第9章 類別圖與物件圖

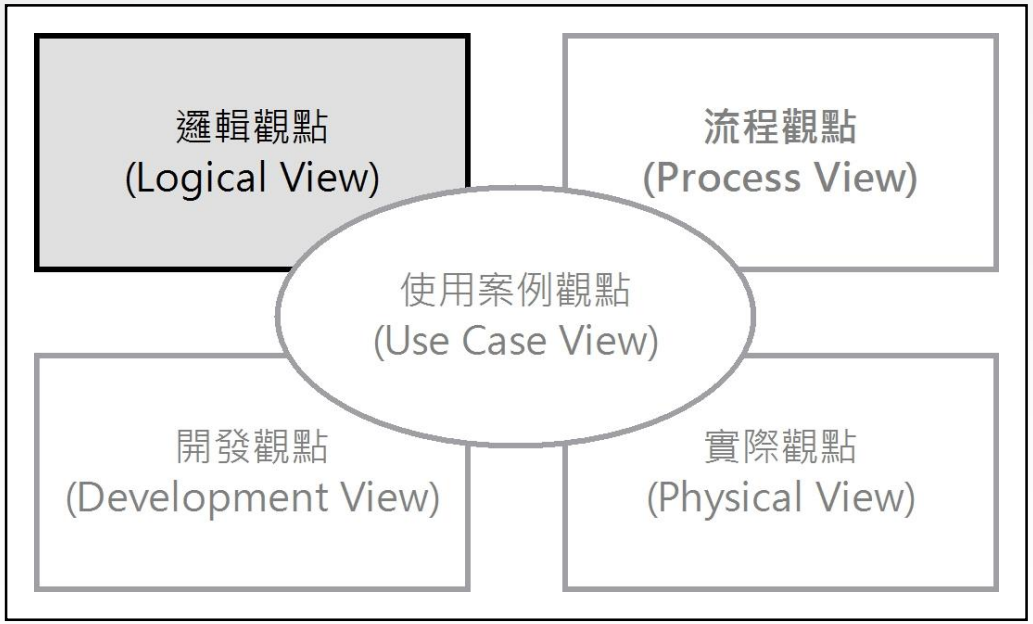
- 9-1 類別圖與物件圖的基礎
- 9-2 類別圖的符號
- 9-3 類別關係
- 9-4 物件圖
- 9-5 繪製類別圖與物件圖
- 9-6 綜合練習

9-1 類別圖與物件圖的基礎-說明

- 類別是物件導向軟體系統的核心，UML類別圖（Class Diagram）可以用來描述軟體系統靜態結構的類別和類別關係（Relationships），它是我們最常使用的UML圖形。
- 物件圖（Object Diagram）就是類別圖的實例（Instances），其基本觀念和類別圖相同，不過，物件圖描述的系統靜態結構是系統某一時間點的快照（Snapshot）。

9-1 類別圖與物件圖的基礎-4+1觀點

- 類別圖與物件圖可以呈現4+1觀點軟體系統模型的邏輯觀點，如下圖所示：



9-1 類別圖與物件圖的基礎-類別圖的目的

- 類別圖的主要目的是建立軟體系統靜態觀點的模型，它是唯一可以直接對應物件導向程式語言的UML圖形，其主要目的如下所示：
  - 建立物件導向分析和設計階段的領域、概念、分析和設計模型。
  - 描述系統的責任（Responsibilities）。
  - 類別圖是套件、元件和部署圖的基礎。
  - 支援UML塑模工具將類別圖轉換輸出成程式碼，或是反向工程將程式碼轉換成UML類別圖。

9-1 類別圖與物件圖的基礎-物件圖的目的

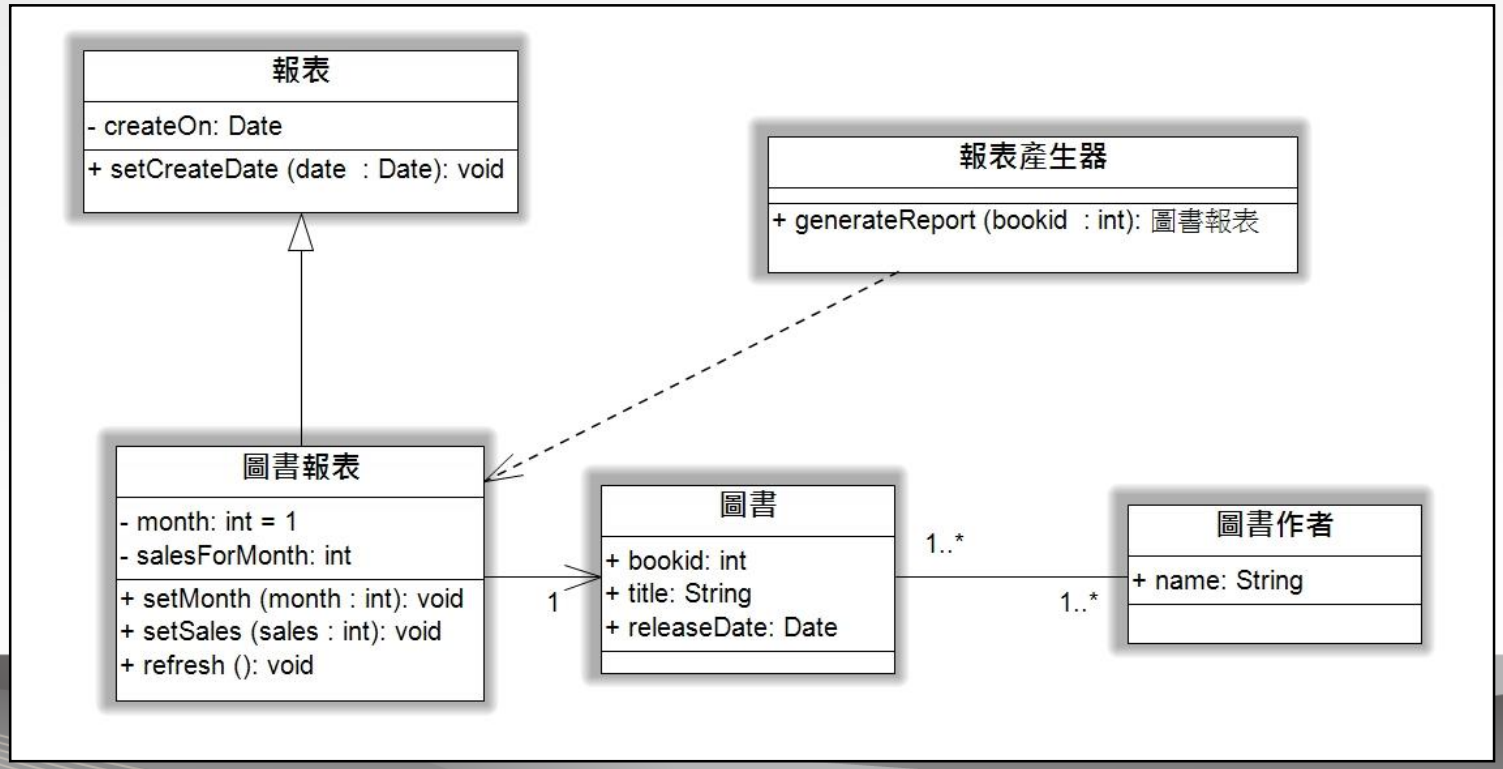
- 物件圖和類別圖的差異在於類別圖是類別和其關係建立的抽象模型，物件圖是在特定時間點建立的實例。換句話說，物件圖比類別圖更接近實際的系統行為，其主要目的如下所示：
  - 支援輸出程式碼和反向工程。
  - 描述系統的物件關係。
  - 描述物件之間互動的靜態觀點。
  - 了解特定觀點的物件行為和其關係。

