

Chapter 9

9.1 導論

9.2 類別圖與物件圖

9.3 物件結構塑模

9.4 建構類別圖與物件圖

9.5 結論

附錄 9A：應用 Enterprise Architect 繪製類別圖之實作

附錄 9B：套件圖

附錄 9C：模型圖

本章習題

參考文獻

物件結構塑模

本章學習重點

詳讀本章，你至少能瞭解：

- 物件導向系統分析與設計中之物件結構塑模工作。
- 何謂類別圖與物件圖及其用途。
- 類別圖與物件圖之建構步驟與準則。
- 如何封裝類別。
- 如何以類別圖與物件圖進行物件結構塑模。

9.1 導 論

在物件導向系統發展過程中，完成了使用個案塑模後，接下來便可進行物件的互動行為塑模、物件結構塑模等塑模活動以開始架構系統（請參考第 3 章圖 3-8）。本章介紹之物件結構塑模主要以類別圖或物件圖表達系統之物件靜態結構（其中以類別圖為主），然而於真實世界中，物件的數量相當龐大，為降低問題複雜度，在物件結構塑模時大多採用類別而較少使用物件來進行塑模。若需瞭解系統套件或元素的組織方式，請參考套件圖；如需呈現系統的抽象或特定觀點，相關模型圖見附錄 9B 與 9C。

類別圖主要用於表達系統內部的靜態結構，並描述系統中物件的類型 (Type)、類型間及與子類型間之靜態關係。此外，類別圖還須表示類別內部的屬性、操作及物件連結所應遵守的限制等 (Fowler, 2018)。物件圖則用於描述一系統於某一時間點的靜態結構，可用來表達一系統的複雜資料結構或藉由一時間序列的系統影像來表達系統行為。上述之進階概念可參考 Booch (2017)、與 Rumbaugh (2010) 之著作，或參考 UML 2.5.1 版的規格書 (OMG, 2017)。

本章將先介紹類別圖與物件圖之元件，再介紹物件結構塑模的方法論，接著延續第 5 章便當王網路訂購系統案例之使用個案（需求）塑模，進一步說明如何由需求塑模結果進行物件結構塑模。

9.2 類別圖與物件圖

物件與類別之關係為：類別是描述物件的結構，並定義了組成物件的屬性與欄位，而物件是類別的執行個體。換句話說，類別就像是高爾夫球模具，定義了高爾夫球頭之特性、參數，而物件就是利用模具製作出來之高爾夫球頭。**類別圖 (Class Diagram)** 有兩個主要元件：類別與類別間之關係。物件圖也有兩個主要元件：物件與連結線。茲分別介紹如下。

9.2.1 類別圖—類別

一群相關物件的定義、描述或樣板 (Definition, Description or Template) 稱為一個**類別**，類別具有名稱、屬性與操作，同一類別之物件具有相同屬性與操作。類別常由矩形表示，該矩形之四邊為實線，如圖 9-1 所示。

Figure
圖 9-1 類別之表示符號

● 類別種類

從執行觀點，類別可分為永存類別 (Persistent Class) 與暫存類別 (Transient Class) 兩種。永存類別指的是當程式執行後，該物件之資料須被儲存在資料庫者；而暫存類別則是指當程式執行後，該物件之資料不須被儲存在資料庫者且該物件會被刪除者。從實作觀點，類別圖中之類別可分為實體類別 (Entity Class)、介面類別 (Interface Class) 和控制類別 (Control Class) 三種（關於各類別之找尋，可參考第 6 章之物件找尋準則）。分別介紹如下：

1. 實體類別

實體類別通常以企業的領域術語命名，用來表示使用個案完成後仍須儲存在資料庫中的資料（永存類別），例如第 5 章便當王網路訂購系統案例中的「訂單資料」就是實體類別，其物件之資料須被儲存在資料庫中。但某些實體類別也可以是暫存的資料，例如搜尋結果，當使用個案執行結束後，這些資料也跟著消失（暫存類別）。因此，實體類別可以是暫存類別或永存類別，而大部分是屬於永存類別。

2. 介面類別

介面類別也稱邊界類別，其物件可能包含表單、報表、硬體介面以及其他和系統溝通的介面，也是行為者與系統交談的媒介。介面類別的屬性包含顯示在硬體介面上的元件，例如欄位、超連結、選取方塊與按鈕等；其操作則是將使用者在介面上輸入的資料傳遞給控制類別。當程式執行完畢後，介面類別之物件都將被刪除，因此介面類別屬於暫存類別。

3. 控制類別

控制類別負責協調其他類別的工作，控制類別大部分是傳送許多訊息給其他類別，或是將工作指派給其他類別，同時也會選擇執行的流程，當有錯誤發生時

知道該做什麼，所以控制類別也稱管理類別。控制類別屬於暫存類別，因為當程式執行完畢後，該類別之物件都將被刪除。一般而言，一個使用個案至少須搭配一個控制類別，藉由控制類別來控制使用個案中各項事件的發生順序。

