

## 第十一章 Java+MySQL 專題：

### 銀行儲蓄存款系統

目前資料庫取得容易，甚至 MySQL 可免費使用，因此，幾乎不會有人將資料寫入檔案(如第六章介紹)，除非序列的控制程式。目前大多將資料直接儲存於資料庫內，不僅可以存放資料還可查詢操作，非常方便。本書就以 Windows 系統下的 AppServ 套件，來連結 MySQL 資料庫，並且用一個簡單『Bank\_db』資料庫為範例，製作一套『銀行儲蓄存款系統』專題研討。

請先將此環境建立起來。本書附有：AppServ 套件、bank\_db.sql 等兩只軟體，以下分別說明之。

#### 11-1 AppServ 套件安裝

AppServ 套件是很容易取得的免費軟體，安裝後包含下列套件：

- (a) Apache 網頁伺服器(Apache Web Server)：存放使用者所開發的 HTML 文件，可供網路使用者透過瀏覽器，閱讀文件內容。
- (b) PHP 編譯器：負責編譯網頁文件的 PHP 程式片段。亦是，負責將 PHP 所編寫的程式轉換成 HTML 文件。
- (c) MySQL 資料庫伺服器(Database Server)：提供開啟與管理使用者所建立的資料庫。在一個 MySQL 上可允許建立多個資料庫。

安裝步驟如下：

##### (1) 步驟 1：

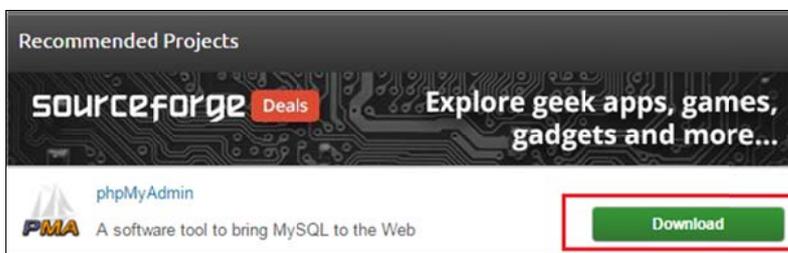
利用 Google 搜尋 AppServ 官方網站(<http://www.appservnetwork.com>)。



(2) 步驟 2：下載最新版本。(目前所看到最新版是 AppServ 2.5.10)



則出現下載頁面：



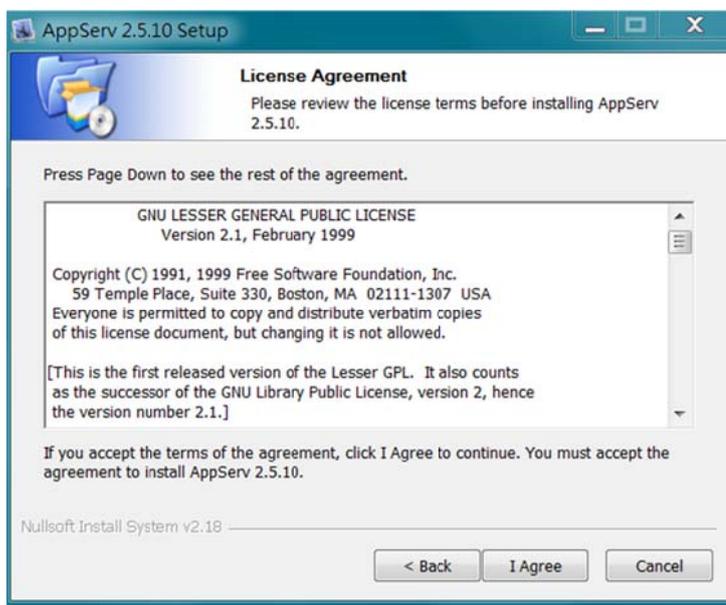
(3) 步驟 3：下載後，點選下載檔案安裝：



則出現 appServ 安裝頁面：



(4) 步驟 4：版權宣告，按『 I agree 』鍵，接下來幾個步驟，只要按『 next 』即可。



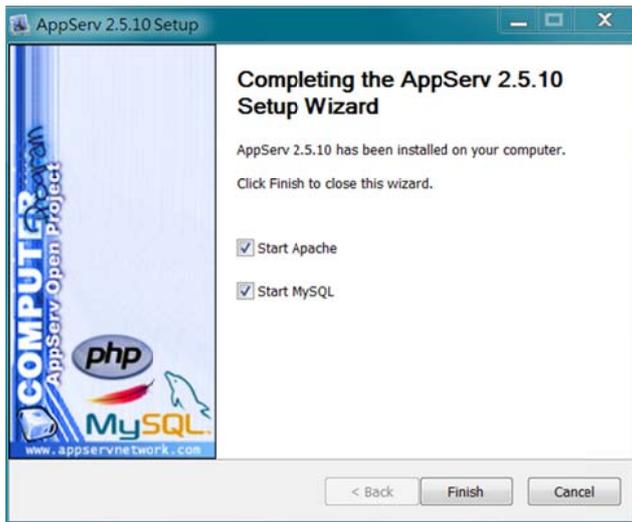
(5) 步驟 5：設定 Apache 伺服器 DNS 名稱，與管理者的 E-Mail(僅能夠單機使用 localhost)，無法外界連線，可隨便輸入。