9-2 類別圖的符號

- 9-2-1 類別符號
- 9-2-2 屬性與能見度
- 9-2-3 操作

9-2 類別圖的符號

- UML類別圖的基本單位是類別符號（Notation），多個類別之間擁有類別關係，我們可以使用多種連接線來標示類別之間不同的類別關係，例如：圖書銷售系統（Book Sales System）的類別圖，如下圖所示：

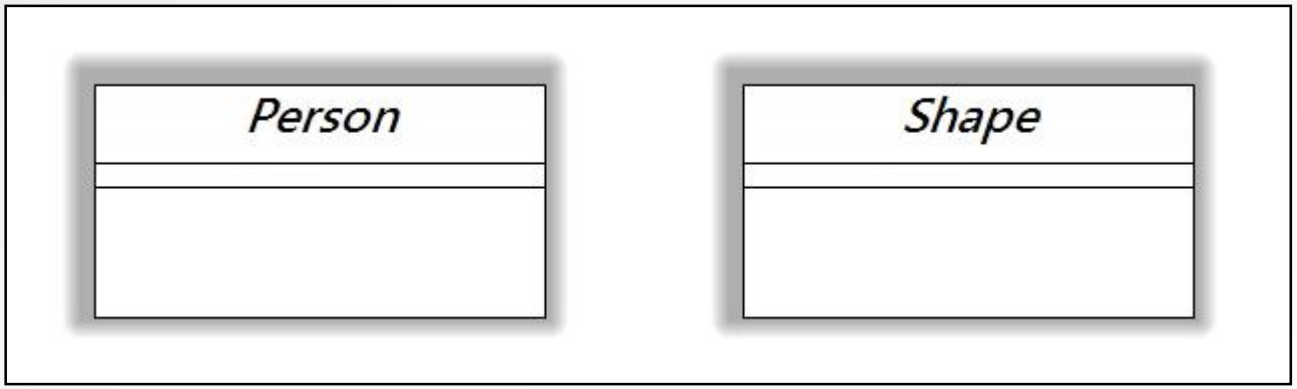


9-2-1 類別符號-類別

- 類別符號是組成類別圖的基本單位，它是使用長方形來表示，在長方形中由上而下分成三個部分：類別名稱、屬性（Attribute）和操作（Operation），如下圖所示：

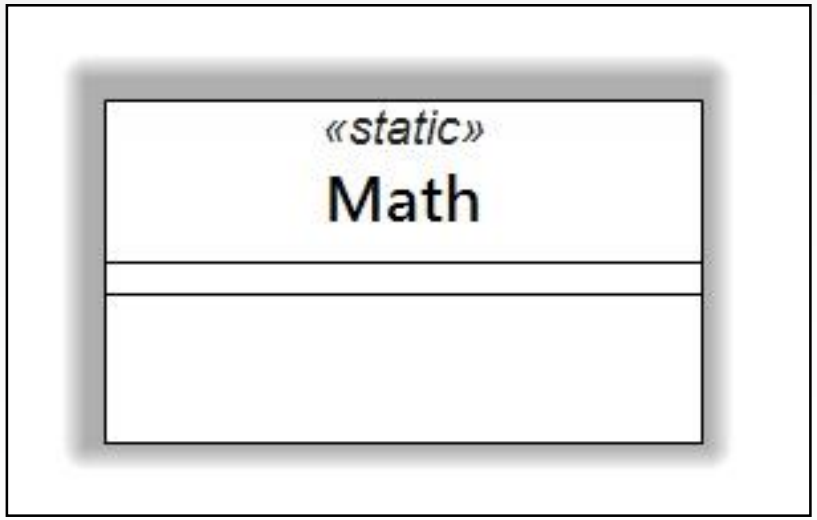
9-2-1 類別符號-抽象類別

- 抽象類別（Abstract Class）是使用斜體字的類別名稱來表示。例如：抽象類別Person，如下圖所示：



9-2-1 類別符號-靜態類別

- 靜態類別（Static Class）和其他非靜態類別並沒有什麼不同，唯一差異是靜態類別不能建立物件，而且在類別名稱上方加上<<static>>模版來表示，如下圖所示：



9-2-2 屬性與能見度-屬性語法

- 在類別符號中間部分是屬性（Attributes）清單，這是類別的性質、特徵或狀態，每一個屬性自成一列來表示。
- 類別屬性的基本語法，如下所示：

  能見度 屬性名稱 : 資料型態[= 初值]

- 上述語法的開頭是能見度符號+、-、#和~，符號之後和「:」符號之前是屬性名稱，「:」符號之後是資料型態，如果屬性有初值，可以在「=」等號後指定初值。例如：類別的month和salesForMonth屬性，如下所示：

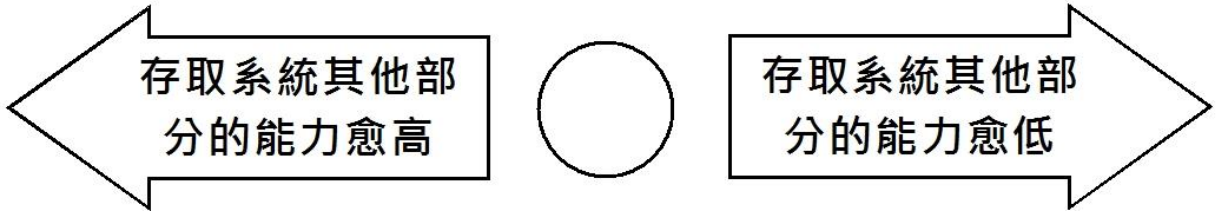
  - month : int = 1

  - salesForMonth : int

9-2-2 屬性與能見度-能見度(種類)

- 在操作和屬性前可以加上存取修飾子，稱為能見度（Visibility），或稱為可見度和可見性。能見度是物件導向的封裝機制，可以指定操作或屬性的存取等級，避免屬性被任意修改，或操作被任意呼叫。UML的能見度分為四種，如下圖所示：

名稱	Public	Protected	Package	Private
符號	+	#	~	-



9-2-2 屬性與能見度-能見度(說明1)

