

## 以 Blockly 為基礎之視覺化運算思維測驗系統

張顧耀 王炫鈞 黃鈴玲\*

大葉大學資訊工程系

51591 彰化縣大村鄉學府路 168 號

\*lhuang@mail.dyu.edu.tw

### 摘要

運算思維是指一套以計算機可處理的模式來表達問題與解決問題的方法。隨著資訊科技的發達，世界各國都在積極推動運算思維教育，並認為所有領域的學生都應該加強學習。在運算思維的教學中，程式設計被認為是熟練運算思維觀念的最佳方式，但非資訊領域的學生多半會覺得程式語言學習起來相當困難；因此目前運算思維課程，多採用視覺化程式設計平台的積木式指令來教學，讓初學者容易理解程式概念。但常見的視覺化程式設計平台，缺乏能讓教師自行命題並做評分的測驗系統，以及班級學生成績查詢的功能，在班級與學生眾多時更為不便。因此在本論文中，我們使用 Google 提供的 Blockly 函式庫，設計並實作一個線上視覺化程式設計的測驗系統，該系統除了能讓教師設定題目的文字或圖片內容、設定標準答案所用到的程式積木及配分、學生能登入自我練習或進行測驗並由系統評分外，同時也與大葉大學的校務系統整合，能將評分結果匯入學生成績系統，對運算思維課程的授課教師提供了相當大的助益。

**關鍵詞：**運算思維，視覺化程式設計，測驗系統，Blockly

## Visual Computational Thinking Test System Based on Blockly

KU-YAW CHANG, HSUAN-CHUN WANG and LINGLING HUANG\*

*Department of Computer Science and Information Technology, Da-Yeh University*

*No. 168, University Rd., Dacun, Changhua, 51591, Taiwan, R.O.C.*

\*lhuang@mail.dyu.edu.tw

### ABSTRACT

Computational thinking involves a set of problem-solving skills related to expressing problems and solutions according to procedures that computers can execute. With the rapid development of information technology, countries worldwide are actively promoting education related to computational thinking. Students from all fields are encouraged to take computational thinking courses. Among the strategies for teaching and learning, programming is considered the best for mastering the concept of computational thinking. However, students who are not computer science majors often find programming difficult to learn. Therefore, computational thinking courses should use visual programming platforms with simple building-block instructions to allow beginners to easily understand programming concepts. The majority of visual programming platforms lack testing systems that allow



teachers to create exam questions, automatically calculate scores, and query students' grades. In this paper, we describe an online visual programming test system based on the Google Blockly library. In addition to allowing teachers to create questions and students to self-practice, take tests, and automatically receive scores, the system integrates Dayeh University's schoolwide information system. Test scores can be directly imported into the student score system, offering substantial benefits to teachers of computational thinking courses.

**Key Words:** computational thinking, visual programming, test system, Blockly

## 一、緒論

運算思維是在 2006 年由美國卡內基梅隆大學周以真 (Jeannette M. Wing) 教授首倡 [18]，她對運算思維的定義如下：運算思維 (Computational Thinking) 是一種有效運用計算機能處理的模式，來描述問題並表達解法的思維過程 [19]。周教授認為所有領域的人都應該積極學習並使用運算思維，不應只侷限在計算機領域的工作者。

近年許多國家也開始重視運算思維的教育，並陸續有相關的政策開始推行；我國在 108 學年度開始實施的十二年國民基本教育課程綱要中，對科技領域的資訊科技課程所規劃的基本理念也是以「運算思維」為主軸。希望能透過電腦相關知能的學習，培養學生邏輯思考、系統化思考等運算思維，並藉由資訊科技之設計與實作，增進學生對運算思維的應用能力、解決問題能力、團隊合作以及創新思考的能力 [2]。

早期的運算思維課程是以授程式設計為主，對於非資訊科技背景的學生來說較為困難，因此近年各級學校在此類課程的安排上，逐漸傾向於採用重視程式邏輯思維而不強調語法的模式，也就是視覺化程式設計 [17]。在教育部的推動下，大葉大學也於 106 學年度將運算思維加入大學入門課程中，並以運算思維觀念作為課程內容，先推行運算思維的廣度；在 107 學年度開始則以推行運算思維的深度為主，因此也加入視覺化程式設計作為輔助，利用視覺化的效果提升學生的興趣，讓學生學習運算思維的觀念後，再透過實作加以驗證，以視覺化程式設計作為運算思維的專業橋梁。

視覺化程式設計是指利用圖形化的元素來進行程式設計，如微軟的 Visual Programming Language，就是透過方塊和箭頭來產生程式。由於是以視覺化的方式將程式建構出來，對程式設計的初學者相當有幫助。近年有許多線上練習程式設計的平台也採用了視覺化程式設計的模式，並更加美化以提高初學者的接受程度；例如：Code.org [3] 在網站裡提供的一小時玩程式課程、Scratch 開發平台 [9]、Blockly Game [4] 與 App Inventor [8] 等，都是提供預先設定好的積木，讓學

習者拖曳堆疊出指令來控制物件及背景的變化。

由於堆積木以及能快速看到結果讓學生感覺像在玩遊戲，許多運算思維課程都採用這類平台來授課，學生也很感興趣；但在學習成效的評量上，這些受歡迎的平台並未提供測驗的功能，即便有提供小型課程的完課證明，教師也不容易取得或查詢。以大葉大學來說，由於運算思維的授課對象是 大一全部學生共計一千多人，若有統一的視覺化程式設計學習與測驗平台，除了課堂上可供學生練習，評量時提供教師出題與測驗評分功能外，再結合校務系統，讓教師可以很容易查詢到各班學生的測驗成績，對授課教師來說必定相當方便。因此本論文的目的是要利用 Google 提供的 Blockly 函式庫 [5]，實作一套符合上述需求的視覺化程式設計測驗評量系統，做為運算思維課程的學習與測驗工具，方便教師命題，以及在系統評分後取得學生測驗成績。

本論文其餘章節說明如下：第二節針對本論文相關技術與研究進行說明與介紹，第三節介紹本系統的架構與資料庫的設計與流程，第四節說明實作的結果與畫面，而結論及未來研究方向則在第五節說明。

