**作業系統**

**Thread專題報告**

**組長(唯一組員)：林政儒**

**學號：101502039**

目錄

1. **基本功能介紹**
   1. 構想起源
   2. 細節增添
   3. 完整功能
2. **環境使用安裝**
   1. Java standard edition development kit 7
   2. Eclipse
3. **程式碼介紹**
   1. GUI
   2. Test Interface
   3. Timer
4. **遭遇困難與解決**
   1. 如何搜尋檔案
   2. 如何偵測完成測驗
   3. 擴充單字庫
   4. 無法複製jar檔內的資料庫出來
5. **心得感想**
6. **專案開發歷史紀錄**

**基本功能介紹**

**構想起源：**

說到Thread，第一個想法就是計時器。因為無論做什麼樣的工作，要計時的話必定得將計時視為另一個獨立的程序，並且同時與主程序一起執行。否則便無法達到計時準確度就會不佳。

那麼計時器應該用在哪？我是按照自己的需求去想我該打什麼樣的程式。我目前在準備托福和GRE的英文檢定，而這兩項檢定需要訓練英打速度。如果我能為自己設計程式幫助自己訓練英文打字能力，一定很棒！

**細節增添：**

一開使輸入使用者名稱。若有該使用者則讀取他的資料，若沒有則新建一個使用者資料庫。接著來到選擇關卡畫面。

選擇特定的關卡後，程式會從字彙庫中挑選該類的10個單字，限定1分鐘內要使用者輸入完。

無論完成或失敗，最後程式將新的結果存入使用者的資料庫內。

選擇關卡的畫面每10秒會更新，將最新的10筆成績顯示出來。

**完整功能(與基本使用)：**

[開始介面輸入身分]

* + 1. 程式會讀取哪位使用者要使用本程式，並載入該使用者的資料，若無該使用者，則新增一個使用者資料庫。

[關卡選擇及使用者資料畫面]

* + 1. 使用選單(J Combo Box)逐步選擇要挑戰的關卡。例如，我要選verbal advantage中的level6。第一個選單選verbal advantage，接著第二個選單才有level6顯示出來可以選。選定後按「確認挑戰」就會啟動關卡畫面。
    2. 右邊的text area是顯示該使用者過往的紀錄。最新的紀錄會顯示在最上方，每10更新一次。最頂端的數字代表上一次完成測驗花了多少秒，第二列的數字為上上次…，以此類推。

[關卡畫面]

* + 1. 最左邊會提示要輸入的單字，有時候甚至是一個詞或片語。
    2. 中間的text field是輸入框
    3. 「00:00」是計時器。
    4. 最右邊是開始按鈕，按下去後計時器便會開始啟動。
    5. 開始計時後，我們要依照提示輸入「一模一樣」的字詞。每當我們成功輸入一個一詞，便會換下一個提示。在60秒內成功輸入10個字詞就算完成。若不幸60秒內無法完成，那麼該次挑戰成績就是60，不會往上遞增。

**環境使用安裝(Java&Eclipse)**

先說明一下我們這個程式所使用的環境，然後簡單的介紹安裝與使用方法。

作業系統：Window7

選用語言：Java

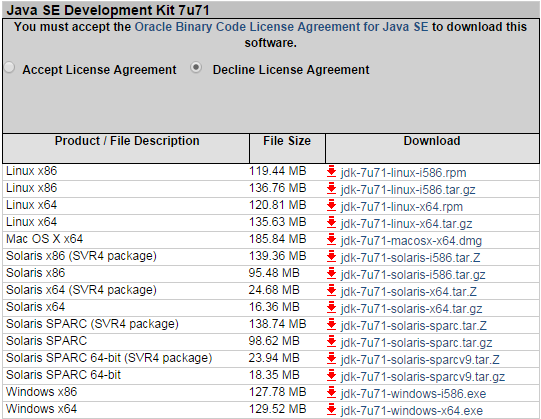
Java開發工具：JDK7(Java standard edition development kit 7)

編寫環境：Eclipse(eclipse-java-luna-SR1-win32-x86\_64.zip)

安裝介紹：

**JDK7**

* + 1. <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>



* + 1. 由控制台->系統及安全性->系統->進階系統設定->環境變數->選擇Path->編輯為C:\Program Files\Java\jdk1.7.0\_06\bin->按下確定即可

PS: Java的版本資訊在控制台->程式及->Java可以找到

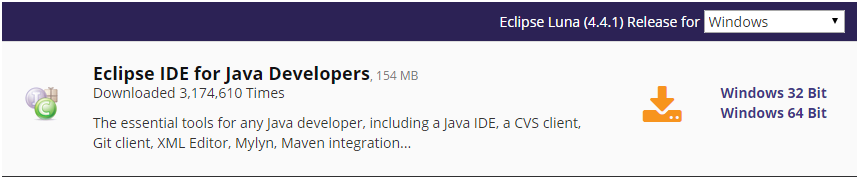
PS: 如果未安裝**JRE(Java Runtime Environment)**，則也需要安裝，這個是用來讓系統執行java，一般電腦原本就已經安裝好了(因為很多軟體都是由java執行的)。