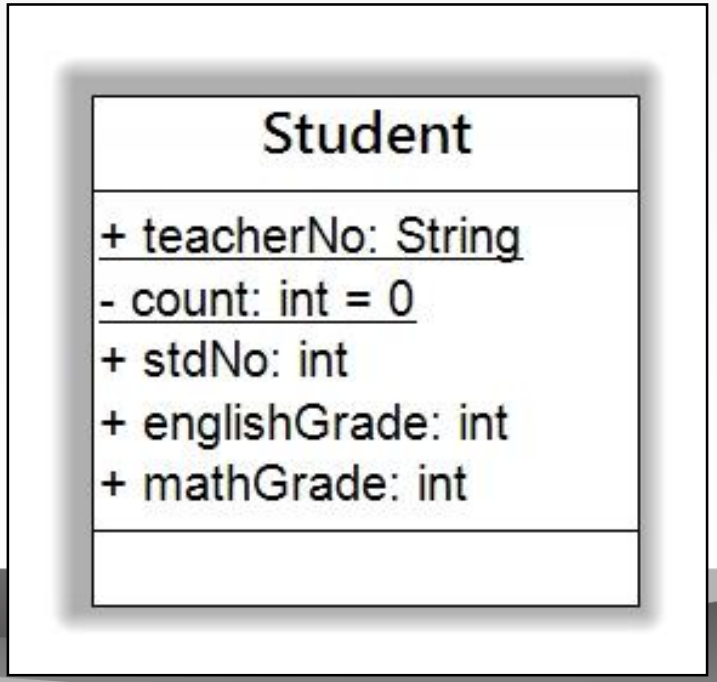
- Public能見度：允許其他任何類別存取的屬性或操作，事實上，在類別中宣告Public能見度的屬性和方法就是類別對外的使用介面，允許其他類別存取和使用。它是使用「+」符號表示Public能見度。
- Protected能見度：允許類別本身和其繼承的子類別存取屬性和使用操作。它是使用「#」符號表示Protected能見度。

9-2-2 屬性與能見度-能見度(說明2)

- Package能見度：只允許同一個套件的類別可以存取屬性和使用操作。它是使用「~」符號表示Package能見度。
- Private能見度：表示它是屬於類別本身，除類別本身之外，不允許任何其他類別存取或使用。它是使用「-」符號表示Private能見度。

9-2-2 屬性與能見度-靜態屬性

- 靜態屬性（Static Attributes）表示此屬性是屬於類別，不論類別建立多少個物件，存取靜態屬性都是存取同一個屬性，在UML類別圖是使用底線標示靜態屬性，例如：Student類別的teacherNo和count屬性，如下圖所示：



9-2-3 操作-語法1

- 在類別符號最下方是操作（Operations）清單，可以描述類別行為或功能，所謂實作類別程式碼，就是將操作轉換成程式語言的程序與函數，或稱為「方法」（Methods）。
- 類別操作的基本語法，如下所示：

  能見度 操作名稱(參數列) : 傳回資料型態
- 上述操作語法的開頭是能見度符號+、-、#和~，在符號之後和「:」符號之前是操作名稱和括號中的參數列，之後是傳回值的資料型態，沒有傳回值是void。

9-2-3 操作-語法2

- 操作參數列的基本語法如下所示：

  參數名稱：資料型態[,參數名稱：資料型態]

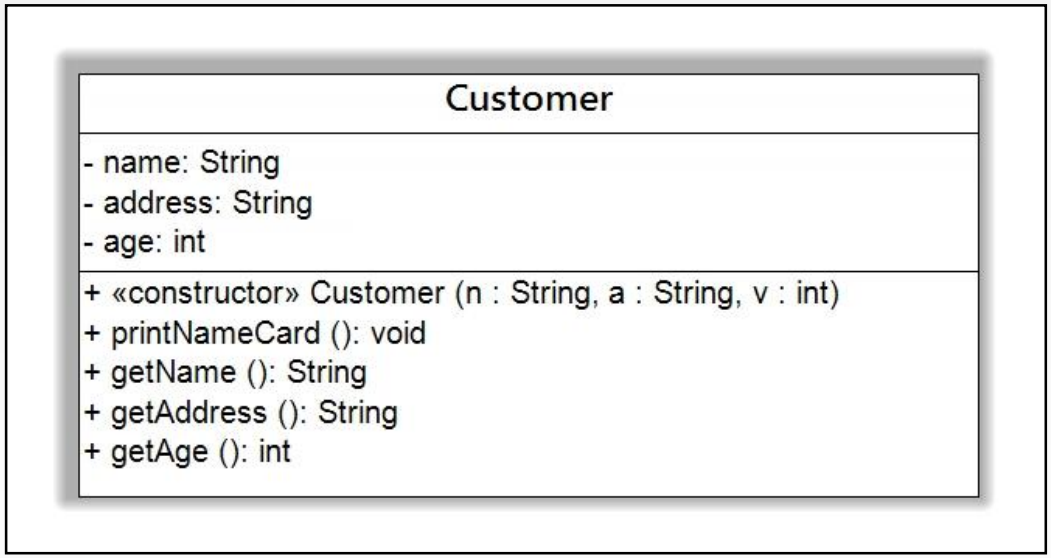
- 上述參數列的每一個參數是使用「,」逗號分隔，之前是參數名稱，「:」符號之後是資料型態，例如：setSales()操作有一個參數；【增加項目()】操作有2個參數，如下所示：

  + setSales(sales : int) : void

  + 增加項目(新項目 : BlogEntry, 作者 : String) : boolean

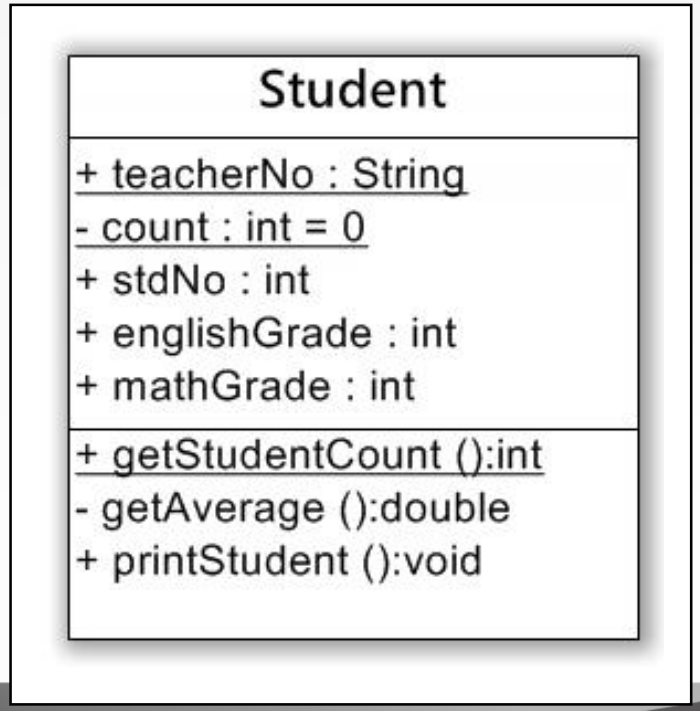
9-2-3 操作-建構子操作

- 建構子操作（Constructor Operations）是類別建構子，可以用來建立物件，在UML類別圖並不會特別區分建構子和一般操作，我們可以在操作之前使用<<constructor>>模版標示為建構子。例如：Customer類別擁有同名的建構子操作，如右圖所示：