## 二、相關研究

### (一) 視覺化程式設計

視覺化程式設計 (Visual Programming) 是以圖形化的方式來表達程式，撰寫程式時利用已設計好的方塊與連接線 (如圖 1) 當作開發程式的物件，選擇各類方塊來做為指令或動作，利用拖曳的動作連接與排序方塊，透過排序方塊的動作來設定程式碼的執行順序。由於視覺化程式設計是以視覺化的方式將程式建構出來，除了對程式設計的初學者有幫助，在其他專業領域上也有相當大的助益，如 [1] 中就針對三維製圖、影像處理和科學視覺化的 C++ 類別函式庫 VTK 設計了視覺化程式設計的介面，方便醫學或影像領域的研究人員能用來開發所需的應用程式。



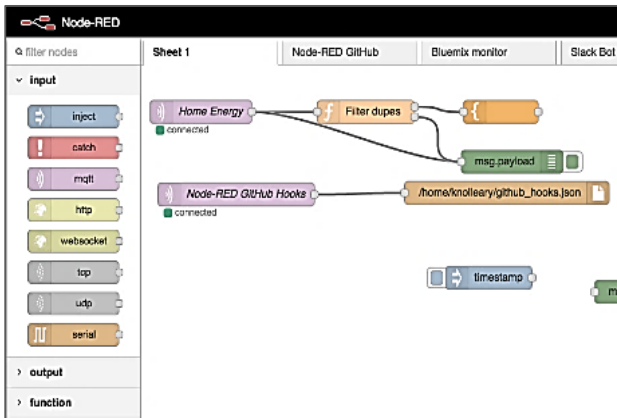


圖 1. Node-RED 利用線來連接方塊

文字式程式設計 (Textual Programming) 與視覺化程式設計的差別在於，文字程式設計過程較為制式，比較講究語法規則，容易因為遺漏記號造成執行錯誤，而視覺化的程式設計則較人性化 [12]，較能理解程式的構造與執行的順序，學生不需注意語法上的細節。例如：在文字式程式設計上需要注意語法後須加上結束符號，視覺化程式設計則不需要加入 [14]，且能透過與彩色方塊的互動上獲得更多的樂趣。

有許多學習程式設計的線上練習平台都是採以視覺化程式設計方式，例如：Code.org [3]、Scratch [9]與 Blockly games [4]等，而且都是專注在國小與國中的年齡層，大部分的課程都主打就算無程式設計經驗也可以學習，且課程時間多在一個小時內，如 Code.org 的一小時玩程式遊戲課程。透過視覺化程式設計輔助下，不僅讓年齡層不再受限制，也讓課程內容更加有趣，課程時間也不會過長。

## (二) Blockly

Blockly 是 Google 根據 Apache2.0 的條款所開發的開放原始碼專案，執行在網頁客戶端的一款 JavaScript 函式庫，目的是用來打造視覺化程式設計的開發環境 [5, 11]。

Blockly 透過拖拉與排序積木 (Block) 的方式進行程式設計，利用 Block 上類似拼圖設計的凹槽與凸點來進行連接，如圖 2 中做為 Repeat 用途的 Block 裡，數字 1 標示的凹槽用來連接前一個 Block，數字 2 標示的凸點則用來連接下一個 Block；圖 3 則是 Blockly 裡數個 Block 相連的樣子。Blockly 在程式設計時會自動判斷語法是否有錯誤，若語法錯誤將無法連接積木 [5]。使用者可以將設計完的 Block 轉出產生五種語言的程式，分別是 JavaScript、Python、PHP、Lua 與 Dart，也能儲存成 XML 的格式，在下次使用時，可以讀取上次所儲存的 XML 來還原之前所設計的 Block。除

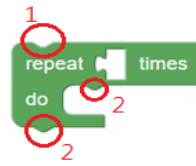


圖 2. Blockly 裡的 Repeat 的 Block

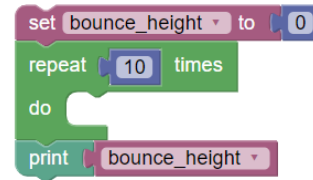


圖 3. Blockly 連接數個 block

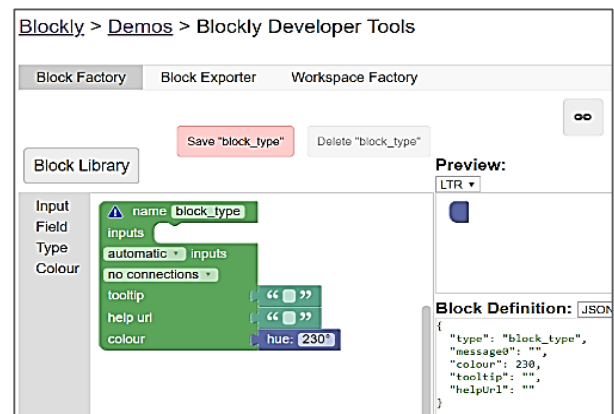


圖 4. Block Factory 是客製化方塊的線上工具

了其內建的 Block 可以使用之外，使用者可以透過 Block Factory (圖 4) 來進行設計客製化 Block[5]。

## (三) Scratch

Scratch 是由 MIT Media Lab Lifelong Kindergarten 開發的一款視覺化程式設計環境軟體，是一項開放原始碼之非商業性質使用的專案，一開始是用 Squeak 程式語言開發，在 2.0 版本改用 ActionScript 程式語言編寫，在 2019 年 1 月正式發布 3.0 版，3.0 版則是採用 HTML5 頁面技術 [7]。

Scratch 3.0 的操作介面如圖 5，主要分為四個部分：圖中左上方是舞台區可預覽結果、左下方是可選用的角色列表、中間的是可用的積木列表，右邊則是用來設計程式的腳本區。它是專門為 8 到 16 歲小孩所設計的，使用者可以透過這個平台設計出自己專屬的故事、遊戲、動畫、音樂...等等，Scratch 官網也提供了探索區 (圖 6)，讓使用者觀賞他人作品、了解程式如何設計或將自己的作品分享，並透過分享頁面下方的討論區互相交流 [7]。