(6) **步驟 6**：設定密碼，為了安全起見(怕忘記)，我們都設定『123456』，字型編碼要選定『UTF-8 Unicode』中文才會顯示正常。



(7) **步驟 7**：安裝完成：



### (8) 步驟 8：檢視管理工具

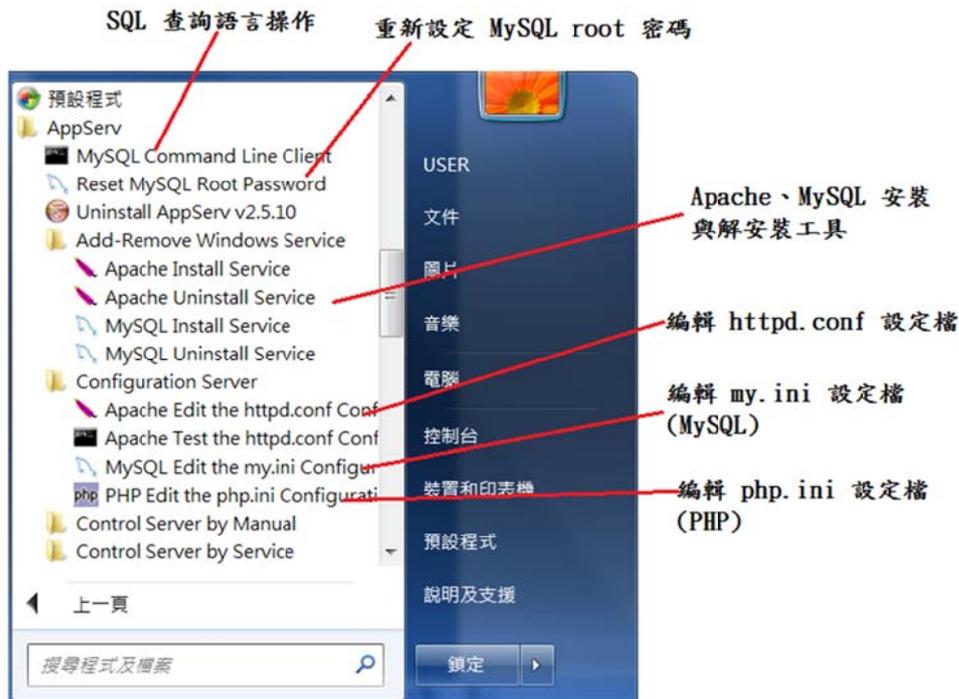
由系統開始 -> 所有程式，就可以觀察到安裝完成的 AppServ 應用程式。這裡是管理工具，並非執行程式。



## 11-2 AppServ 套件管理

### 11-2-1 Apache 與 MySQL 設定檔

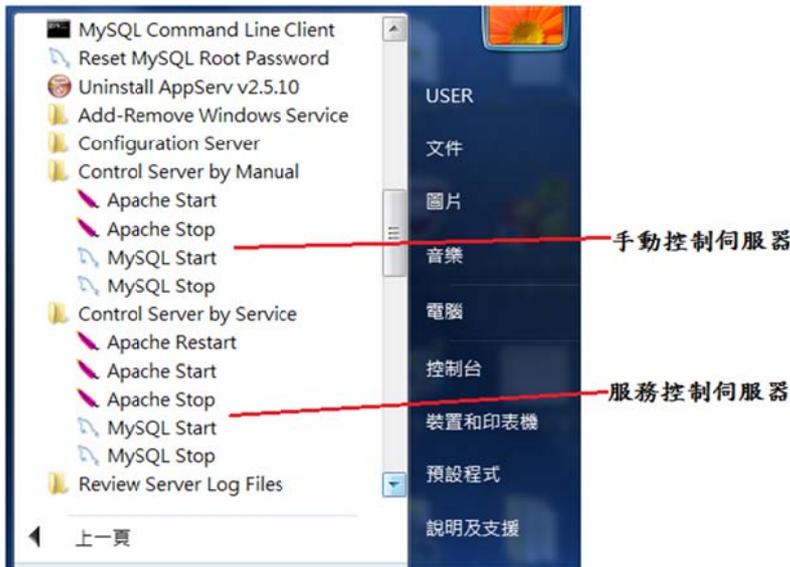
許多情況下，必須編輯 Apache、PHP 與 MySQL 設定檔，來規劃伺服器的工作環境，編輯路徑如下圖所示，說明如下：



- (a) **MySQL Command Line Client**：進入 MySQL 得客戶端，並可利用 SQL 命令操作資料庫。
- (b) **Add-Remove Windows Server**：解除或安裝(重新安裝) Apache 或 MySQL。
- (c) **Apache Edit the httpd.conf**：可編輯 httpd.conf 系統規劃檔，修改後 Apache 須重新啟動。
- (d) **MySQL Edit my.ini**：可編輯 my.ini 系統規劃檔，修改後 MySQL 須重新啟動。
- (e) **PHP Edit the php.ini**：可編輯 php.ini 規劃檔，修改後 Apache 須重新啟動。

