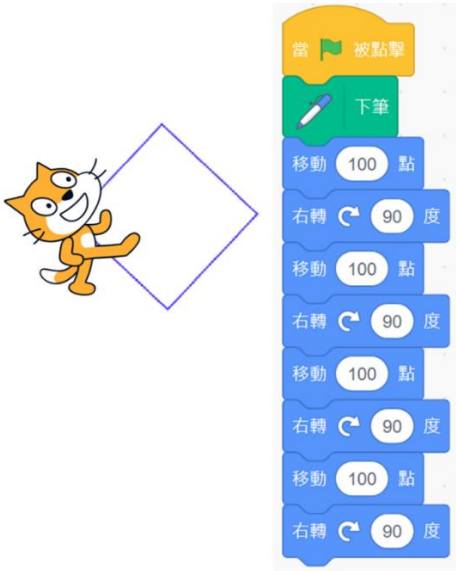

圖 6、這樣算是正方形嗎（Miller, 2019）

1.　思考教師的提問並與同儕討論，以加深數學概念的理解：

問題一：如何讓 Scratch 小貓咪移動？又要移動多長呢？

問題二：如何讓 Scratch 小貓咪旋轉？又要旋轉幾度呢？

問題三：Scratch 小貓咪要旋轉幾圈才能完成任務？

問題四：要使用什麼數學概念來完成這個任務？

當學生討論上述問題時，他們連結程式碼和正方形結構間關係，他們能繪製屬於自己的正方形，並加深正方形的概念性了解，即四邊等長且四個角都是直角的四邊形稱為正方形。

一、互惠關係的教學設計–以 Scratch 繪製幾何圖形為例

互惠關係的教學設計，以「鞏固學生數學思維和培養學生運算思維」的學習為主。例如：在教學任務的設計上，程式的編碼能加深數學概念的學習，且數學亦能進行程式偵錯。下列將以一系列改編自 Sung 和 Black（2020）的任務做說明（表 1）：

表 1. 互惠關係的教學設計–以 Scratch 繪製幾何圖形為例

任務	內容說明
A	參考「Scratch 程式積木的說明」與「畫一個正方形的程式碼」，用 Scratch 小貓咪畫一個正三角形。
B	用 Scratch 的「重複積木」簡化「畫一個正方形的程式碼」
C	對「正五芒星圖形程式碼」進行除錯。

（一）任務 A：用 Scratch 畫一個正三角形

參考「Scratch 程式積木的說明」與「畫一個正方形的程式碼」，用 Scratch 小貓咪畫一個正三角形。

閱讀程式碼是學習撰寫程式的一部分，學生若能先讀懂、理解程式碼的內容，並進一步對程式碼進行修改，可促進學生運算思維的學習。任務 A 提供學生「Scratch 程式積木的說明（表 2）」與「畫一個正方形的程式碼（圖 4）」，讓學生參考仿照以完成圖 7 所示的正三角形。學生不僅要讀懂正方形的程式碼、將其修改為三角形的程式碼（圖 8），又程式碼的內容與撰寫涉及數學概念，除了幫助學生運算思維與程式撰寫的技能，亦能鞏固學生的數學概念。

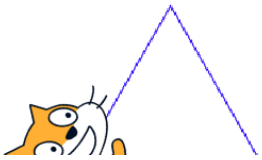

圖 7、Scratch 小貓咪畫一個正三角形

圖 8、Scratch 小貓咪畫一個正三角的程式碼

表 2. Scratch 程式積木說明

積木類型	積木外觀	積木功能
事件類		當綠旗被點擊後，開始執行下方的指令積木。
畫筆		使用角色來畫圖
動作類		將角色移動到指定的位置
		將角色從目前的角度，向右旋轉指定的角度

（二）任務 B：用 Scratch 的「重複積木」簡化「畫一個正方形的程式碼」

執行程式時，若程式碼中某些指令重複執行多次，可使用「重複結構」（例如：Scratch 的「重複積木」）來簡化程式碼，讓程式更容易閱讀與理解。任務 B 即是請學生使用「重複積木」來簡化正方形的程式碼（圖 9）。在簡化程式碼的過程中，學生對正方形性質概念的理解與正方形結構的掌握是簡化程式碼的基礎，而運算思維技能的使用則有助於反應數學思維，在此任務中，學生不僅能夠更好地理解和應用數學概念，亦能提高程式撰寫的技能與解決問題的能力。