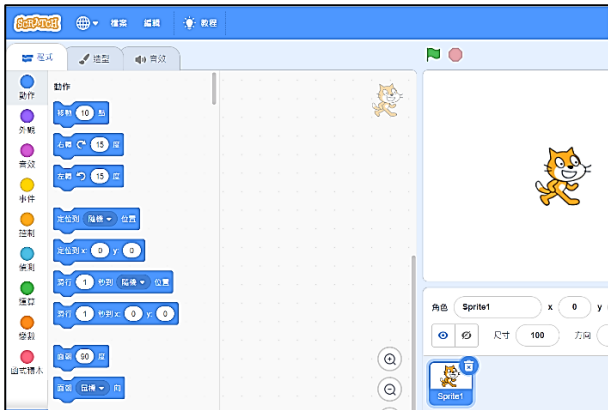


圖 5. Scratch 3.0 的介面

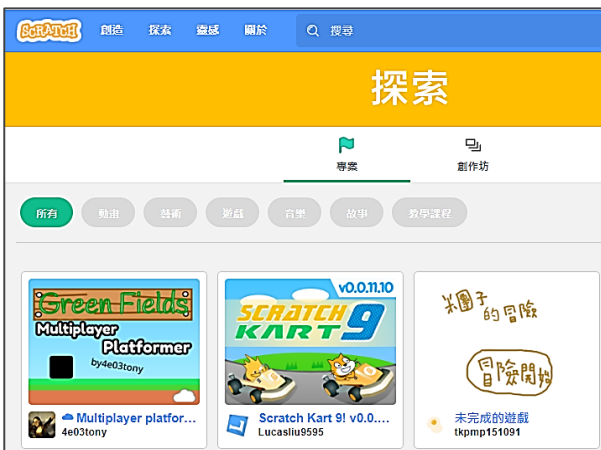


圖 6. Scratch 官網的探索區

Scratch 另外還衍生出 Snap! 語言 [10]，是以 JavaScript 編寫並在瀏覽器上執行的圖形化程式語言，除了有 Scratch 本身的功能，還能使用高階函數，而且因為跨平台的特性，程式碼很容易轉為如 Python 等其他語言的程式，也常運用在物聯網的開發過程中。

#### (四) 測驗功能

Scratch 本身並無測驗功能，網路上雖有一些兒童程式教育機構為其開發了測驗或檢定，但不易取得，可能也不適合在大學使用。Johnson 在 [6] 中使用 Python 為 Scratch 2.0 開發了 ITCH 自動測驗系統，由於 Scratch 沒有針對本地檔案的輸入輸出機制，ITCH 是使用 Scratch 專案內的狀態將測試結果嵌入到保存的 Scratch 項目，然後再解析儲存的專案文件以提取結果。此外，因 Scratch 程式本身沒有傳回值的特性，ITCH 只提供程式執行結果報告，並未進行評分。

Stahlbauer 等人在 [13] 中則針對協助師生解決在

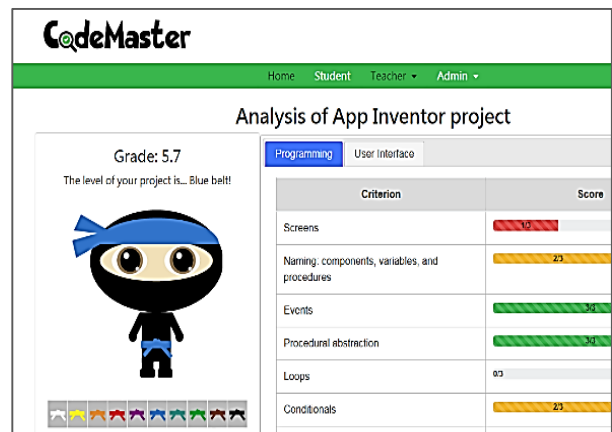


圖 7. CodeMaster 對上傳的 App Inventor 專案評分

Scratch 程式的除錯問題，開發了自動測試工具 Whisker，它能針對程式提供 property-based 的測試，也就是給定測試資料的特性後，測試工具會自動產生符合該特性的多筆資料來執行測試，再透過測試結果幫助學習者找出程式的問題。

von Wangenheim 等人在 [16] 中為 Snap! 及 App Inventor 開發了一個名為 CodeMaster 的程式評分系統 [15]，使用者可在該網站上傳程式檔進行評分，但分數高低的考量主要是看程式的複雜度，如變數命名、是否使用迴圈、條件式、字串、使用介面用到幾個顏色、幾張圖片等等。圖 7 是在該網站上傳一個 App Inventor 專案進行評分的結果，因為不是在評專案的正確性，其實不適合作為初學者的評分工具。

#### (五) 本系統採用之開發工具

針對前述 Blockly 與 Scratch 兩種開發工具，我們製作了如表 1 的功能比較表，以下簡單說明。在開放原始碼的部分，只有 Blockly 是可以進行營利，而 Blockly 還可以客製化方塊與將程式碼轉成其他語言的程式，在開發語言的部分，兩款都是以 JavaScript 所開發。基於上述的理由，故本論文選擇 Blockly 來製作本系統。

此外，由於本論文預計建置的是測驗系統網站，因此需使用 HTML、CSS、JavaScript 進行網頁開發，並使用 HTML DOM 物件模型、JSON 資料交換格式、以及 AJAX 技術等。

表 1. 兩款工具比較

	開放原始碼	客製化方塊	轉成其他語言的程式	開發語言
Blockly	可營利	有	有	JavaScript
Scratch	不可營利	無	無	JavaScript



### 三、系統分析與設計

本論文建置一個以「Blockly 為基礎之視覺化運算思維測驗系統」，幫助教師在教授運算思維的課程時，除了能使用積木式的程式平台讓學生練習外，也能自行命題評量學生的學習成效。由於全校大一學生都要上該課程，因此也希望直接整合校務系統，讓學生能用學校的帳密登入進行測驗，系統自動評分完成後直接匯入成績系統，減輕老師的負擔。以下介紹本系統的系統分析與系統設計。

#### (一) 系統分析

本系統之使用者包括學生、老師、系統管理員並且與大葉大學校務系統做結合之實際上線使用，因此本系統不考慮身分判定，其使用者案例圖如圖 8 所示。

系統中將使用者分成以下四類：

1. 學生：可以參加線上測驗與線上練習。
2. 老師：可以透過題目系統進行查詢、新增、編輯與刪除題目，考卷系統進行查詢、新增、編輯、刪除與批改考卷，作答紀錄查詢學生作答紀錄。
3. 系統管理員：擁有最大的權限，可以對所有的子系統進行新增、編輯與移除。
4. 大葉大學校務系統：透過校務系統進行使用者身分判斷。

#### (二) 系統設計

以下介紹本系統的系統架構、題目畫面設計、設計客製化方塊、設計工具箱 (Toolbox) 與作答區 (Workspace)、儲存答案流程、檢查答案流程、資料庫建置與如何透過資料集減少資料庫的負擔。