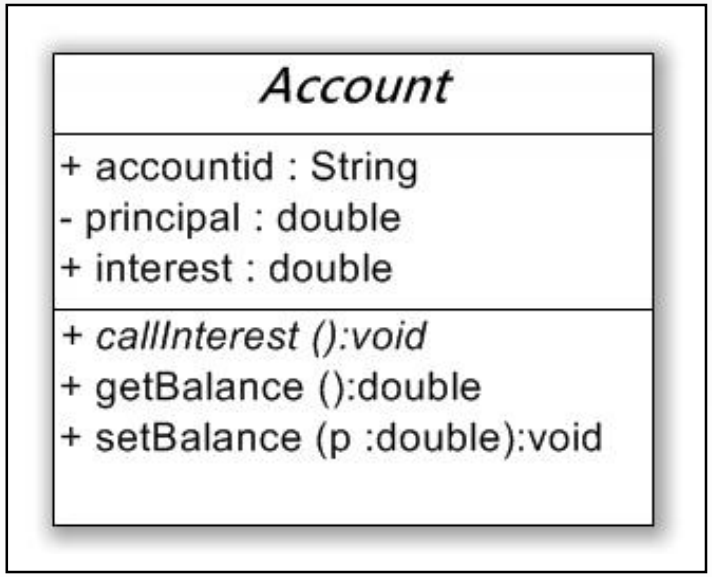
9-2-3 操作-靜態操作

- 靜態操作（Static Operations）如同靜態屬性是屬於類別，不論類別建立多少個物件，使用的靜態操作都是同一個操作，在UML類別圖是使用底線標示靜態操作，如下圖所示：



9-2-3 操作-抽象操作

- 在第9-2-1節的抽象類別可以擁有抽象操作（Abstract Operations），這是使用斜體字標示的操作。例如：Account抽象類別擁有名為callInterest()的抽象操作，如下圖所示：

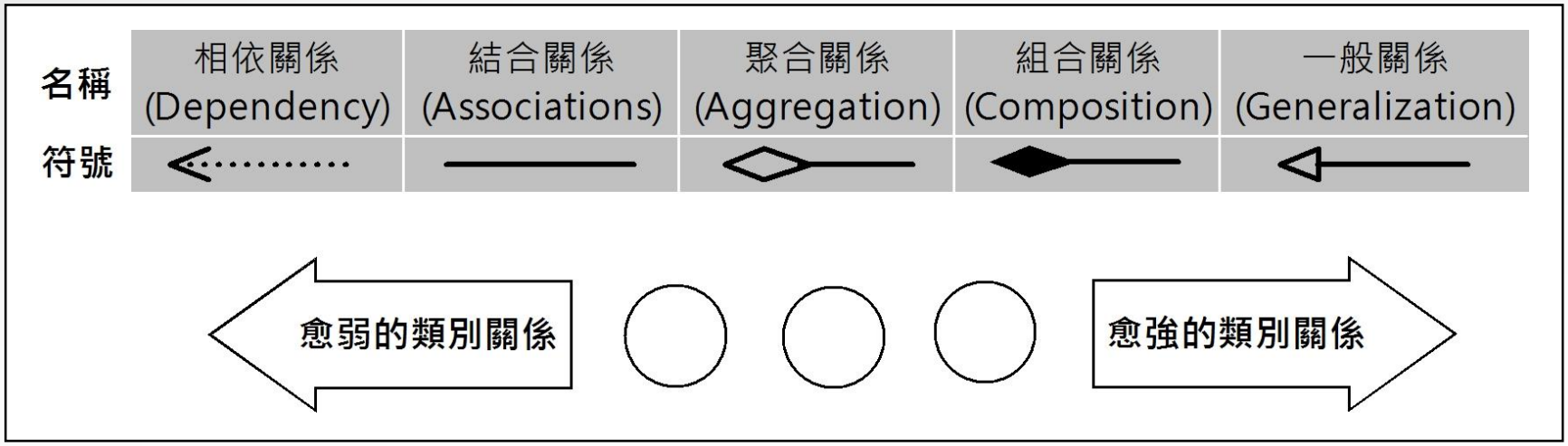


9-3 類別關係

- 9-3-1 類別關係的基礎
- 9-3-2 結合關係
- 9-3-3 聚合關係
- 9-3-4 組合關係
- 9-3-5 角色名稱與多重性
- 9-3-6 結合類別
- 9-3-7 反身關係
- 9-3-8 一般關係
- 9-3-9 相依關係
- 9-3-10 實現關係

9-3-1 類別關係的基礎

- 類別關係是指類別之間擁有的合作關係，UML類別擁有很多種不同強度的類別關係，在此的強度是指類別彼此之間依賴的程度，稱為耦合度（Coupling），如下圖所示：



9-3-2 結合關係-結合關係

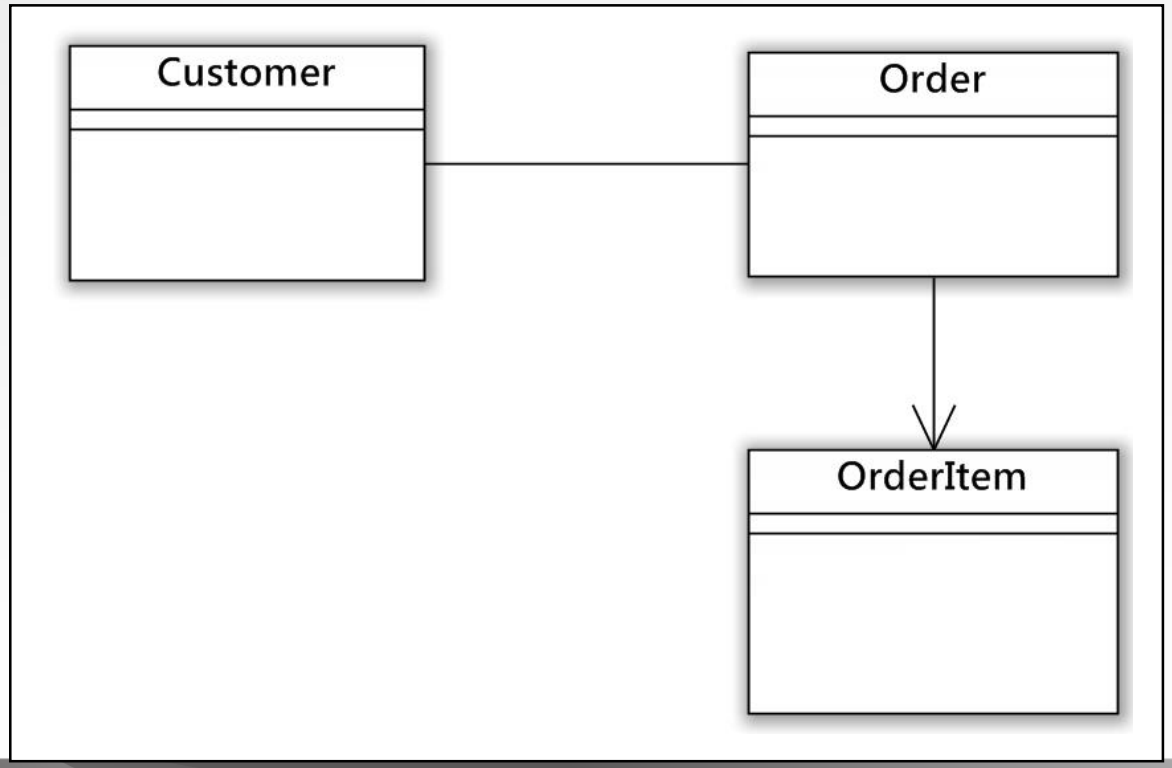
- 結合關係（Associations）是指兩個類別相互知道另一個類別存在的關係，或稱為雙向結合關係，我們可以將兩個類別之間的關係，視為扮演不同的角色。例如：Customer客戶下Order訂單，所以客戶知道下了哪一張訂單；Order訂單需要知道是屬於哪一位客戶所下的訂單，如下圖所示：