<http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>

若.jar檔仍未能開啟，右鍵點選檔案->內容->變更開啟程式->選取最新版本JRE

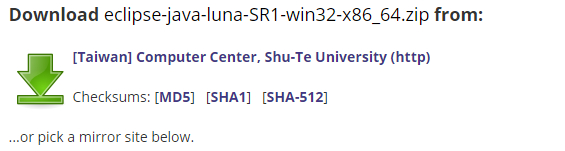
**Eclipse**

* + - * 1. <http://www.eclipse.org/downloads/>

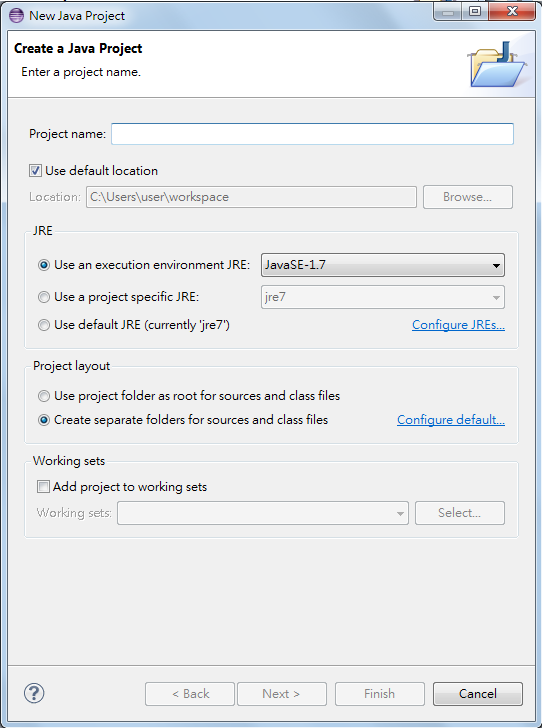


* + - * 1. 點選後進入

<http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/luna/SR1/eclipse-java-luna-SR1-win32-x86_64.zip>



使用方法介紹：

1. 點選eclipse捷徑
2. File->New->Java Project
3. 輸入專案名稱按下確定
4. 點選專案->File->New->Class即可開始編寫java程式

**程式碼介紹**

首先說明程式碼大致架構共具有GUI、Test interface、Timer三個class，以及安裝資料庫的部分程式瑪。

**先由GUI開始實作**

Class GUI的架構下內含2個panel和一個thread。

分別是login panel、Start panel，與每10秒儲存資料與更新資料的thread。

Login panel：用輸入的user name去搜尋使用者的資料，將資料存在一個陣列user data的陣列裡。

如果沒有該使用者，就新生成一個使用者資料，也同樣放進user data內。接著畫面進入到start panel。

Start panel：從login panel拿到使用者資料(包含字串data path, 陣列user data)顯示在畫面上。

同時，啟動自身的thread(GUI本身是runnable物件)，用來持續更新使用者的成績紀錄。

最後，我設計用選單(J combo box)逐步選擇關卡的單字庫。先選擇要哪類的單字，托福、gre、verbal advantage？選擇其中一個後，第二個combo box才會顯示該類的單字庫做進一步的選擇。

選擇好關卡後，按下確認便會啟動另一個視窗，也就是test interface。

**再進行Test interface物件的實作**

Test interface分成三部分，測驗方式、計時器、結束。

測驗方式：開啟test interface這個介面時，程式會從選擇的單字庫中挑選10個不重複的單字儲存在questionWords的陣列裡，並且用JLabel顯示一個。我實作一個setQuestionVisible，這個fuction決定現在該顯示哪一個單字。每當我們成功輸入一個單字時，我們將輸入完的單字消掉，接著顯示下一個單字。過程中我們用變數cur\_question來紀錄現在挑戰到第幾個單字。

計時器：最重要的計時器會在我們按下開始鍵後，計時一分鐘。如果一分鐘到了，計時器會顯示「End」。

結束：我定義測驗結束的時機為timer顯示「End」就代表程式結束了。Test Interface本身也是runnable物件。當Test interface畫面一出現，就已經開始在偵測timer是否顯示為「End」。當成是結束，我們會讀取計時器上的秒數，並且更新user data。

**實作Timer物件**

簡單的計時器，同時也是一個runnable物件。有兩個變數second和mini second來紀錄時間。Start function用來啟動計時功能，stop fuction用來停止計時。

class內部的run function內含一個while判斷一個布林變數isRunning來決定是否執行迴圈。如果我們要停止計時，啟動stop function使 isRunning = false。這時後timer會跳出while迴圈，計時也就停止了。

也就是說，這是一個一次性的計時器，不能暫停再恢復計時。

當計時到一分鐘時，timer顯示為「End」。

**安裝資料庫**

程式一開始執行時，會先判斷C:\English typing training這個目錄下是否有user\_info與word\_bank這兩個資料夾。如果有則正常執行程式，正常讀檔。如果沒有，則將jar檔內包好的資料庫複製進去，作為初始資料庫。