##### 1. 系統架構

本論文提出的系統架構圖如圖 9，茲將各子系統內含模組說明如下。

- (1) 測驗子系統：
  - a. 批改模組：讀取學生答案，並且自動批改考卷。
  - b. 作答查詢模組：查詢學生測驗當時所填答案。
- (2) 練習子系統：
  - a. 檢查答案模組：檢查學生所填答案並顯示結果。
- (3) 題目管理子系統
  - a. 題目查詢模組：查詢題庫內已創立的題目。
  - b. 題目設計模組：設計新的題目。
  - c. 題目更新模組：更新題目。
  - d. 題目刪除模組：刪除題目。

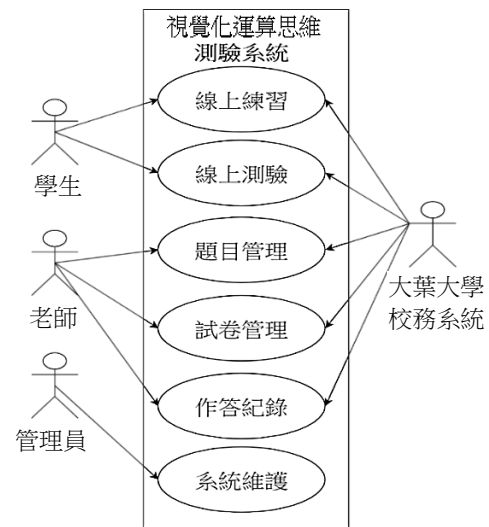


圖 8. 本系統之使用者案例

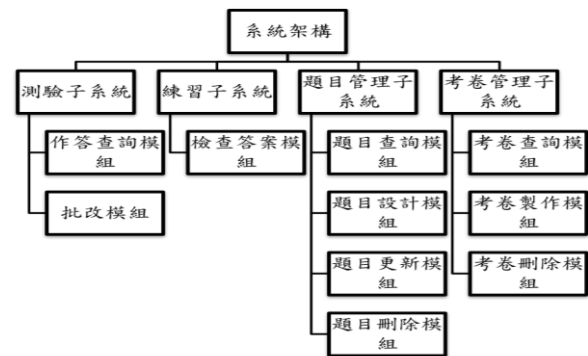


圖 9. 系統架構圖

##### (4) 考卷管理子系統

- a. 考卷查詢模組：查詢已建立的考卷。
- b. 考卷製作模組：挑選所需要的題目製成考卷。
- c. 考卷刪除模組：刪除考卷。

##### 2. 題目畫面設計

題目畫面共分為四個區域，其畫面如圖 10，分別為下列 Mission、Instruction、Toolbox 與 Workspace，以下將個別說明。

- (1) Mission：挑選運算思維題目後，將題目置於此區。
- (2) Instruction：放置提示並規定學生使用特定的方塊以便之後檢查答案流程能夠統一進行，並且在下方加入圖片輔助說明。
- (3) Toolbox：方塊依照類別區分在各個標籤內，工具箱內容可以依照題目需求進行客製化的調整。
- (4) Workspace：使用者作答區，並會依照題目需求來放置預設方塊。





圖 10. 題目畫面設計

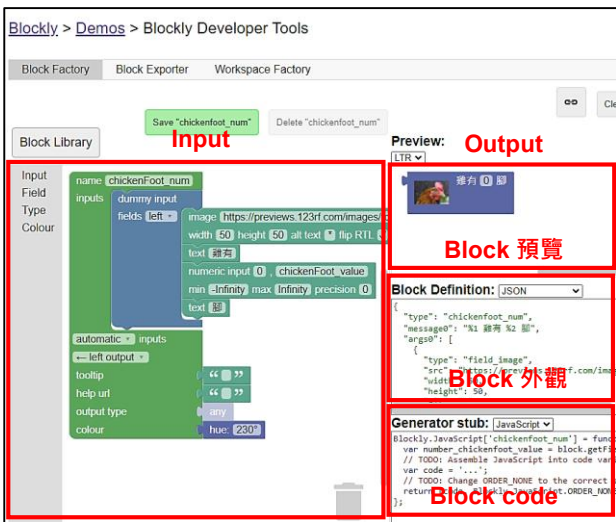


圖 11. 客製化方塊工具畫面

### 3. 設計客製化方塊

Blockly Developer Tools 是 Blockly 所提供的線上客製化方塊工具，畫面上共分成 Input 與 Output 兩區，其畫面如圖 11。其中 Input 區可以設定 Block 的外觀、屬性與類...等等，而 Output 區則分成三區，分別是 Block 預覽圖、Block 外觀與 Block Code，以下將個別說明。

- (1) Block 預覽圖：隨著使用者所設定的內容而跟著改變 Block 的預覽外觀。
- (2) Block 外觀：產生定義 Block 外觀的程式碼，分別可以輸出成 Json 與 JavaScript 兩種格式。
- (3) Block Code：產生預設的程式碼，使用者根據程式碼去設計出 Block 的流程與動作，並且可以選擇輸出五種



圖 12. Toolbox 設計畫面

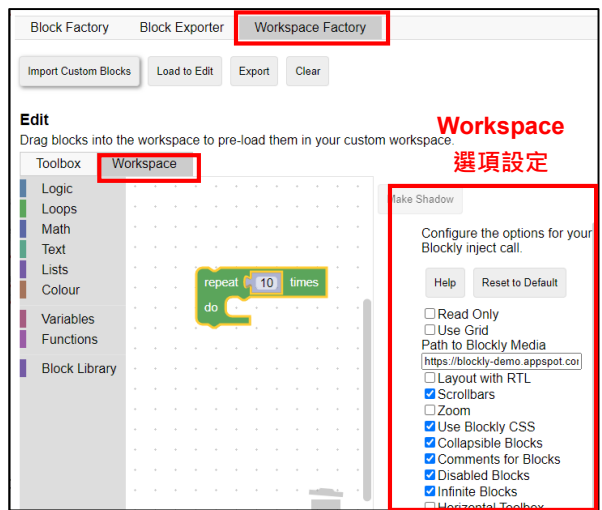


圖 13. Workspace 設計畫面

語言的程式，分別是 JavaScript、Python、PHP、Lua 與 Dart。

### 4. 設計工具箱與作答區

Workspace Factory 是 Blockly 所提供線上設計 Toolbox 與 Workspace 的工具，如圖 12，共分為 Edit 與 Preview 兩區，透過 Edit 區可以選擇所要設計的內容，分別為 Toolbox 與 Workspace，Preview 區則可以瀏覽設計的內容。

