書名：Java SE7 技術手冊

出版社：碁峯

讀書心得報告作者：余忠訓

**Chapter 1：Java SE簡介**

Java從問世以來有很多版本，現在大多是SE7和SE8使用較多

* JVM（虛擬機器）執行Java開發程式所需要的執行編譯工具
* JRE（執行環境）運行Java程式使用的環境執行介面
* JDK （開發環境）在執行環境介面額外多增加開發程式與API的工具
* Java進行編譯時，需要有System.java編譯為System.class，才能使用類別（class）上out物件（Object）的println方法（Method）。

**Chapter 2：從JDK到IDE**

* 開始撰寫Java程式，需要先建立專案，建立和定義類別（Class）、區塊（Block）、Main( )方法，在main裡撰寫陳述（Statement）

**Chapter 3：基礎語法：型態、變數與運算子**

* 型態：基本型態（Primitive type）、類別型態（Class type），主要有整數、位元組、浮點數、字元與布林。
* Java裡變數名稱命名，使用駝峰式（Camel case）命名法，例如：FirstNumber
* 運算流程控制：if、else；switch條件式；for迴圈；while迴圈；break；continue

**Chapter 4：類別與物件**

* 要產生物件（Object）之前須先建立類別（Class），類別是物件的設計圖，物件是類別的實例（Instance）。
* 新建立的稱為物件，未建立的名稱則是綁在物件上稱為參考（Reference）

例如：

* Clothes c1; 🡪稱為參考
* Clothes c1 = new Clothes( ); 🡪稱為物件
* 物件指定與相等性
* = 是用來指定參考名稱參考某物件（將某個名牌綁到某個物件）
* == 是用在比較兩個參考名稱是否參考同一物件（比較兩個名牌是否綁到同一物件）
* 建立陣列物件

例如建立長度為10的陣列物件，int[ ] scores = new int[10];

如果預設初始値不符合需求，可使用java.util.Arrays的fill( )來設定新建陣列的元素値。

* 增強for語法

for (int[ ] row : cords) { } 可以簡短撰寫for迴圈的功能表示式。

* 陣列複製索引的元素，可用System.arraycopy()
* 字串特性
* 字串常量與字串池
* 不可變動（Immutable）字串

以雙引號””包圍的內容稱為字串，寫下的字串稱為字串常量（String Literal），當比較字串實際字元內容是否相同，不能使用==，而是用equals()

**Chapter 5：封裝**

封裝可將class中需使用的變數進行定義來使用，例如

Class CashCard {

String number;

int balance;

int bonus;

}

重覆使用的流程可用建構式（Constructor），在物件成員之前加上this關鍵字。

Class CashCard {

String number;

int balance;

int bonus;

CashCard(String number, int balance, int balance) {

this.number = number;

this.balance = balance;

this.bonus = bonus;

}

}

* 類別Public可公開使用，Private只能私有使用。
* 被宣告為static的成員，不會讓個別物件擁有，是屬於類別。
* 傳値呼叫是指在物件的( )中輸入參數，會回傳後改變此物件。

**Chapter 6：繼承與多型**

* 繼承是用於避免多個類別間重複定義共同行為，以子類別繼承父類別，避免重複的行為定義。

例如在子類別的public … { this.blood = blood; } 以this將父類別內blood變數的功能繼承下來，不需要再重複定義相同的行為定義。

* 多型在Java中，子類別只能繼承一個父類別，除了可避免重複的行為定義外，會有is-a的關係，例如子類別的SwordMan這角色只會繼承父類別的SwordMan的角色和功能，不會繼承到其他的角色。
* 抽象方法（abstract method）不用撰寫{ }區塊，直接; 結束即可。
* protected 可將子類別的變數直接擁有父類別變數的存取權限。
* super() 可指定父類別中的某建構式來執行，但與this只能擇一呼叫。

**Chapter 7：介面與多型**

* 類別要實作介面，必須使用implements，對介面中定義的方法有兩種

1. 實作介面中定義的方法
2. 再度將該方法標式為abstract

以Java語意來說，繼承會有「是一種」關係，實作介面則表示*「擁有行為」*，但不會有「是一種」的關係。

* 類別可以同時繼承某個類別，並實作某些介面。

**Chapter 8：例外處理**

* 例外處理是針對該錯誤做出一些處理，例如嘗試回復正常流程、進行日誌（Logging）紀錄、以某種型是提醒使用者。
* 使用try、catch，在try{ }裡作執行，若是發生錯誤會往外丢進catch{ }處理
* 不同的錯誤（Error）、例外（Exception）和執行例外（Runtime Exception）等皆屬於可回拋（Throwable）的子架構。
* 堆疊追蹤（Stack Trace）是直接呼叫例外物件的printStackTrace( )，假設前提是不能有程式碼私吞例外的行為，顯示每個步驟發生的蹤跡，以往回尋找發生例外的根源。
* Finally是無論try區塊有無發生例外都會一定會被執行，通常會在此區塊用來使用.close( )關閉檔案。

**Chapter 9：Collection與Map**

* 收集（Collection）API介面由List、Set、Queue等子架構組成，主要功能是收集時記錄每個物件的索引順序，並可依索引取回物件。
* 具有索引的List
* 內容不重複的Set
* 支援佇列操作的Queue
* 走訪物件的Iterator
* 泛型是定義收集此物件之型態，讓物件重新扮演該型態。
* MAP是根據鍵（Key）與值（Value）作為存放和取值的物件，放入時使用.put( )，取出時使用.get( )

**Chapter 10：InputStream與OutputStream**

串流設計觀念

* 來源🡪經過資料的串流物件🡪目的地

分為InputStream和OutputStream，把資料經由輸入和輸出作為傳送，提高存取和輸出的效率，需要使用Buffer暫存區讓記憶體作為緩衝的區域，減少直接從來源讀取資料的次數。

**Chapter 12：通用API**

* 日誌格式：Logger會顯示每一步驟操作的訊息
* Handler是負責日誌Logger的輸出
* 地區與時間：使用Locale、Date和DateFormat
* 日曆格式：Calendar

**Chapter 14：整合資料庫**

* JDBC是用於SQL的標準介面，透過JDBC API作為和SQL資料庫的溝通和通訊協定的橋梁。
* JDBC的驅動（Driver）分為四個類型，Type 1、Type 2、Type 3、Type 4，最常見的驅動程式類型是Type 4，驅動程式會將JDBC呼叫轉換為與資料庫特定的網路協定，以進行溝通操作。

**結論**

在進行Java撰寫時，了解程式的底層系統架構、編譯過程和執行運作，開始撰寫後，事先擬定程式邏輯的流程，在架設整體架構後再撰寫細部的程式內容會順暢許多，本書從基本程式架構、語法、變數、型態、類別和方法等，解析實例程式撰寫和說明，在Java作業中有確實使用上並實際演練，增加自己撰寫程式上的印象和觀念。