## 11-2-2 管理伺服器運作

透過下列選單可以啟動、停止、重新啟動 Apache 或 MySQL。



## 11-3 MySQL 伺服器管理

### 11-3-1 連結 AppServ

開啟瀏覽器(IE 或 chrome)後，在 URL 位置輸入：http://localhost，則出現下圖畫面。



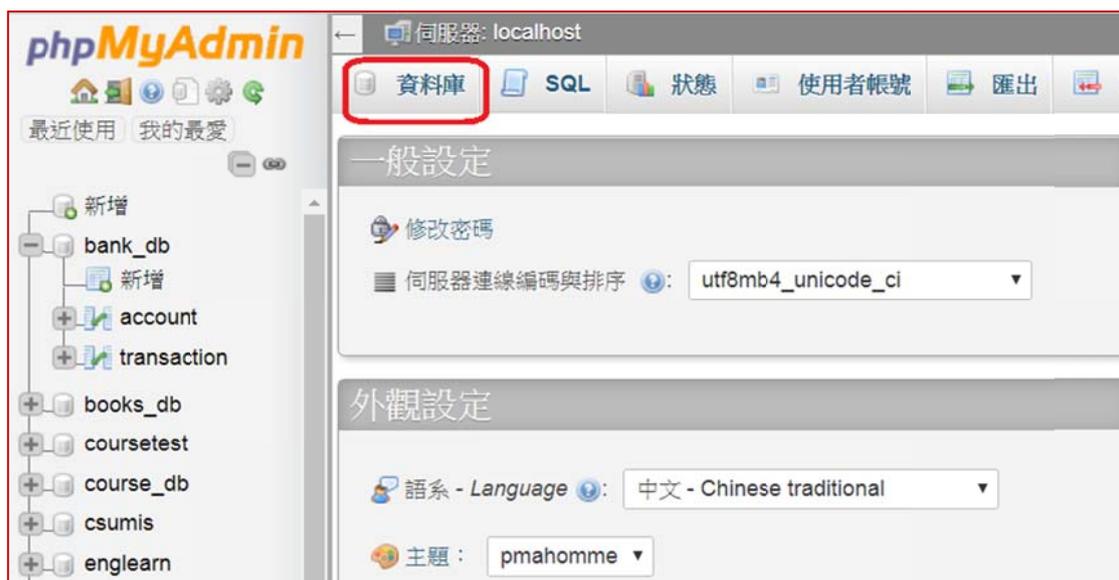
輸入使用者名稱(root) 與密碼(12345678)，再按登入即可。



## 11-3-2 建立資料庫 – Bank\_db

### (A) 進入 phpMyAdmin 首頁

登入 phpMyAdmin 套件後，出現其首頁，如下：(選擇欲操作的資料庫)



### (B) 建立資料庫 – Bank\_db



### (C) 匯入資料庫 – Bank\_db.sql

請到教學網站下載 bank\_db.sql 檔案。



#### (D) 瀏覽 Bank\_db 資料庫內容

包含 account 與 transaction 兩資料表。



## 11-4 Java + MySQL 連結

### 11-4-1 Java 連接 MySQL 設定

接著必須設定 JDBC(Java Database Connection) , Java 程式才能連結到 MySQL , 有下面幾個步驟 :

#### (A) 下載 MySQL Connector/J

到 MySQL 網頁上下載(<https://dev.mysql.com/downloads/connector/j/>) , 如下 :



Generally Available (GA) Releases | Development Releases

### Connector/J 5.1.45

Select Operating System:  
Platform Independent

Platform Independent (Architecture Independent), Compressed TAR Archive	5.1.45	3.3M	Download
(mysql-connector-java-5.1.45.tar.gz)			MD5: ab9ac454a959859a297b53bdf156f3c   Signature
Platform Independent (Architecture Independent), ZIP Archive	5.1.45	3.6M	Download
(mysql-connector-java-5.1.45.zip)			MD5: 2e3a933f8f7642345d5f34bf43374f8e   Signature

下載之後，將其解壓縮，接下來。

## (B) 複製 Connect Jar 檔案

將 mysql-connector-java-5.1.45-bin.jar 複製下來，如下：



名稱	修改日期	類型	大小
src	2017/12/6 上午 1...	檔案資料夾	
build	2017/11/22 下午 ...	XML 檔案	90 KB
CHANGES	2017/11/22 下午 ...	檔案	241 KB
COPYING	2017/11/22 下午 ...	檔案	18 KB
mysql-connector-java-5.1.45-bin.jar	2017/11/22 下午 ...	Executable Jar File	977 KB
README			60 KB
README			63 KB

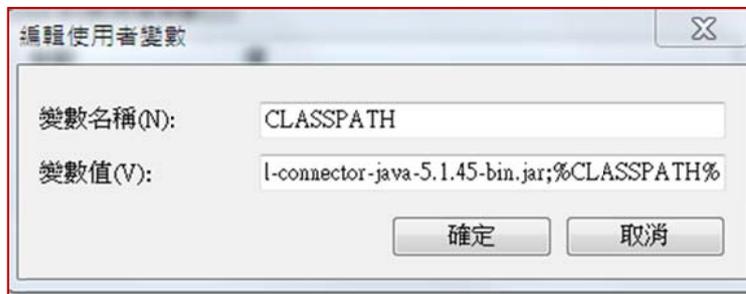
## (C) 張貼上 Connect Jar 檔案

接著，將此檔案複製到個人方法庫的目錄下，譬如，選定於 C:\JAR 目錄下，複製到該目錄下，如下：



### (C) 設定 CLASSPATH 環境變數

接著，必須告訴 Java 方法庫在哪裡。由『桌面』=>『電腦』按又鍵=>『內容』=>『進階系統設定』=>『環境變數』=>『新增』，則『變數名稱』輸入 CLASSPATH，『變數值』輸入『.;C:\JAR\mysql-connector-java-5.1.45-bin.jar;%CLASSPATH%』，如下：



## 11-4-2 開發 JDBC 連結 MySQL 步驟

以 Java 連結 MySQL 程式設計，由下列幾個步驟完成：

- (a) 載入 JDBC 驅動程式：利用 `Class.forName("com.mysql.jdbc.Driver")`。
- (b) 開啟連線：`Connection conn = DriverManager.getConnection("URL/資料庫", "帳號", "密碼")`。URL 如：`"jdbc:mysql://localhost:3306/bank_db"`。
- (c) 產生 SQL 執行物件：`Statement st = conn.createStatement()`。
- (d) 資料表更新命令：`st.executeUpdate(SQL 命令)`，如傳回 1 表執行成功；傳回 0 表執行失敗。如：`st.executeUpdate("insert into account value("345678912", "張友德", "2000")`。
- (e) 查詢命令：`ResultSet rs = st.executeQuery(SQL 命令)`。
- (f) 取得資料表訊息：`ResultSetMetaData rsmeta = rs.getMetaData()`；取得資料表訊息，如下：

1. 欄位數量：int cols = rsmeta.getColumnCount();
  2. 欄位名稱：String col\_name = rsmeta.getColumnLable(i);
  3. 欄位資料型態：String col\_tye = rsmeta.getColumnType(i);
- (g) 游標移動：rs.next()，為 0 表示結束，1 表示到下一筆。
- (h) 擷取欄位 i 資料：rs.getString(i)、rs.Date(i)、rs.getTime(i)、getDouble(i)、getInt(i)，依欄位資料型態擷取。
- (i) 關閉資料庫連線：conn.close();。