使用者在设计 Toolbox 的時候，可以从標籤內挑選 Block 加入至 Toolbox 當中，並且可以匯入客製化方塊 (Import Custom Blocks)，中間的 Category 可以增減 Toolbox 的類別。

使用者在设计 Workspace 的時候 (如圖 13)，可以从標籤挑選 Block 加入至 Workspace，並從右側視窗設定有關 Workspace 的選項。

### 5. 儲存答案流程

學生完成測驗後，系統會將學生所作答紀錄 (如圖 14)



轉成 JavaScript 語法並且儲存在字串當中（如圖 15），再將字串儲存到資料庫中作為後續檢查答案資料來源之使用。除了儲存成 JavaScript 之外，也會將答案轉成 xml 格式（如圖 16），再轉成字串儲進資料庫，以供往後查詢學生作答紀錄之用途。

## 6. 檢查答案流程

檢查答案流程如圖 17，系統會先讀取資料庫當中以

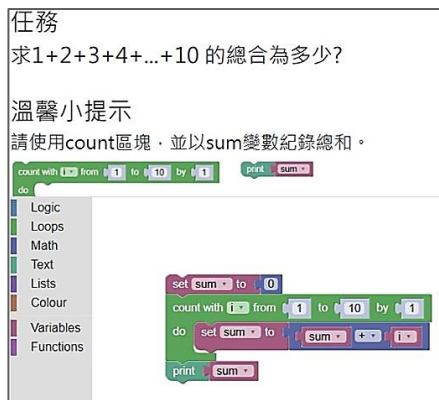


圖 14. 學生作答紀錄

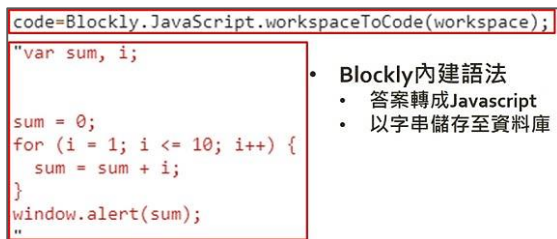


圖 15. 將學生作答紀錄轉成 JavaScript 並以字串儲存



圖 16. 將學生作答轉成 xml 格式

JavaScript 格式儲存的字串，並透過 JavaScript 的 eval() 語法執行字串，判定題目所規定的變數的值是否正確，若其值不正確則直接輸出成 0 分，正確則會替換另一組初始值並執行，然後再重複上述動作一次，總共替換兩次初始值，並判斷三次值是否正確。

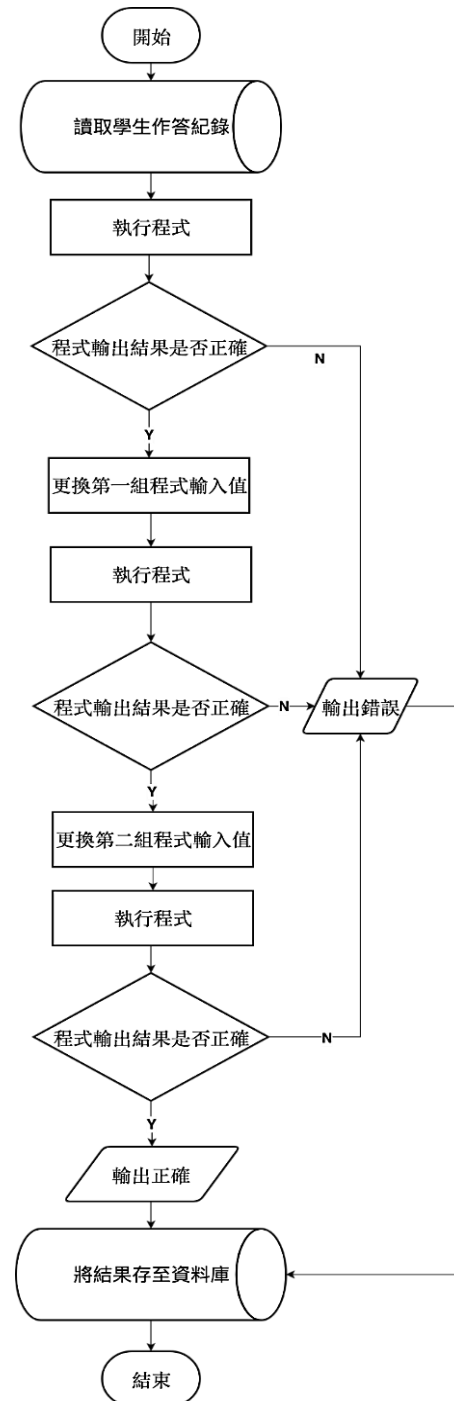


圖 17. 檢查答案流程圖



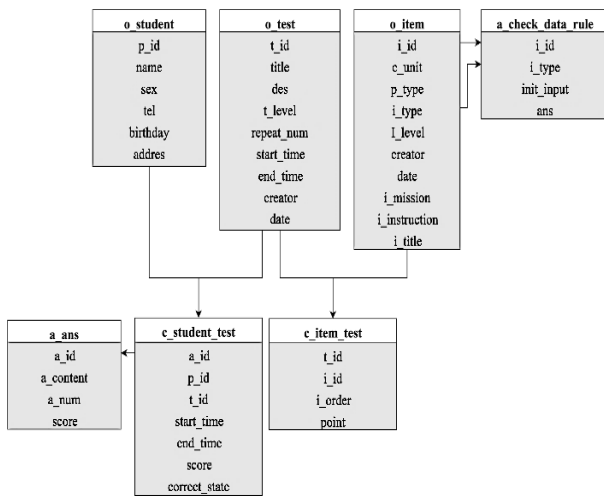


圖 18. 本系統使用之 Schema

## 7. 資料庫建置

本論文資料庫是以物件導向觀念進行設計，並將資料庫分成兩部份，分別是物件 (Object) 與關聯 (Connection) 如圖 18，物件與物件之間以關聯做為連接，讓資料庫更加地彈性化。

(1) 物件 (Object) 類型的資料表主要分成以下三個：

- **Item**：此資料表主要紀錄有關題目的基本資料，包括運算思維觀念、程式類別、題目難度、題目名稱、提示...等等。
- **Test**：此資料表主要紀錄有關考卷的基本資料，包括考卷名稱、考卷難度、可重複次數、開始時間、結束時間...等等。
- **Student**：此資料表主要紀錄有關學生的基本資料，包括生日、電話、email、住址...等等。