首先利用url來存入包在jar檔內的單字表(word bank)與使用者資料(user info)。將兩者的的資料用java.io.File讀取後複製到C槽下。程式碼在GUI.class的底端，有分複製檔案與複製整個資料夾。複製檔案的部分利用ioStream與byte陣列去實作，而複製整個資料夾的部分則是用deep-first-search的方式實作。例如： 我們要複製資料夾A，首先我們去看資料夾A下的所有檔案和子資料夾，遇到檔案就複製，遇上資料夾則遞回傳入這個子資料夾複製該資料夾下的所有內容。

**遭遇困難與解決**

**如何搜尋檔案**

一開始思考是否要另外寫一個class來做deep-first-search或是wide-first-search先把全部資料庫掃瞄一次，後來發現這樣做太麻煩。因為我們可以一步一步選擇我們要資料夾，最後選擇我們要的資料檔案。不必一開始就將目錄樹的全部紀錄下來。

**如何偵測完成測驗**

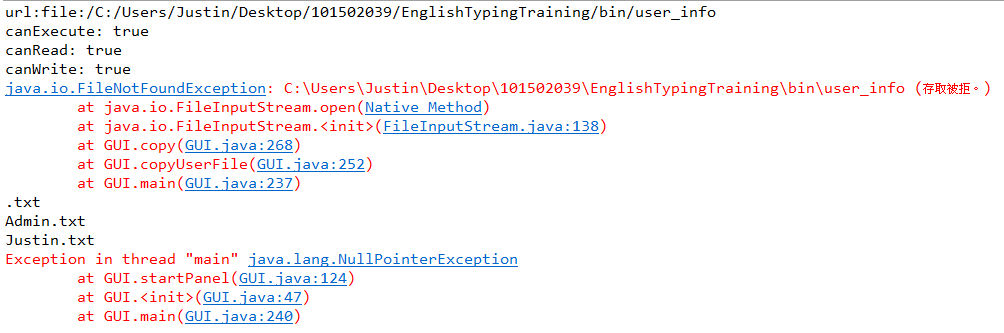
因為java沒有一種可以偵測JLabel變化的listener，所以只好回歸原始另外開啟一個thread來偵測。

**擴充單字庫**

這是這個程式最耗精力的地方。大部分的的單字表都是英文+詞性+解釋，而我的程式只需要英文的部分。一開始，我用人工去縮排和刪去不要的部分，可是這樣做半小時弄不完一個關卡。於是我另外又寫了一個簡單的程式幫我將英文部分擷取出來，大大加快建立資料庫的時間。

另外因為托福的單字量實在太多，而手中又沒有現有的單字表檔案，只能以人工的方式製作三分之一的單字庫代表，慎感抱歉。

**無法複製Jar檔內的資料庫出來**



如上圖所示，Console中顯示jar內的檔案可以被執行、被讀取、被複寫，卻意外跳出「存取被拒」的錯誤提示。後來發現是因為我寫的複製function是用來複製檔案而不是資料夾，重新再寫複製整個資料夾的function即可。

**總工時**

程式碼編撰：20小時

報告整理：6小時

總時數：26小時

**心得感想**

老實說這次的作業並沒有太多過程中的心得或感想。因為目標太明確，我就是想打一個能幫助我熟悉英文單字，並加快我英打速度的程式。過去的大小專題也早就讓我對thread有基礎認識，其中也包括如何建立實作runnable的類別。

當我的目標明確，我就盡最大的努力以最短的時間內完成。這次的突破應該就是開發時間吧！遠比過去開發相同程度的專案快得多。上學期開發socket多人聊天室光程式部分用去了將近40小時在上面，這40小時還不包括與同學之間的技術討論，以及摸索socket。而這次僅憑一己之力以盡可能短的時間完成專案，對自己開發專案的進步速度感到很滿意。

另外這也是一個可以擴充單字庫的專案，只要將單字表照著我們統一的格式作成txt檔，丟進C:\English typing training內的word bank資料夾內，接著開啟jar檔執行這個專案就可以選取到這個檔案，並且挑戰它了。

Ps.單字庫檔的格式為一個單字一行，一份檔案至少要10個單字。

Pps.只可以在word bank下新增資夾。若在其它地方新增資料夾，程式不會讀到，甚至可能會造成程式錯誤。

**專案開發歷史紀錄**

專案目標：使用thread完成一個程式

2015/5/10：直覺thread可以當作計時器，用來算時間、做事的速度。

2015/5/13： 有了想法，幫助托福、GRE考生計時練習英打，並蒐集一小部分的托福單字做成txt檔當成測試資料。

2015/5/14：製作GUI，用Jcombobox逐步取得關卡資料

2015/5/14：製作關卡介面

2015/5/14：加入計時器，記錄使用者資料

2015/5/15：顯示過去10筆該使用者資料

2015/5/16：擴充單字庫，但一己之力有限，只能擴充部份單字 ps.托福字彙含蓋領域過廣，難以建檔。

2015/5/22：嘗試圖破現狀建立jar檔，並且以安裝的方式在C槽內建立資料庫卻遭不明原因失敗。

2015/5/22：偵測出不明原因，已修正。程式已完成。