### 11-4-3 連結 MySQL 程式範例

#### (A) 程式功能：Ex11\_1.java

請先在 Windows 7 系統上安裝 AppServ 套件，並匯入 Bank\_db 資料庫(下一節說明)。  
此程式連結到本機 localhost 上的 MySQL 伺服器內的 Bank\_db 資料庫  
(jdbc:mysql://localhost:3306/bank\_db)，並查詢 Account 資料表內容，再印出。

```
01 //Ex11_1.java
02 import java.sql.*;
03 public class Ex11_1 {
04     public static void main(String[] args) {
05         String driver = "com.mysql.jdbc.Driver"; // 連結 MySQL 驅動程式
06         Connection conn = null;
07         Statement st = null;
08         ResultSet rs = null;
09         ResultSetMetaData rsmeta = null;
10         String url = "jdbc:mysql://localhost:3306/bank_db";
11             // 連結主機 localhost:3306 伺服器內 Bank_db 資料庫
12
13         String user = "root"; // 帳戶 root
14
15         String password = "12345678"; // root 密碼
16         try {
17             Class.forName(driver);
18             conn = DriverManager.getConnection(url, user, password);
19             // 連結到主機 MySQL 內 Bank_db 資料庫
```

```
20         System.out.println("成功連結 Bank_db 資料庫");
21
22         st = conn.createStatement(); // 產生 SQL 敘述物件
23         rs = st.executeQuery("select * from account");// 下達 SQL 命令
24
25         while (rs.next()) { // 擷取下一筆紀錄
26
27             System.out.printf("%s\t", rs.getString("ID")); // ID 欄位內容
28             System.out.printf("%s\t", rs.getString("name")); // name 欄位內容
29             System.out.printf("%s\t", rs.getInt("balance")); // balance 內容
30             System.out.printf("\n");
31         }
32         conn.close(); // 關閉資料庫連線
33     }
34     catch (ClassNotFoundException e) {
35         e.printStackTrace();
36     }
37     catch (SQLException e) {
38         e.printStackTrace();
39     }
40 }
41 }
42 }
43 }
```

執行結果：

```
D:\Java2_book\chap11>javac -encoding utf-8 Ex11_1.java
D:\Java2_book\chap11>java Ex11_1
Thu Dec 07 15:26:37 CST 2017 WARN: Establishing SSL connection without
s
.....
成功連結 Bank_db 資料庫
1234512345783  林郁評  220000
1234567891232  郭大豪  150000
2134567812343  林惠容  2300000
3412345671238  劉嫻君  1100000
4512341231235  林秀氣  3021000
9872345123902  陳立人  4301234
```

### 11-4-4 SQL 命令匯集

命 令	格 式	命 令	格 式
Select	Select "欄位" From "表格名";	Distinct	Select Distinct "欄位" From "表格名";
Where	Select "欄位" From "表格名" Where "condition";		And/OR
	In	SELECT "欄位" FROM "表格名" WHERE "欄位" IN ('值 1', '值 2', ...);	
Like	SELECT "欄位" FROM "表格名" WHERE "欄位" LIKE {模式};	Between	
	Count	SELECT COUNT("欄位") FROM "表格名";	Order By
Group by	SELECT "欄位 1", SUM("欄位 2") FROM "表格名" GROUP BY "欄位 1"	Having	
Create	CREATE TABLE "表格名" ( "欄位 1" "欄位 1 資料種類", "欄位 2" "欄位 2 資料種類", ... );	Insert	INSERT INTO "表格名" ("欄位 1", "欄位 2", ...) VALUES ("值 1", "值 2", ...);
	Drop		DROP TABLE "表格名";
Truncate	TRUNCATE TABLE "表格名";	(Not) Like	*、_(底線)、%、[ABC]、Pattern
Update	UPDATE "表格名" SET "欄位 1" = [新值] WHERE "條件";	運算	+、-、*、/、%
	比較	=、<>、<、>、>=、between	
聚合	Count(欄位)、Ave、Sum、Max、Min		

## 11-4 Bank\_db 資料庫介紹

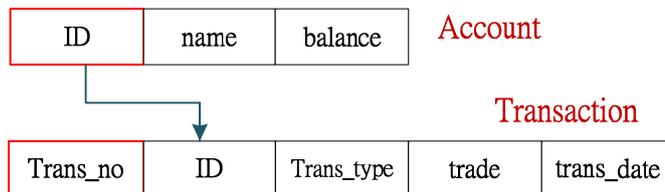
### 11-4-1 系統功能

『藝術銀行』期望製作一套『活期儲蓄存款系統』，期望包含有下列功能：

- (1) 顯示帳戶：顯示系統每一帳戶的姓名、帳號與餘額。
- (2) 存款：客戶提款功能，選擇後會要求輸入客戶帳號，再要求輸入存款金額，執行後會顯示目前餘額。
- (3) 提款：選擇後會要求輸入客戶帳號，再要求輸入提款金額，執行後會顯示目前餘額。
- (4) 查詢餘額：選擇後會要求輸入客戶帳號，再顯示餘額。
- (5) 查詢交易額：可供查詢某一帳戶存款/領款的交易過程。

### 11-4-1 資料庫 E-R 圖

吾人將此資料庫定名為 Bank\_db，其 E-R 關係圖如下所示，其中包含兩個資料表。



#### (A) 帳戶資料表 - Account

Account 資料表包含三個欄位(如 7-5-1 節 Account.class 類別)：

1. ID：帳戶編碼(主鍵)、CHAR(20) 格式
2. Name：帳戶姓名、CHAR(20) 格式
3. Balance：帳戶餘額、int 格式

#### (B) 交易資料表 - Transaction

Transaction 資料表有四個欄位(請自行設計 transaction.class 類別)：

1. Trans\_ID：交易編碼(主鍵)、int 格式。
2. ID：帳戶編碼(外來鍵)、CHAR(20) 格式

3. trans\_type：1: 表存款、0: 表提款、int 格式
4. trade：交易額、int 格式
5. trans\_date：交易日期/時間。datetime 格式