(2) 關聯 (Connection) 類型的資料表主要分成以下兩個：

- **item\_test**：此資料表主要紀錄 item 與 test 之間的關係，例如：題目 item 在考卷 (test) 裡面的順序與配分，將會紀錄在本資料表。
- **Student\_test**：此資料表主要紀錄 student 與 test 之間的關係，例如：學生 student 進行該份考卷 test 的起訖時間，能將每一位學生的起訖時間都設定程不同的時段，將會紀錄在本資料表。

## 四、實作與結果

本節將介紹系統的軟、硬體環境，以及系統所使用的資料表結構及欄位用途，再將各個功能與設計詳細說明。

表 2. 硬體設備規格

CPU 類型	Intel (R) Core (tm) i7-4770 CPU @3.40GHz
主機板名稱	Acer Veriton M4630G
系統記憶體	8192 MB
BIOS 版本	Acer P11-B2L

表 3. 軟體設備規格

作業系統	Microsoft Windows 10 business edition
Web server	Apache 2.4.35
程式語言	PHP 7.2.10
資料庫	MySQL 5.7.23
資料庫管理系統	phpMyAdmin 4.8.3

### (一) 系統建置

本論文所使用硬體規格如表 2，而軟體工具如表 3。

### (二) 測驗系統之實作與結果

本小節將會介紹四個子系統之實作結果，分別為題目管理子系統、考卷管理子系統、線上練習子系統與線上測驗子系統。

#### 1. 題目管理子系統：

分為題目製作、題目查詢、題目新增與題目刪除模組。

- (1) 題目製作：題目製作總共分為兩大部分，分別為題目資訊與檔案上傳，題目資訊包含了題目名稱、運算思維觀念、程式類型、題目等級 (Level)、任務 (Mission)、提示 (Instruction) 與答案 (Answer)，檔案上傳包含了任務圖片 (MissionImg)、提示圖片 (InstructionImg)、工具箱 (Toolbox) 與作答區 (Workspace)，如圖 19 所示。

圖 19. 題目製作畫面





(2) 題目查詢、更新與刪除：題目查詢條件共有四項（如圖 20），分別為題目名稱、運算思維觀念、程式類別與題目等級（Level），設定完條件並點選查詢，查詢結果將會於下方以表格形式來顯示（如圖 21），其左下方會有更新與刪除的功能，勾選單筆資料並點選更新按鈕後，會跳轉到編輯的頁面並且會在頁面上顯示該筆資料內容（如圖 22），使用者更改完資料後，即可點選左下方按鈕來決定是否要更新。

圖 20. 題目查詢畫面

checkbox	item_id	i_title	core_unit	programming_type	i_level	creator
<input type="checkbox"/>	1	ball_bounce	Dec,PR	arit,if-else	8	test
<input type="checkbox"/>	2	AtoB	Dec,PR	arit,if-else	5	test
<input type="checkbox"/>	3	1to100	Dec,PR	arit,if-else	5	test
<input type="checkbox"/>	4	1to99	Dec,PR	arit,if-else	5	test
<input type="checkbox"/>	5	sum100	Dec,PR	arit,if-else	5	test
<input type="checkbox"/>	6	2to200	Dec,PR	arit,if-else	5	test
<input type="checkbox"/>	7	teeeeeest	Dec	arit	2	test

Update Delete

圖 21. 題目查詢的結果顯示畫面

圖 22. 編輯題目畫面

2. 考卷管理子系統：

分為考卷製作、考卷查詢與考卷刪除。

(1) 考卷製作：考卷製作模組利用 Blockly 來做為輔助，工具箱內分為兩項標籤，分別是 Test 與 Item，Test 標籤（圖 23）內存放考卷的相關資訊，考卷相關資訊的 Block 設計成全部都是綠色，代表這是與其有相關的 Block，而 Item 標籤內（圖 24）存放所有的題目，每一個題目為一個 Block，全部都設計成藍紫色，代表這是與其相關的 Block，右邊的 Block（圖 24 中紅圈處）為該題的配分。透過挑選所需要的題目 Block 並連結考卷資訊來製作考卷（圖 25）。



圖 23. 考卷製作模組的 Test 標籤內的考卷資訊



圖 24. 考卷製作模組的 Item 標籤內題目



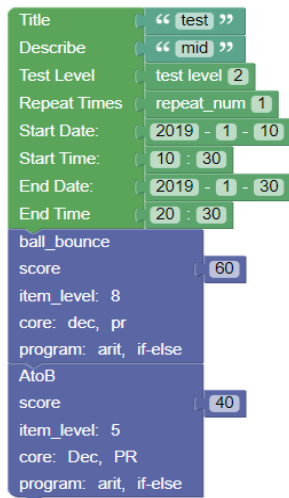


圖 25. 透過挑選題目 Block 並連結考卷資訊來製作考卷

(2) 考卷查詢與刪除：考卷查詢條件共分為三大部分（圖 26），分別為考卷名稱、等級（Level）與起訖時間，設定完條件並點選查詢，如果時間條件沒有設定，在點選查詢後會跳出警訊，而查詢結果將會於下方以表格形式來顯示（如圖 27），第一個表格顯示考卷相關資訊，第二個表格則是顯示每個考卷內的題目名稱、排序與配分，其左下方會有刪除的功能。

3. 線上練習子系統：

線上練習子系統與運算思維課程教材配合，提供學生線上練習使用，共有四題（圖 28），分別為自由練習區、1 到 100 練習題、1 到 99 練習題、球與自由落體練習題。

進到練習題後（圖 29），學生能夠利用工具箱（Toolbox）與作答區（Workspace）設計程式，完成程式後，能夠直接按 Check 鈕檢查答案，透過瀏覽器直接顯示答案正確與否。若學生利用小聰明的方式來設計程式，例如：直接將  $1+2+\dots+100$  的結果 5050 設定到答案變數裡（如圖 30），則會出現如圖 31 的警告訊息。

checkbox	test_id	test_title	des	t_level	repeat_num	start_time	end_time	creator
<input type="checkbox"/>	1	test1	test1	5	1	2018-10-01 10:00:00	2018-12-01 20:00:00	test1
<input type="checkbox"/>	3	test888	test888	1	0	2018-12-01 10:00:00	2018-12-20 15:00:00	test1

test_id	test_title	item_title	i_order	point
1	test1	ball_bounce	1	25
1	test1	AtoB	2	25
1	test1	1to100	3	25
1	test1	teeeeeest	4	25
3	test888	1to100	2	50
3	test888	ball_bounce	1	50