9-3-2 結合關係-可導覽的結合關係(說明)

- 可導覽的結合關係（Navigable Associations）是一種擁有方向性的結合關係，稱為互通性（Navigability），也就是在連接線上標示箭頭來表示訊息傳遞的方向。
- 可導覽結合關係是一種單向結合關係，在兩個類別中，只有其中一個類別擁有指標指向另一個類別，反過來，就沒有建立互通訊息的管道。

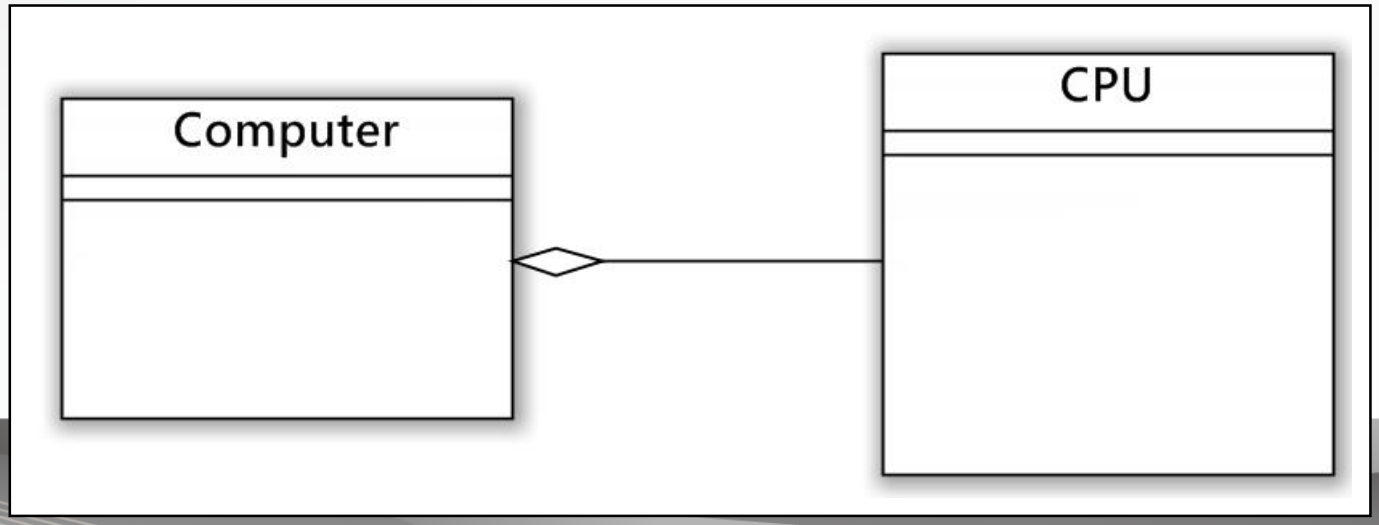
9-3-2 結合關係-可導覽的結合關係(範例)

- 只允許從Order訂單類別查詢OrderItem訂單項目資料，就是可導覽的單向結合關係，如下圖所示：



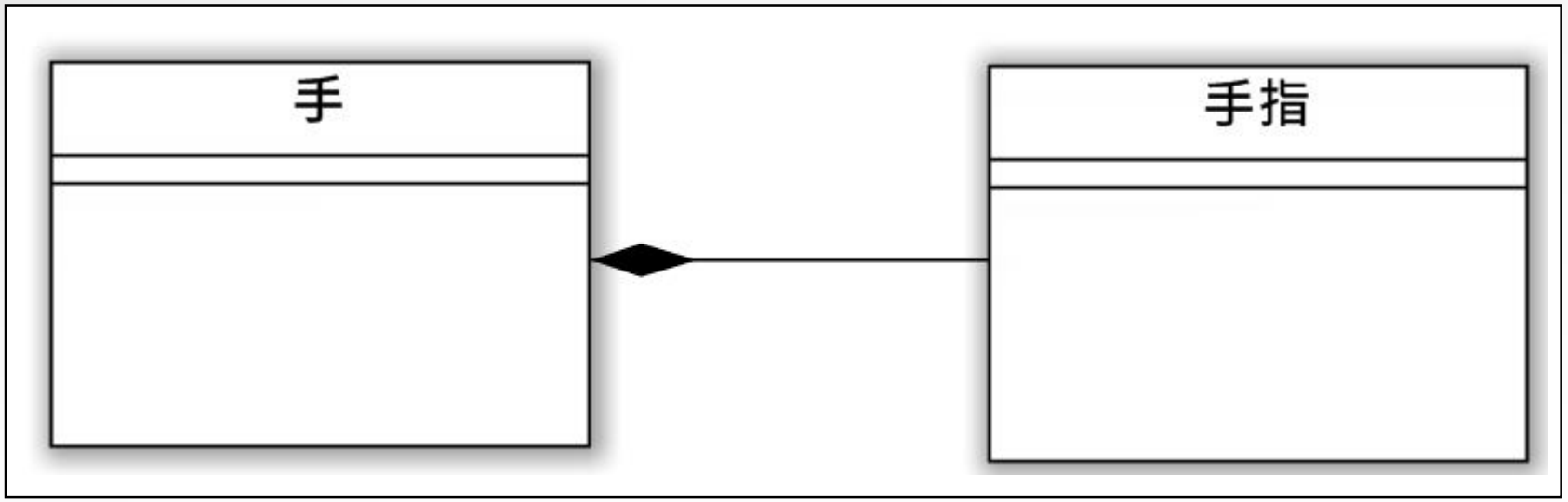
9-3-3 聚合關係

- 聚合關係（Aggregation）是一種關係較強的結合關係，屬於成品和零件（Whole-Part）的類別關係。這是使用空菱形的實線從零件指向成品，屬於通用零件。例如：Computer電腦類別擁有CPU類別的中央處理器，CPU是電腦零件，不同電腦可以使用同一種零件，如下圖所示：