### 11-5-3 匯入 Bank\_db 資料庫

#### (A) 建立 Bank\_db 資料庫

由 <http://localhost> 進入 Mysql 之後，選擇『資料庫』並建立 Bank\_db 資料庫，如下：



#### (B) 匯入 Bank\_db 資料庫

請在教學網站上下載 Bank\_db.sql，再將它匯入 Bank\_db 資料庫內。如下：



#### (C) 瀏覽資料庫內容



### (D) 瀏覽 account 資料表內容

ID	name	balance
1234512345783	林郁評	220000
1234567891232	郭大豪	150000
2134567812343	林惠容	2300000
3412345671238	劉嫻君	1100000
4512341231235	林秀氣	3021000
9872345123902	陳立人	4301234

### (D) 瀏覽 transaction 資料表內容

trans_no	ID	trans_type	trade	trans_date
1	1234567891232	1	2000	2017-11-14 00:00:00
2	1234512345783	1	12000	2017-11-15 00:00:00
3	3412345671238	0	20000	2017-10-20 00:00:00
4	1234567891232	0	10200	2017-10-21 00:00:00
5	1234512345783	1	2000	2017-10-22 00:00:00
6	1234567891232	1	2000	2017-10-23 00:00:00

## 11-6 專題製作 – 儲蓄存款系統

『藝術銀行』期望製作一套『活期儲蓄存款系統』，程式設計師很難了解系統需求，因此依照可能狀態分段製作，每一階段完成並得到上級同意後，再往下一階段實現，吾人將各步驟分批實施如下。

### 11-6-1 範例研討：步驟(1)建立存款帳戶規格

## (A) 系統功能：Ex11\_2.java、account.java

系統需要建立客戶帳戶。每一存款帳戶包含：姓名 (String name)、帳戶 (String No) 與存款餘額 (int balance)，各項資料限制如下：

- (1) 帳戶：由 12 個數字與 1 個檢查碼所構成 (合計 13 碼)，檢查碼計算方式是利用一個加權系列：1 2 1 2 1 2...等 交替變換，帳號第一個數字乘以 1、第二位乘以 3、第三位乘以 1...等依此類推，再將所乘的結果相加 (= total)，再取 10 的餘數 (value = total % 10)，檢查碼即是 10 減該餘數 (= 10 - value)。
- (2) 存款餘額：如果取款後，餘額會少於 0，則會拒絕提款。

請製作一個帳戶類別 (Account.class) 是其具有上述功能，再至作主程式引用該類別，驗證是否正常。當建立帳戶時，只要輸入 12 個帳號，系統自動產生檢查碼；處理帳戶而輸入帳號，系統也會檢查檢查碼是否正確。期望驗證帳戶類別功能的結果如下：

```
D:\Java2_book\chap11>java Ex11_2
***建立新帳戶***
請輸入姓名=>粘添壽
請輸入帳號(12 位數字) =>456789130123
****建立完成****
完整的新帳戶(13 碼)為=>45678913012310
請輸入存款金額 =>50000
餘額 = 50000
列印帳戶資料
帳戶姓名： 粘添壽
帳戶號碼(13 碼)：45678913012310
餘額= 50000
```

## (B) 帳戶類別(Account.class)：

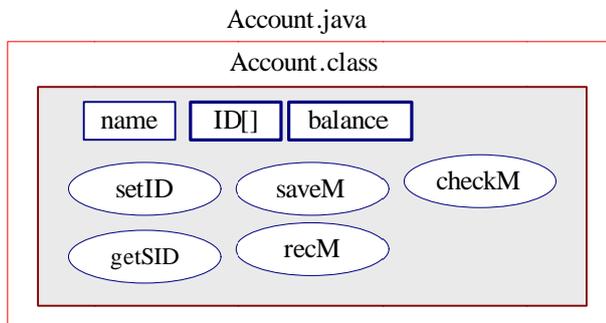
首先必須建立帳戶類別，但它必須與 Bank\_db 資料庫內 Account 資料表相配合，Account 表格的結構如下圖：

名稱	型態	編碼與排序	屬性	空值	預設值	備註	額外資訊
ID 	char(20)	utf8_unicode_ci		否	無		
name	char(20)	utf8_unicode_ci		是	NULL		
balance	int(11)			是	NULL		

各欄位功能如下：

- (1) 欄位 ID：帳號，資料型態是 CHAR(20)，但程式內必須經過運算，Java 程式採用 int[13] 格式。
- (2) 欄位 name：帳戶姓名，資料型態是 CHAR(20)，Java 程式採用 String 格式。
- (3) 欄位 balance：餘額，資料型態是 int，Java 程式採用 int 格式。

另外 ID 是 12 個整數再加入 1 個檢查碼，皆必須符合檢查碼計算方法。餘額 balance 不可以小於 0，否則拒絕存取，因此須將 ID 與 balance 設定為私有變數，必須透過物件方法存取。Account 類別結構如下圖所示。



**圖 11-2 Account.java 程式架構**

Account 程式範例如下：

```

01 //Account.java
02 /* 設定客戶姓名、帳號(ID) 與存款餘額 */
03 import java.io.*;
04 import java.util.Scanner;
05
06 class Account{
07     String name;
08     private int[]ID = new int[13];
09     private int balance;
  