圖 27. 考卷查詢後，搜尋結果以表格顯示在下方

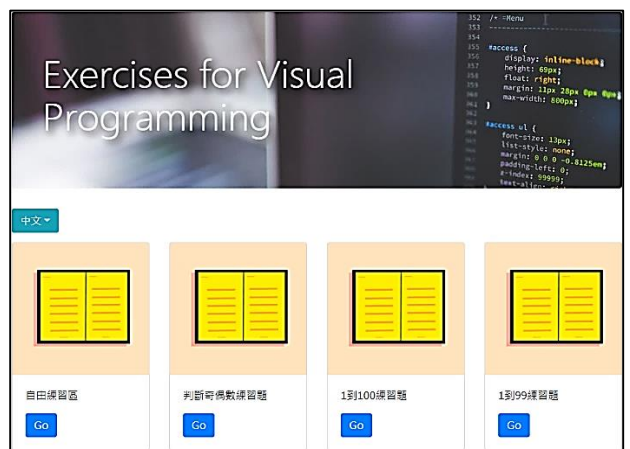


圖 28. 線上練習畫面

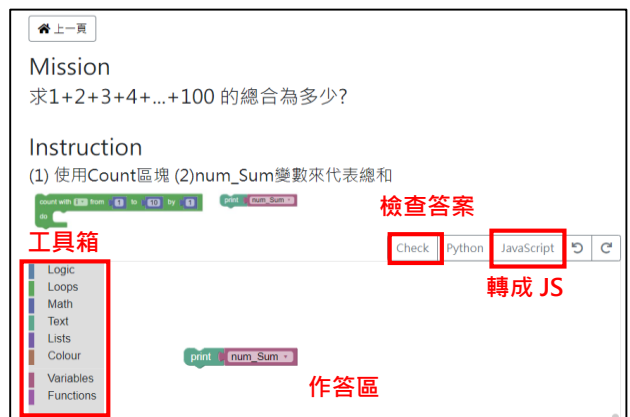


圖 29. 練習題畫面

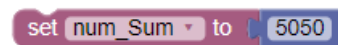


圖 30. 直接設定答案

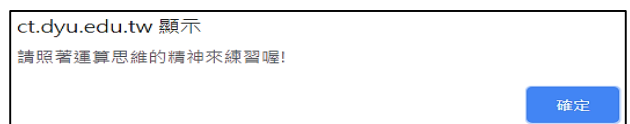


圖 31. 警告訊息

### 考卷查詢

考卷名稱:

Level:

開始時間:

結束時間:

Item info will be listed here...

圖 26. 考卷查詢畫面，搜尋條件共分為三大部分



系統除了檢查答案的功能之外，還有能將學生所設計的 Block (如圖 32) 轉成 JavaScript 的功能 (如圖 33)，利用上述的功能來幫助學生實際演練運算思維。

#### 4. 線上測驗子系統：

線上測驗、批改、作答結果查詢模組與測驗結果。

- (1) 線上測驗：這次的測驗是正式上線，是作為大葉大學的運算思維通識課程的期末測驗 (圖 34)，測驗日期為 2019/1/7 至 2019/1/16，測驗內容分兩部分，分別為學科與術科，學科內容是以運算思維大核心觀念為主，題行為選擇或是非題總共占 80% 分數，術科則是視覺化程式設計為主，總共 20% 分數，術科部分總共兩題，第一題配分為 20 分，而第二題為加分題，答對者加 10 分。進入測驗之前，必須先登入校務系統中 (圖 35)，登入成

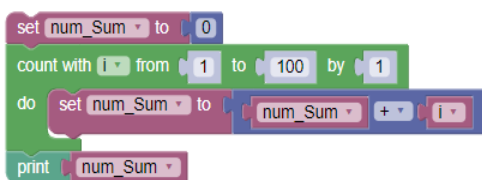


圖 32. 使用者設計的 Block

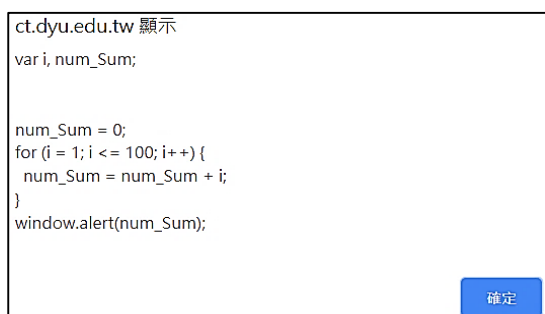


圖 33. 系統轉成 JavaScript 並顯示



圖 34. 大葉大學運算思維通識課程期末測驗資訊

功後會看到任務開始的畫面 (圖 36)，點選後將會分別進入第一題與第二題 (圖 37、38)，而這次的測驗中兩題的題目都需要運用到運算思維的其中三個觀念，分別為 Pattern Recognition、Abstraction 與 Algorithm Design，設計出符合題目要求的程式。

- (2) 批改模組：批改模組執行在客戶端 (圖 39)，考慮到大量數據的運算可能會導致瀏覽器崩潰，因此將兩題的批改分開進行，執行完的結果，第一題共有 1530 筆且花了 24 秒完成批改，第二題共有 1409 筆則花了 18 秒完成批改，這次的批改相當地順利沒遇上瀏覽器崩潰的問題，從批改時間來看也相當快。



圖 35. 測驗登入畫面



圖 36. 任務開始畫面



圖 37. 測驗第一題畫面





圖 38. 測驗第二題畫面



圖 39. 批改考卷畫面

- (3) 作答結果查詢：作答結果條件共分為兩項（圖 40），分別為學號與題號，設定完條件後並點選查詢後，會將當時的任務內容、提示、工具箱（Toolbox）與作答結果顯示下方（圖 41 與圖 42），除了顯示當時作答結果內容之外，還會還原當時 Block 所在的位置。
- (4) 測驗結果：這次測驗結果，應參加人數為 1800 人，然而實際上有 1530 人參加這次考試，然而在答題的部分，第一題配分為 20 分，共有 1530 筆，答對率有 55.8%，而第二題為加分題，答對者加總十分，共有 1409 筆，答對率有 7.026%。  
從這次測驗當中抽查部分學生的答案，學生經系統自動判斷後視為正確的，但是所設計出的程式卻是與出題者所希望的方式不同（如圖 43），經過這次的經驗，在題目與自動化判斷的方面需要設計得更加完善，才能讓自動化判斷更加的精準。
- (5) 成績匯出與匯入：本系統與大葉的校務系統有結合，因此學生能以學校資訊系統的帳密登入考試，同時各班級的學生名單也可以匯入此系統。當學生考完試後，本系統可以將學生成績匯出 Excel 檔，再由授課教師將 Excel 檔上傳到學校的成績系統，成績系統裡即可看到此次的期末考試成績（如圖 44），無需教師手動一個個輸入分數。