9-3-4 組合關係-說明

- 在聚合關係中最強的類別關係稱為組合關係（Composition），這是一種專屬零件。使用實心菱形的實線從零件指向成品，如下圖所示：

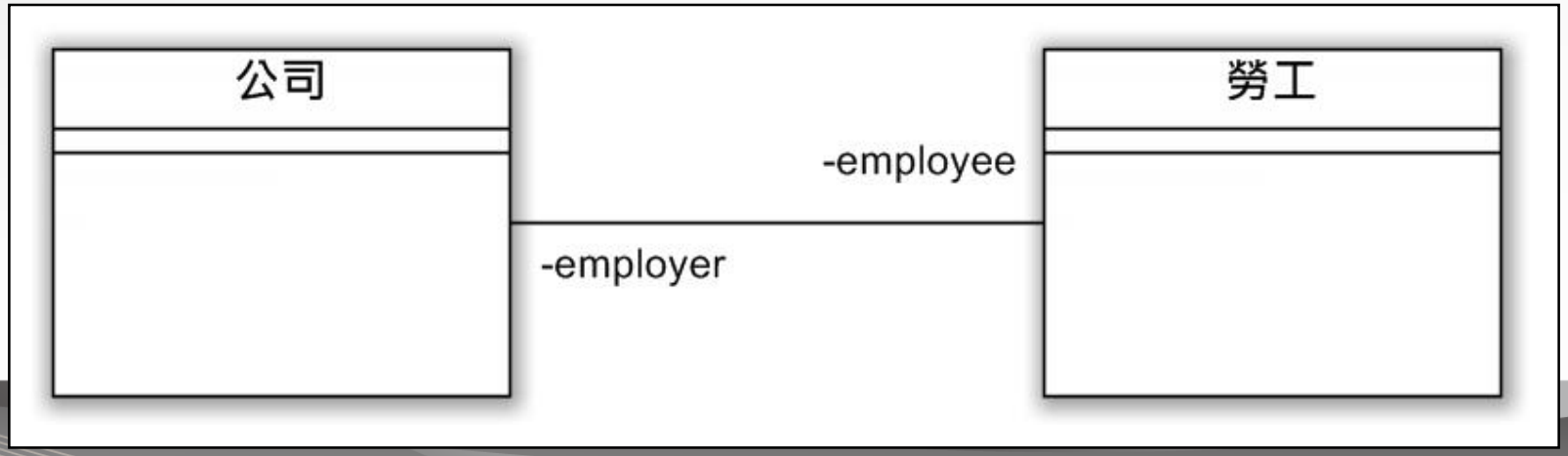


9-3-4 組合關係-差異

- 組合關係和聚合關係的主要差異，如下所示：
  - 組合關係的零件是只能使用在成品的專屬零件；聚合關係的零件是可以使用在其他成品的通用零件。
  - 成品如果不存在，組合關係的零件也不會存在，換句話說，組合關係的零件並不能單獨存在。但是，聚合關係的零件因為是通用零件，所以可以單獨存在。

9-3-5 角色名稱與多重性-角色名稱

- 在類別之間如果擁有結合關係、聚合或組合關係，在類別圖連接線兩端的上方，可以額外標示角色名稱與多重性。
- 角色名稱（Role Name）表示類別在類別關係中扮演的角色。例如：公司（Company）類別是勞工（Worker）的雇主（Employer），如下圖所示：



9-3-5 角色名稱與多重性-多重性(說明)

- 多重性（Multiplicity）表示類別實例參與類別關係的個數是一對一、一對多或多對多，類似資料庫實體關聯圖的一對一、一對多和多對多關聯性。

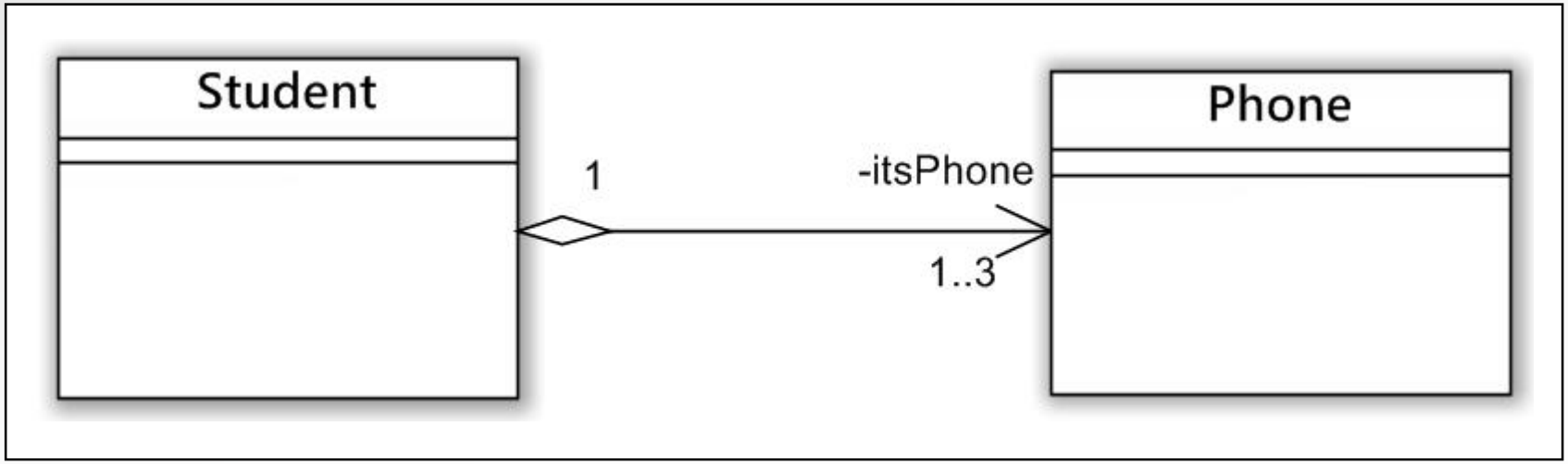
9-3-5 角色名稱與多重性-多重性(數字範圍)

- 在類別關係連接線的兩端，我們可以加上數字範圍的多重性來標示類別參與關係的物件數，其說明如下表所示：

表示符號	說明
1	表示只有一個，例如：公司只有一位董事長；每一個科系只有一位系主任
3	表示有3個，例如：公司有3位副董事長
或0..	表示從0到多個，例如：學生可以選擇0到多個輔系
0..1	表示0或1個，例如：員工有0或1位配偶
1..*	表示至少1個，從1到多個，例如：教師指導1至多位學生的畢業報告
10..*	表示至少10個，從10到多個，例如：學生必須選10至多門必修課
3..5	從「m..n」前的數字m到之後n的範圍，以此例是3到5個，例如：公司員工每年有3~5天的給薪假