```

```
10 // 設定帳號
11     int setID(String ID_Str) {
12         if(ID_Str.length() !=12){
13             System.out.printf("需 12 個字元,請重新輸入!!\n");
14             return 0;
15         }
16     }
17     else{
18         Scanner s = new Scanner(ID_Str).useDelimiter("");
19         int total = 0;
20         for(int i=0;i<12;i++){
21             ID[i]=s.nextInt();
22             if((i+1)%2==0)
23                 total+=ID[i];
24             else
25                 total+=ID[i]*2;
26         }
27         ID[12] = (10-(total%10));
28         System.out.print("****建立完成****\n 完整的新帳戶(13 碼)為=>");
29         for(int i=0;i<13;i++)
30             System.out.print(ID[i]);
31         System.out.println();
32         return 1;
33     }
34 }
35
36 // 取得帳號
37     int[] getID(){
38         return ID;
39     }
40 }
41 // 寫入帳號
42     int writeID(String ID_STR1) {
43         if(ID_STR1.length() !=13){
44             System.out.printf("長度不對\n");
45             return 0;
46         }
47     }
48     else{
49         Scanner s = new Scanner(ID_STR1).useDelimiter("");
50         int total = 0, check, check_R;
51         for(int i=0;i<12;i++){
52             ID[i]=s.nextInt();
53             if((i+1)%2==0)
54                 total+=ID[i];
55             else
56                 total+=ID[i]*2;
```

```
56         }
57         check_R = s.nextInt();
58         check = 10-(total%10);
59         if (check == check_R) {
60             ID[12] = check;
                return 1;
        } else {
                System.out.printf("檢查碼不對\n");
                return 0;
        }
    }
}

// 存款
int saveM(int money){
    balance = balance + money;
    return balance;
}

// 取款
int recM(int money) {
    int m = balance - money;
    if (m >=0) {
        balance = m;
        return balance;
    }
    else {
        return -1;
    }
}

// 查詢餘額
int checkM(){
    return balance;
}
}
```

經編譯後，產生一只 Account.class，如下：

```
D:\Java2_book\chap11>javac -encoding utf-8 Account.java

D:\Java2_book\chap11>dir/b Acc*
Account.class
Account.java
```

### (C) 主類別(Ex11\_2.class)：

我們寫一個簡單程式來驗證 Account 類別是否能滿足所需，程式範例如下：

```
01 //Ex11_2.java
02 import java.util.*;
03 public class Ex11_2{
04     public static void main(String[] args) {
05         Scanner keyin = new Scanner(System.in);
06         String ID_Str;
07         Account customer = new Account();
08         System.out.printf("***建立新帳戶***\n 請輸入姓名=>");
09         customer.name = keyin.nextLine();
10         System.out.printf("請輸入帳號(12 位數字) =>");
11         ID_Str = keyin.nextLine();
12         int flag = customer.setID(ID_Str);
13         if (flag == 0)
14             return;
15         System.out.printf("請輸入存款金額 =>");
16         int money = keyin.nextInt();
17         int balance = customer.saveM(money);
18         System.out.printf("餘額 = %d\n", balance);
19         /* 列印帳戶清單 */
20         System.out.printf("\n 列印帳戶資料\n");
21         System.out.printf("帳戶姓名： %s \n", customer.name);
22         int[] ID = customer.getID();
23         String ID_S = "";
24         for (int i=0; i<13; i++)
25             ID_S = ID_S + ID[i];
26         System.out.printf("帳戶號碼(13 碼)： %s", ID_S);
27         System.out.printf("\n");
28         System.out.printf("餘額= %d\n", customer.checkM());
29     }
30 }
31 }
```

## 11-6-2 範例研討：步驟(2) 建立帳戶與異動管理

### (A) 系統功能：Ex11\_3.java、Transaction.java

無論建立新帳戶、帳戶存款或帳戶提款，都有可能變更到帳戶內的餘額(balance)，除了需

登錄於帳戶資料表內(Account)，也必須將交易過程登錄於交易檔(transaction)內，因此，在 Bank\_db 資料庫內有 Account 與 transaction 兩資料表，前者已介紹過，接下來介紹 transaction，其結構如下圖：

#	名稱	型態	編碼與排序	屬性	空值	預設值	備註	額外資訊
1	trans_no 	int(11)			否	無		AUTO_INCREMENT
2	ID	char(20)	utf8_unicode_ci		否	無		
3	trans_type	int(11)			是	NULL		
4	trade	int(11)			是	NULL		
5	trans_date	datetime			是	NULL		

銀行內所有帳戶的交易狀況皆登錄於此資料表內，每一筆紀錄說明每一帳戶的交易過程，個欄位說明如下：

- (a) trans\_no：交易編碼，每一筆交易都有一個唯一識別值，是此表格的主鍵。
- (b) ID：帳戶編碼，此交易資料的帳戶。
- (c) trans\_type：交易型態，1 表存款；0 表提款。
- (d) trade：交易金額。
- (e) trans\_date：交易時間日期。

依照上述規格，吾人將 transaction 類別宣告如下：(Transaction.java)

```

01 //Transaction.java
02 /* 設定交易檔規格：trans_no, ID, trans_type, trade 與 date*/
03
04 class Transcation{
05     int trans_NO;
06     String ID;
07     int trans_type;
08     int trade;
09     String trans_date;
10 }

```

經編譯後，產生一只 Transaction.class，如下：

```
D:\Java2_book\chap11>javac -encoding utf-8 Transaction.java

D:\Java2_book\chap11>dir/b Tr*
Transaction.class
Transaction.java
```

### (B) 主類別(Ex11\_3.class)：

我們編寫一只程式可以建立一個新帳戶，並存入開戶金額，接著，再由 Account 表格內讀出該帳戶資料，並由 Transaction 表內讀取該帳戶的異動資料。其中將資料插入資料表語法如下：

```
Insert Into account
    Value("ID", "name", balance);
Insert Into transaction
    Value("trans_ID", "ID", "trans_type", "trade", "trans_date");
```