● 屬性與操作

類別的屬性（或操作）可以定義有哪些物件能存取（或使用）該屬性（或操作），這就是類別屬性（或操作）的可視性 (Visibility)，第 3 章所述的封裝即可透過可視性來達成。封裝為將類別的實作細節（包含屬性和操作）隱藏，只允許特定類別存取或使用其資訊，至於是哪些特定類別，便是透過可視性來決定。可視性分為公開的 (Public)、私有的 (Private)、保護的 (Protected) 和套件式 (Package) 等不同的權限等級，符號表示如表 9-1 所示 (UML 表示法)。

1. 公開的：表示該屬性（或操作）可被系統中所有類別存取。
2. 私有的：為最嚴格的存取限制，表示該屬性（或操作）只允許在巢狀類別或同一個類別內被存取，不論是否繼承此類別都無法存取。
3. 保護的：與「私有的」相似，但允許與被存取類別間有繼承關係之類別存取該屬性（或操作）。例如附錄 3D 子類別 MonthlyEmp 可以存取父類別 Employee 內的 employeeName 屬性。
4. 套件式：表示該屬性（或操作）只能被同一個套件內的其他類別存取，不論是否繼承此類別都無法存取。例如附錄 3D 之 com.mod3.employee 套件中的類別 Employee 有兩個常數 PERCENT 和 WAGE 是可以在同一個套件範圍內被存取，因此在同一套件中的 MonthlyEmp 類別可以存取這兩個常數。

Table
表 9-1 可視性表示符號

| 分類 | UML 表示法 | 權限 |
|-----|---------|--------------------------|
| 公開的 | + | 可被系統中所有類別存取 |
| 私有的 | - | 可允許巢狀類別或同一個類別存取 |
| 保護的 | # | 可允許有繼承關係之類別、巢狀類別或同一個類別存取 |
| 套件式 | ~ | 只允許同一個套件內的其他類別存取 |

而類別中的操作除了可視性外，還會描述能夠分辨同類別中特定操作的元素集合，也就是操作簽章 (Signatures)。操作簽章可以幫助我們深入瞭解每個操作內包含的元素並能夠辨認不同的操作。操作簽章定義了三項有關於操作的元素：「操作名稱」、「參數串列」以及「回傳值型態」，此三項元素可以在 UML 圖形中，

依據以下的順序將每個操作予以呈現：

<name>(<parameter-list>):<return-type>

舉便當王系統之「修改訂購數量0」操作為例，該操作包含操作簽章的表達方法如下所示：

+ 修改訂購數量（便當編號: **String**, 數量: **Integer**）: **Boolean**

9.2.2 類別圖－類別間之關係

類別間之關係指的是類別間的連結，在物件導向塑模中，類別間最重要的關係有相依、一般化、關聯與實現化四種關係。這些關係分別用不同型態的線條來表示，例如相依用虛線箭頭；一般化用實線的空心三角形箭頭；關聯用實線；實現化則用虛線的空心三角形箭頭表示（如表 9-2 所示）。物件間並非完全獨立，彼此須知道對方的存在才能完成某一工作，而物件間之關係可分成靜態 (Static) 與動態 (Dynamic) 關係兩種。前者係指一物件知道另一物件之存在，或一物件會用到另一物件之服務，但此種關係並不給予一個物件去改變另一物件資訊的權限；後者係指兩物件間有相互進行溝通。

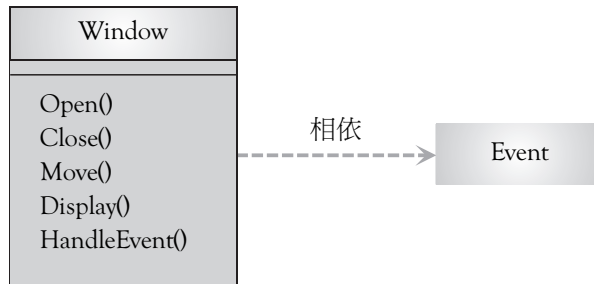
Table 9-2 類別間之關係與表示符號

| 類別間之關係 | 符號 |
|--------|--------|
| 相依關係 | -----> |
| 一般化關係 | —————▶ |
| 關聯關係 | ————— |
| 實現化關係 |▶ |

1. 相依關係

相依 (Dependency) 關係是一種「使用」的關係 (Using Relationship)，表示一個類別會用到其他類別，且被使用之類別的改變可能會影響到使用它的類別，但反之則不必然。相依關係之箭頭是由使用類別指向被使用類別，以虛線箭頭作為表達符號。圖 9-2 表示 Window（視窗）類別使用 Event（事件）類別，兩類別間為相依關係，且 Event 類別的更改會影響 Window 類別的操作，所以相依關係的箭頭方向是由 Window 類別指向 Event 類別，即 Window 類別上的操作行為會依照 Event 類別不同的行為而改變。例如滑鼠在 Window 類別開啟 (Open)、關閉 (Close) 或移動 (Move) 等按鈕點選事件 (Event)，會使此 Window 類別產生不同的操作行為。