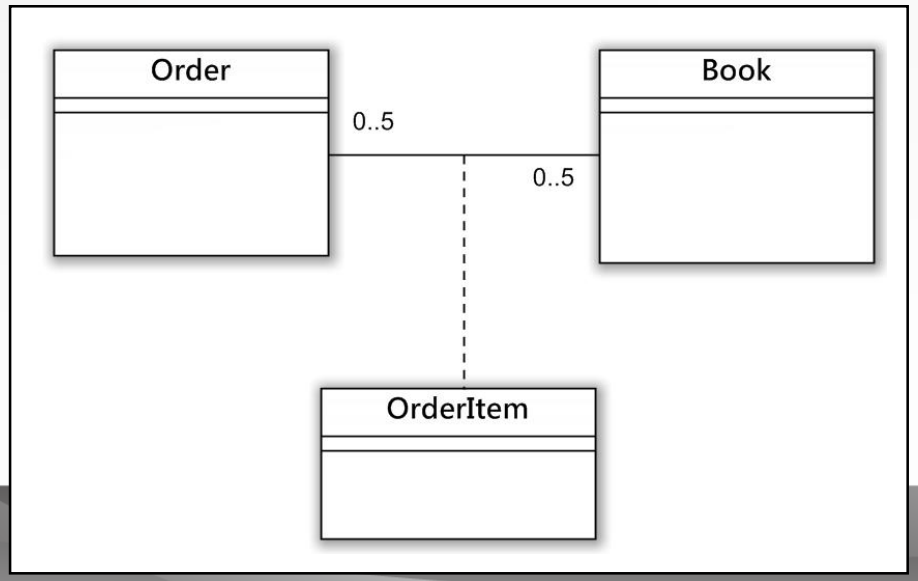
9-3-5 角色名稱與多重性-多重性(範例)

- 例如：學生（Student）擁有住家電話、宿舍電話和手機等1至3個電話（Phone）物件，如下圖所示：



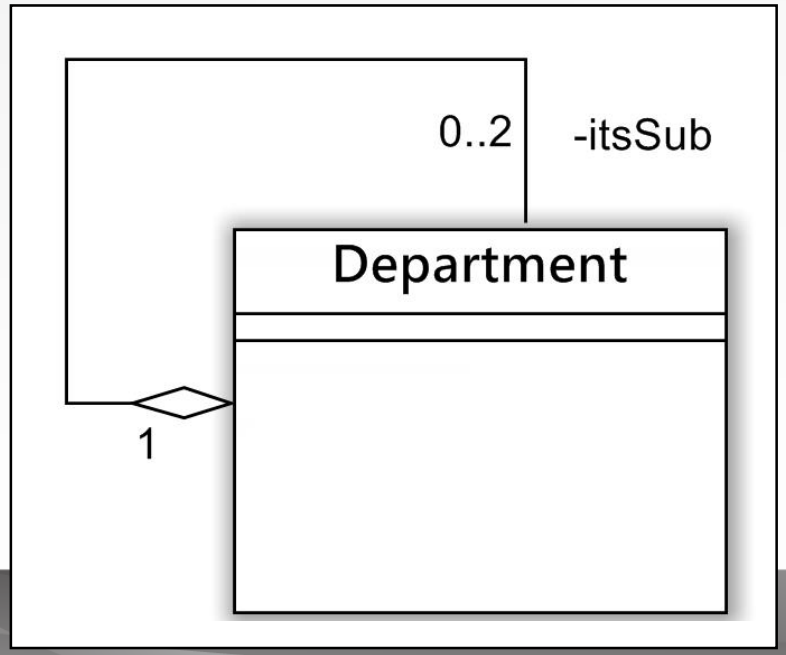
9-3-6 結合類別

- 「結合類別」（Association Class）是一種使用在類別結合關係的中間類別，通常是使用在多對一（Many-to-one）或多對多（Many-to-Many）的結合關係。
- 例如：一個多對多的結合關係，Order訂單（每次最多處理5筆訂單）擁有很多Book的購買圖書（每筆訂單最多5本書），反過來，很多圖書屬於不同的訂單，如下圖所示：



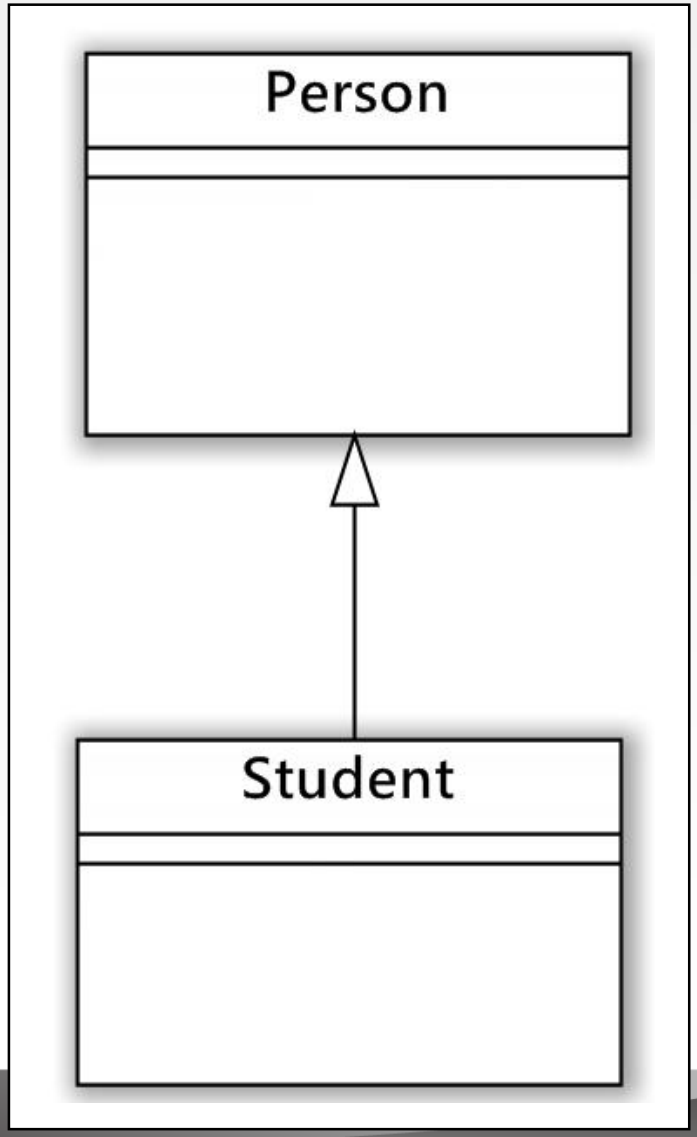
9-3-7 反身關係

- 反身關係（Reflexive Associations）可以使用在結合、組合或聚合關係，它是指類別擁有參考到自己的指標，以聚合關係來說，類別本身是成品；也是零件。
- 例如：學校Department科系類別可以分成很多子科系，每一個子科系物件也是一種Department類別，如下圖所示：