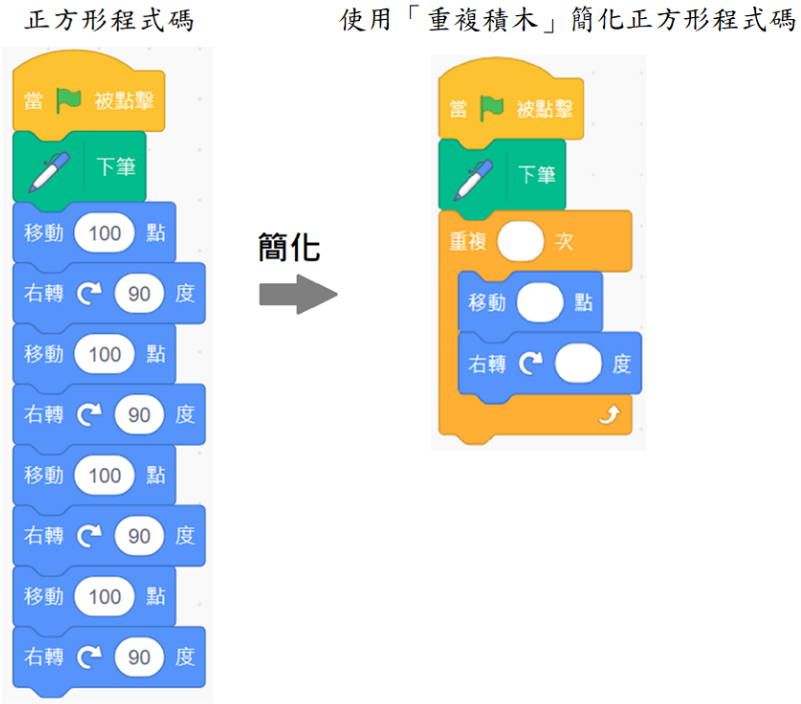

圖 9、使用「重複積木」來簡化正方形的程式碼

（三）任務 C：對「正五芒星圖形程式碼」進行除錯

程式除錯（Debug）是指尋找、識別和修正程式碼錯誤的過程（McCauley, 2008），這些錯誤可能包括語法錯誤、邏輯錯誤、執行程式碼錯誤等。為確保程式碼能順利執行，程式除錯是學習程式撰寫不可或缺的一部分。任務 C 即是請學生針對「正五芒星圖形程式碼」進行除錯（圖 10）。在程式除錯的過程中，學生對正五芒星圖形相關數學概念的理解是解題關鍵，其中涉及到角度與長度的應用，而運算思維幫助學生觀察變量值以找到可能的錯誤，學生同時運用數學思維和運算思維，修正代碼中的錯誤，並提高他們對數學概念的理解和應用。

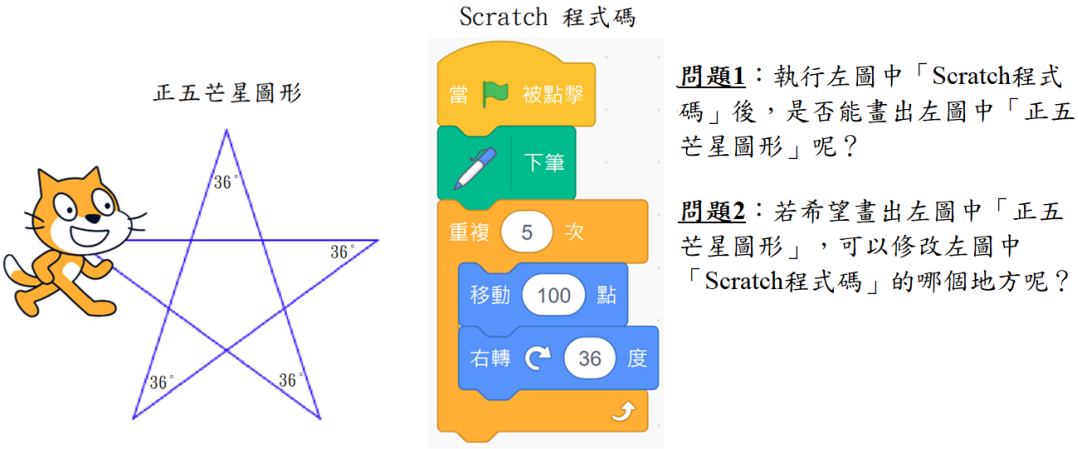

圖 10、針對「正五芒星圖形程式碼」進行除錯

108 課綱加重了電腦課的上課時數，科技融入各科課程的規劃成為新的學習趨勢，在數學思維與運算思維的關係中，無論是三種關係中的哪一種，都呼應了教育的轉變與趨勢，學習不再僅僅侷限於傳統的紙本教材，也為學生數學思維和運算思維的結合上提供了更多的可能性。

吳宛柔

臺北市立東湖國中教師

參考資料

[1] McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simon, B., Thomas, L., and Zander, C.（2008）. Debugging: a review of the literature from an educational perspective. *Computer Science Education, 18,* 67–92.

[2] Milicic, G., Wetzel, S., and Ludwig, M.(2020). Generic Tasks for Algorithms. *Future Internet, 12*（9）, 152.

[3] Miller, J. （2019）. STEM Education in the Primary Years to Support Mathematical Thinking: Using Coding to Identify Mathematical Structures and Patterns. *ZDM The International Journal on Mathematics Education, 51*（6）, 915–927.

[4] Organization for Economic Co-operation and Development （OECD)(2023). *PISA 2025 learning in the Digital world framework（second draft）*. https://www.oecd.org/media/oecdorg/satellitesites/pisa/PISA%202025%20Learning%20in%20the%20Digital%20World%20Assessment%20Framework%20-%20Second%20Draft.pdf

[5] Pérez, A.（2018）. A Framework for Computational Thinking Dispositions in Mathematics Education. *Journal for Research in Mathematics Education, 49*（4）, 424–461.

[6] Rambally, G. (2016). The Synergism of Mathematical Thinking and Computational Thinking. *In Cases on Technology Integration in Mathematics Education,* 416–437.

[7] Savard, A., & Highfi K.（2015）. Teachers' talk about robotics: Where is the mathematics? In M. Marshman, V. Geiger, & A. Bennison（Eds.）, *Proceedings of the 38th Annual Conference of the Mathematics Education Research Group of Australasia*（pp. 540–546）. Sunshine Coast: MERGA.

[8] Sung, W., J. Ahn, & Black, J. B. （2017）. Introducing Computational Thinking to Young Learners: Practicing Computational Perspectives through Embodiment in Mathematics Education. *Technology, Knowledge and Learning, 22*（3）, 443–463.