查詢資料表語法如下：

```
Select *
From account
Where ID = "帳戶編碼";
Select *
From transaction
Where ID = "帳戶編碼";
```

期望操作結果如下：

```
D:\Java2_book\chap11>javac -encoding utf-8 Ex11_3.java

D:\Java2_book\chap11>java Ex11_3
***建立新帳戶***
請輸入姓名=>柯大仙
請輸入帳號(12 位數字) =>341234871912
****建立完成****
完整的新帳戶(13 碼)為=>3412348719128
請輸入存款金額 =>892102
餘額 = 892102

列印帳戶資料                                [直接列印新增帳戶資料]
```

```
帳戶姓名： 柯大仙
帳戶號碼(13 碼)： 3412348719128
餘額= 892102
Sat Dec 09 09:16:50 CST 2017 WARN: Establishing SSL connection without server's
.....
成功連結 Bank_db 資料庫           [顯示資料庫連結成功]
帳戶 account 新增正常             [顯示資料庫 account 帳戶新增成功]
交易檔 transaction 新增正常       [顯示資料庫 transaction 交易新增成功]

顯示新增帳戶 account 資料：       [由資料庫讀出新增帳戶資料]
3412348719128 柯大仙 892102

顯示新增交易 transaction 資料：   [由資料庫讀出新增交易資料]
交易編碼: 26
帳戶編碼: 3412348719128
交易種類： 存款
交易金額： 892102
交易日期： 2017-12-09 09:16:50.0
```

程式範例如下：**(Ex11\_3.java)**

```
01 //Ex11_3.java
02
03 import java.util.*;
04 import java.sql.*;
05 import java.text.*;
06 public class Ex11_3{
07     /* 宣告資料庫連結訊息 */
08     static String driver = "com.mysql.jdbc.Driver";
09     static Connection conn = null;
10     static Statement st = null;
11     static ResultSet rs = null;
12     static ResultSetMetaData rsmeta = null;
13     static String url = "jdbc:mysql://localhost:3306/bank_db";
14     static String user = "root";
15     static String password = "12345678";
16
17     public static void main(String[] args) throws SQLException {
18         Scanner keyin = new Scanner(System.in);
19
```

```
20
21     /* 建立新帳戶 */
22         String ID_Str;
23         Account customer = new Account();
24         System.out.printf("***建立新帳戶***\n 請輸入姓名=>");
25         customer.name = keyin.nextLine();
26         System.out.printf("請輸入帳號(12 位數字) =>");
27         ID_Str = keyin.nextLine();
28         int flag = customer.setID(ID_Str);
29         if (flag == 0)
30             return;
31
32         System.out.printf("請輸入存款金額 =>");
33         int money = keyin.nextInt();
34         int balance = customer.saveM(money);
35         System.out.printf("餘額 = %d\n", balance);
36
37
38     /* 列印帳戶清單 */
39
40         String name = customer.name; // 新增帳戶名稱
41         balance = customer.checkM(); // 新增帳戶餘額
42         int[] ID = customer.getID();
43         String ID_S = "";
44         for (int i=0; i<13; i++)
45             ID_S = ID_S + ID[i]; // 新增帳戶號碼
46
47         System.out.printf("\n 列印帳戶資料\n");
48         System.out.printf("帳戶姓名： %s \n", name);
49         System.out.printf("帳戶號碼(13 碼)： %s\n", ID_S);
50         System.out.printf("餘額 = %d\n", balance);
51
52     /* 連結 bank_db 資料庫 */
53     try {
54         Class.forName(driver);
55         conn = DriverManager.getConnection(url, user, password);
56         System.out.println("成功連結 Bank_db 資料庫");
57         st = conn.createStatement();
58         String SQL;
59
60     /* 將帳戶寫入 account 資料表 */
61         SQL = String.format("INSERT INTO account VALUES('%s', '%s', '%d'",
62             ID_S, name, balance);
63         if (st.executeUpdate(SQL) == 1)
```

```
66         System.out.println("帳戶 account 新增正常");
67
68     /* 由異動訊息寫入 transaction 資料表*/
69         Transaction trans = new Transaction();
70         java.util.Date now = new java.util.Date();
71         SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
72         trans.trans_date = sdf.format(now);
73         trans.trans_type = 1;
74         trans.trade = balance;
75         trans.ID = ID_S;
76         SQL = String.format("INSERT INTO transaction(ID, trans_type, trade, trans_date)" +
77     "VALUES('%s', '%d', '%d', '%s')", trans.ID, trans.trans_type, trans.trade, trans.trans_date);
78         if (st.executeUpdate(SQL) == 1)
79             System.out.println("交易檔 transaction 新增正常");
80
81     /* 由 account 讀出新增帳戶的資料 */
82         SQL = String.format("Select * From account where ID = %s", ID_S);
83         rs = st.executeQuery(SQL);
84         System.out.printf("\n 顯示新增帳戶 account 資料：\n");
85         while (rs.next()) {
86             System.out.printf("%s\t", rs.getString("ID"));
87             System.out.printf("%s\t", rs.getString("name"));
88             System.out.printf("%s\t", rs.getInt("balance"));
89             System.out.printf("\n");
90         }
91
92     /* 由 transaction 讀出新增交易資料 */
93         SQL = String.format("Select * From transaction where ID = %s", ID_S);
94         rs = st.executeQuery(SQL);
95         System.out.printf("\n 顯示新增交易 transaction 資料：\n");
96         while (rs.next()) {
97             System.out.printf("交易編碼： %d\n", rs.getInt("trans_no"));
98             System.out.printf("帳戶編碼： %s\n", rs.getString("ID"));
99             System.out.printf("交易種類： ");
100            if (rs.getInt("trans_type") == 1)
101                System.out.printf("存款 \n");
102            else
103                System.out.printf("提款 \n");
104            System.out.printf("交易金額： %d\n", rs.getInt("trade"));
105            System.out.printf("交易日期： %s\n", rs.getString("trans_date"));
106        }
107        conn.close();
108
109
110
111
```

```
112     }
113     catch (ClassNotFoundException e) {
114         e.printStackTrace();
115     }
116     catch (SQLException e) {
117         e.printStackTrace();
118     }
119 }
}
```