Figure 9-2 類別之相依關係範例



2. 一般化關係

一般化 (Generalization) 關係是指一般性類別（稱為超類別或父類別）與僅具有此一般性類別之某些特定性質的類別（稱為次類別或子類別）間的關係。一般化乃是類別間繼承的特性，與之相反的特性為**特殊化 (Specialization)**。一般化意謂著子類別繼承其父類別的特性，尤其是父類別的屬性與操作。一般化關係之箭頭是由子類別指向父類別，表達符號為實線的空心三角形箭頭，如圖 9-3 中之客戶與公司客戶和個人客戶間之關係 (Fowler, 2018)。

Figure 9-3 類別之一般化關係範例

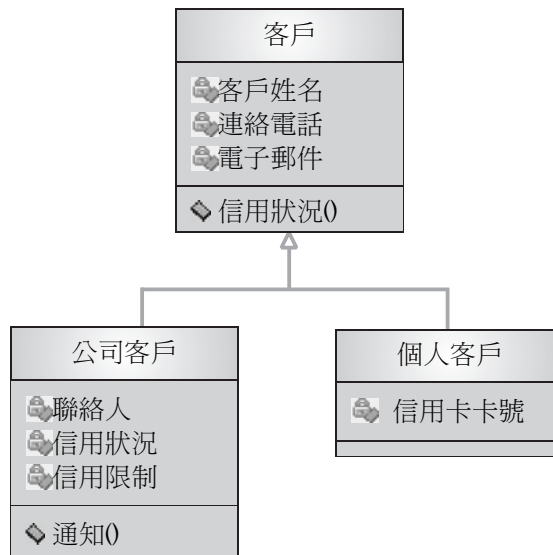


圖 9-3 表示公司客戶和個人客戶兩者有相似但也有相異之處。例如兩者皆有名稱與地址，但個人客戶比公司客戶多出信用卡卡號，而公司客戶比個人客戶多出聯絡人、信用狀況與信用限制。兩者相似處可一般化成一個父類別稱為客戶，公司客戶與個人客戶均繼承其父類別之結構（屬性）與行為（操作），但又分別

擁有各自的特性；而客戶可延伸出公司客戶與個人客戶兩類別，即所謂的特殊化。

3. 關聯關係

關聯 (Association) 關係是一種類別間之靜態結構關係，描述類別與類別間之連結。關聯關係意謂著一類別之物件知道另一類別之物件的存在，或一類別之物件使用到另一類別之物件的服務，但不是擁有此服務。例如訂單中會有產品資訊，因此要完成一份訂單需要用到產品的資訊，符合一類別之物件（訂單）使用到另一類別之物件（產品）的服務（訂單會用到產品的資訊），故訂單與產品間會有關聯關係。

關聯關係如（結構化系統分析與設計之）實體關係圖上的關係，可以有兩個類別參與稱為**二元關聯 (Binary Association)**，也可以有兩個以上的類別參與稱為**多元關聯 (n-ary Association)**。若 A 類別與 B 類別間有關聯關係，則意指可從 A 類別之一個物件航向 B 類別之一個物件；反之亦然。關聯關係之表達符號為實線，一般關聯若沒有箭號即表示雙向關係，若在類別間之關聯加上箭頭便表示具有適航性 (Navigability)，即可依箭頭方向找到另一端之資訊。以圖 9-4 為例，表示可從某一個使用者查到其相對應之密碼，但卻無法從密碼查出使用者是誰。另以圖 9-3 中的訂單與客戶類別為例，若由訂單類別至客戶類別畫出其適航性關聯（加一箭頭於客戶端），則表示可由訂單查到訂購的客戶，但無法由客戶查詢其訂購的所有訂單。

Figure 9-4
圖 9-4 類別之關聯關係適航性範例



類別間的關聯關係不一定要給予名稱，但若覺得必要，則被採用的命名方式有兩種：描述關係的本質或角色。而描述關係的本質可用動詞或動詞片語來表達，描述角色可用名詞或名詞片語來表達。例如客戶及登入資訊兩個類別，若兩者間的關聯之命名為「客戶擁有登入資訊」，則是描述關係的本質；若為「客戶是登入資訊的擁有者」，則是描述關係的角色，如圖 9-5a 所示。

另外，以客戶及訂單之關聯關係說明，從關係的本質來看，客戶會根據自身的需求去填寫訂單的內容，名稱的方向是由客戶指向訂單；從關係的角色來看，是客戶扮演擁有者的角色及訂單扮演客戶需求的角色，也就是說客戶是訂單的擁有者，而訂單代表著客戶的需求，這兩種表達方式如圖 9-5b 所示。