圖 40. 學生答案查詢畫面



圖 41. 學生第一題作答結果查詢



圖 42. 學生第二題作答結果查詢



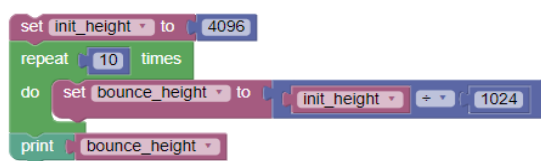


圖 43. 少數例子判斷錯誤的程式

序	學號	姓名	期末考	作業	課堂討論	課程參與...	上學次...	總分	備註
1	B072	1	何 蓉	65.00					
2	B072	4	張 添	20.00					
3	B072	5	詹 誠	56.00					
4	B072	6	蔡 鈞	60.00					
5	B072	0	劉 漢	92.00					
6	B075	3	林 華	86.00					
7	B078	4	蔡 鴻	69.00					
8	B078	4	呂 宜	83.00					
9	F070	4	陳 瑋	47.00					
10	F076	1	孟 傑	47.00					
11	F076	2	歐 琦	49.00					
12	F076	3	張 馨	98.00					
13	F076	4	廖 毅	50.00					
14	F076	6	徐 煥	67.00					

圖 44. 學生期末成績匯入結果

## 五、結論與未來展望

本論文實作出一套以 Blockly 為基礎之視覺化程式設計測驗系統，提供運算思維課程學習與評量之用，除了讓教師與學生在課堂上做練習，教師也可以自行命題，設定答案所需的程式積木並配分，由系統自動評分。由於系統與學校的校務系統整合，學生能使用學校的帳號登入測驗，班級的評量結果也能匯出 Excel 檔供教師再匯入學校的成績系統中，在修課人數眾多的情形下，對授課教師來說確實能提供相當大的便利。

本系統由於是以檢測變數值的方式進行評量，目前僅適合使用在計算類型的題目，較缺乏趣味性；未來希望能再添加如 Scratch 的角色及舞台，提供更有吸引力的練習題目，也讓授課教師的命題能更多元化，讓運算思維課程的教學與評量內容更豐富。

## 誌謝

感謝匿名審稿委員給了極多建設性建議，使得本論文的品質顯著提升。

## 參考文獻

1. 盧一屏 (民 99)，以 VTK 為基礎之視覺化程式設計環境在科學視覺上之應用，大葉大學資訊工程所碩士論文。
2. 賴和隆 (民 105)，應用運算思維於高中資訊教學設計之分享，教育脈動，6，143 - 155。
3. Code.org (2013) Learn today, build a brighter tomorrow, Retrieved July 13, 2019, from <https://code.org>.
4. Google (2006) Blockly games, Retrieved May 3, 2020, from <https://blockly-games.appspot.com>.
5. Google (2012) Blockly | Google Developers, Retrieved May 2, 2018, from <https://developers.google.com/blockly>.
6. Johnson, D. E. (2016) ITCH: Individual Testing of Computer Homework for Scratch Assignments. In: *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 223–227, Memphis, Tennessee.
7. Maloney, J., M. Resnick, N. Rusk, B. Silverman and E. Eastmond (2010) The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*, 10(4), Article 16.
8. MIT App Inventor (2010), Explore MIT App Inventor, Retrieved November 10, 2019, from <http://appinventor.mit.edu/>.
9. MIT Media Lab (2006) Scratch - Imagine, Program, Share, Retrieved March 20, 2019, from <https://scratch.mit.edu>.
10. Mönig, J. and B. Harvey (2011) Snap! Build Your Own Blocks, Retrieved August 5, 2019, from <https://snap.berkeley.edu/>.
11. Pasternak, E., R. Fenichel and A. N. Marshall (2017) Tips for creating a block language with blockly. *2017 IEEE Blocks and Beyond Workshop (B&B)*, 21-24, Raleigh, NC.
12. Revell, M. (2019) What Is Visual Programming?, Retrieved December 15, 2019, from <https://www.outsystems.com/blog/what-is-visual-programming.html>.
13. Stahlbauer, A., M. Kreis and G. Fraser (2019), Testing scratch programs automatically. In: *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 165-175.
14. Tsukamoto, H., Y. Takemura, Y. Oomori, I. Ikeda, H. Nagumo, A. Monden and K. Matsumoto (2016) Textual vs. visual programming languages in programming education for primary schoolchildren. *2016 IEEE Frontiers in Education Conference (FIE)*, 1-7, Erie, PA.
15. von Wangenheim, C. G., M. F. Demetrio, R. Pelle, N. d. C.



- Alves, H. Barbosa and L.F. Azevedo (2018) CodeMaster – automatic assessment and grading of app inventor and Snap! programs. *Informatics in Education*, 17(1), 117–150.
16. von Wangenheim, C. G., J. C. R. Hauck, I. D. S. Solecki, J. V. A. Porto, K. A. Justen, M. F. Demetrio, N. D. C. Alves and R. Pelle (2017) CodeMaster, Retrieved May 25, 2019, from <http://apps.computacaonaescola.ufsc.br:8080/index.jsp>.
17. Wikipedia (2020) Visual programming language, last edited on November 17, 2020, Retrieved November 18, 2020, from [https://en.wikipedia.org/wiki/Visual\\_programming\\_language](https://en.wikipedia.org/wiki/Visual_programming_language).
18. Wing, J. M. (2006) Computational thinking. *Communications of the ACM*, 49 (3), 33-35.
19. Wing, J. M. (2014) Computational Thinking Benefits Society. 40th Anniversary Blog of Social Issues in Computing, Toronto, Retrieved January 5, 2019, from <http://socialissues.cs.toronto.edu/index.html%3Fp=279.html>.

收件：108.12.03 修正：109.04.23 接受：109.11.23