### 11-6-3 自我挑戰：步驟(3)建立儲蓄存款系統

#### (A) 系統功能：PM11\_1.java、Account.class、Transaction.class

帳戶規格 (Account.class)與交易檔規格(Transaction.class) 經過驗證可行之後，期望您進一步請幫『藝術銀行』建立一套『儲蓄存款系統』的雛形系統，包含有下列功能：

- (1) 將客戶資料儲存於 Bank\_db 資料庫的 account 資料表內。
- (2) 客戶開戶、存款或領款都必須登錄於 transaction 資料表內。
- (3) 顯示帳戶：顯示系統每一帳戶的姓名、帳號與餘額。
- (4) 存款：客戶提款功能，選擇後會要求輸入客戶帳號，再要求輸入存款金額，執行後會顯示目前餘額。
- (5) 提款：選擇後會要求輸入客戶帳號，再要求輸入提款金額，執行後會顯示目前餘額。
- (6) 查詢餘額：選擇後會要求輸入客戶帳號，再顯示餘額。
- (7) 查詢客戶交易紀錄：選擇後會要求輸入客戶帳號，再顯示該客戶所有交易紀錄。
- (8) 刪除帳戶：刪除某一帳戶，並刪除它的交易檔資料。

任何操作皆必須隨時寫入 Bank\_db 資料庫內，期望系統有 7 種功能如下：

```
D:\Java2_book\chap11>javac -encoding utf-8 PM11_1.java
D:\Java2_book\chap11>java PM11_1
```

== 藝術銀行 活期儲蓄系統 ==

- |             |             |            |
|-------------|-------------|------------|
| (1) 顯示所有帳戶  | (2) 新 增帳 戶  | (3) 帳戶提款作業 |
| (4) 帳戶存款作業  | (5) 帳戶查詢餘額  | (6) 查詢帳戶交易 |
| (7) 刪 除 帳 戶 | (8) 離 開 系 統 |            |

請選擇工作項目 =>

## (B) 程式提示

程式片段如下，請同學將它補完成。

```
01 //PM11_1.java
02
03 import java.util.*;
04 import java.sql.*;
05 import java.text.*;
06 public class PM11_1{
07     /* 宣告資料庫連結訊息 */
08     static String driver = "com.mysql.jdbc.Driver";
09     static Connection conn = null;
10     static Statement st = null;
11     static ResultSet rs = null;
12     static ResultSetMetaData rsmeta = null;
13     static String url = "jdbc:mysql://localhost:3306/bank_db";
14     static String user = "root";
15     static String password = "12345678";
16
17     /*宣告 Java 相關輸入/輸出 */
18     static Scanner keyin = new Scanner(System.in);
19
20     public static void main(String[] args) throws SQLException {
21
22         mainMenu();
23         int select = keyin.nextInt();
24         while (select !=8) {
25             switch (select) {
26                 case 1:          /* 顯示所有帳戶資料 */
27
28                     dispAccount();
29                     break;
30                 case 2:          /* 新增帳戶 */
31
32                     addAcc();
33                     break;
34                 case 3:          /* 帳戶提款作業 */
```

```
35         recMoney();
36         break;
37     case 4:          /* 帳戶存款作業 */
38         saveMoney();
39         break;
40     case 5:          /* 查詢帳戶餘額 */
41         checkMoney();
42         break;
43     case 6:          /* 查詢帳戶交易資料 */
44         checkTrans();
45         break;
46     case 7:          /* 刪除帳戶 */
47         delAcc();
48         break;
49     default:
50         System.out.printf("錯誤輸入, 請重新選擇 !!\n");
51     }
52     mainMenu();
53     select = keyin.nextInt();
54     keyin.nextLine();
55 }
56
57 /* 顯示主選單 */
58
59 public static void mainMenu() {
60     System.out.printf("\n== 藝術銀行 活期儲蓄系統 ==\n");
61     System.out.printf("(1) 顯示所有帳戶\t");
62     System.out.printf("(2) 新增帳戶\t");
63     System.out.printf("(3) 帳戶提款作業\n");
64     System.out.printf("(4) 帳戶存款作業\t");
65     System.out.printf("(5) 帳戶查詢餘額\t");
66     System.out.printf("(6) 查詢帳戶交易\n");
67     System.out.printf("(7) 刪除帳戶\t");
68     System.out.printf("(8) 離開系統\n");
69     System.out.printf("請選擇工作項目 => ");
70 }
71
72 public static void dispAccount() { /* 顯示所有帳戶資料 */
73
74
75
76
77
78
79
80
```

```
81         // 補完程式
82
83     }
84     public static void addAcc() {          /* 新增帳戶 */
85         // 補完程式
86
87
88
89     }
90     public static void recMoney() {      /* 帳戶提款作業 */
91         // 補完程式
92
93
94     }
95     public static void saveMoney() {    /* 帳戶存款作業 */
96         // 補完程式
97
98
99     }
100    public static void checkMoney() {    /* 查詢帳戶餘額 */
101        // 補完程式
102
103
104    }
105    public static void checkTrans() {    /* 查詢帳戶交易資料 */
106        // 補完程式
107
108
109    }
110    public static void delAcc() {        /* 刪除帳戶 */
111        // 補完程式
112
113    }
114 }
```

### (C) 評分辦法

每一同學都需獨立完成，並以自己的名字當作銀行名稱，也以自己姓名建立帳戶，並執行各項功能(除刪除帳戶外)。可以逐項完成，每一位同學皆須在課堂上操作給老師評分。