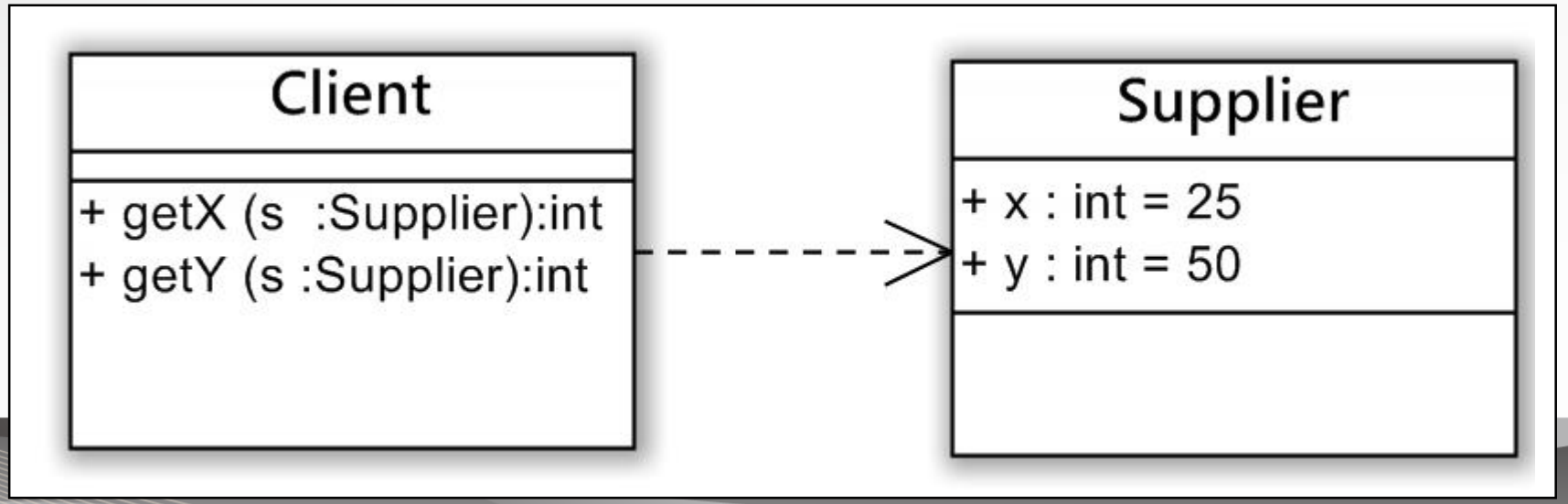
9-3-8 一般關係

- 一般關係（Generalization）就是繼承關係，使用空箭頭的實線從子類別指向父類別，如右圖所示：



9-3-9 相依關係-基礎

- 類別的相依關係（Dependency）是使用虛線箭頭表示一種最弱的結合關係，它是指類別在語意上需要依賴其他類別。
- 當類別之間有相依關係時，更改其中一個類別，有可能會強迫需要更改另一個類別。例如：Client類別的操作參數或傳回值是Supplier類別的物件，如下圖所示：



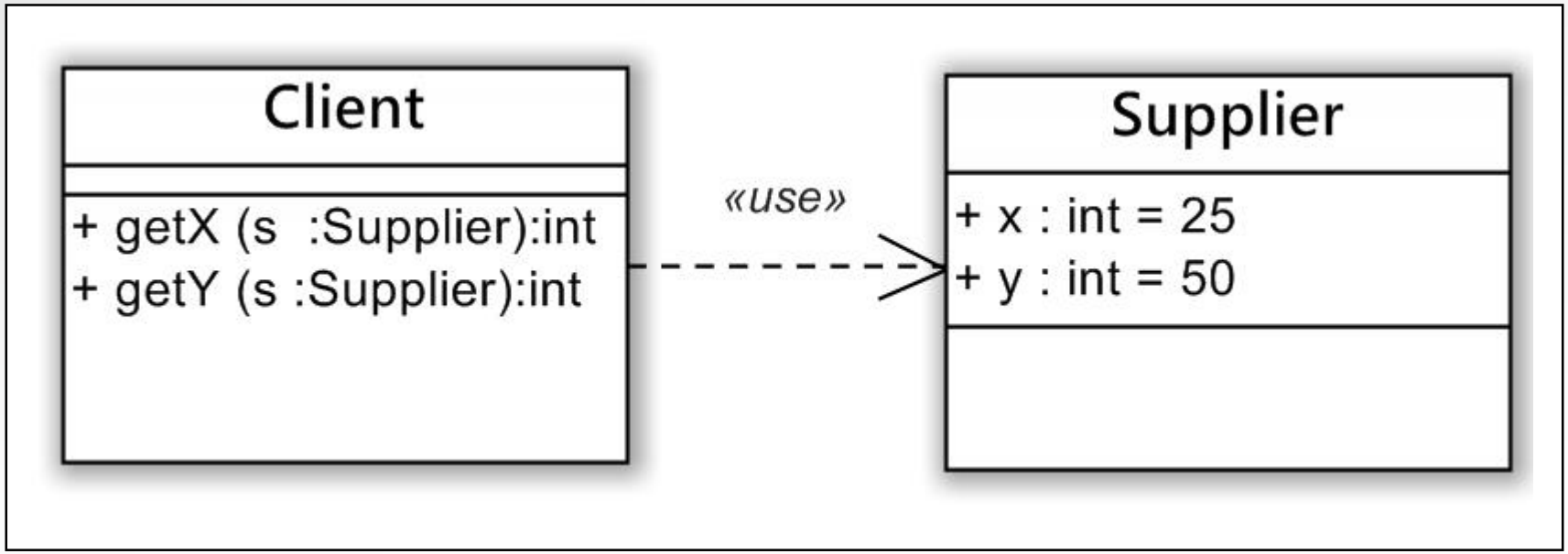
9-3-9 相依關係-
UML 2.x版預先定義的相依關係(說明)

- UML一些常用預先定義的相依關係種類和說明，如下表所示：

模版	說明
«call»	在Client類別的操作呼叫Supplier類別的操作
«create»	在Client類別的建構子建立Supplier類別的實例
«derive»	Client類別的屬性值或其他值是由Supplier類別計算而得
«instantiate»	在Client類別的操作建立Supplier類別的實例
«send»	在Client類別的操作送出一個訊號（Signal）給Supplier類別
«use»	Client類別的成員有使用到Supplier類別

9-3-9 相依關係-
UML 2.x版預先定義的相依關係(範例)

- 相依關係可以在連接線上加上模版來說明是哪一種相依關係，例如：明確標示之前的相依關係為<<use>>，如下圖所示：

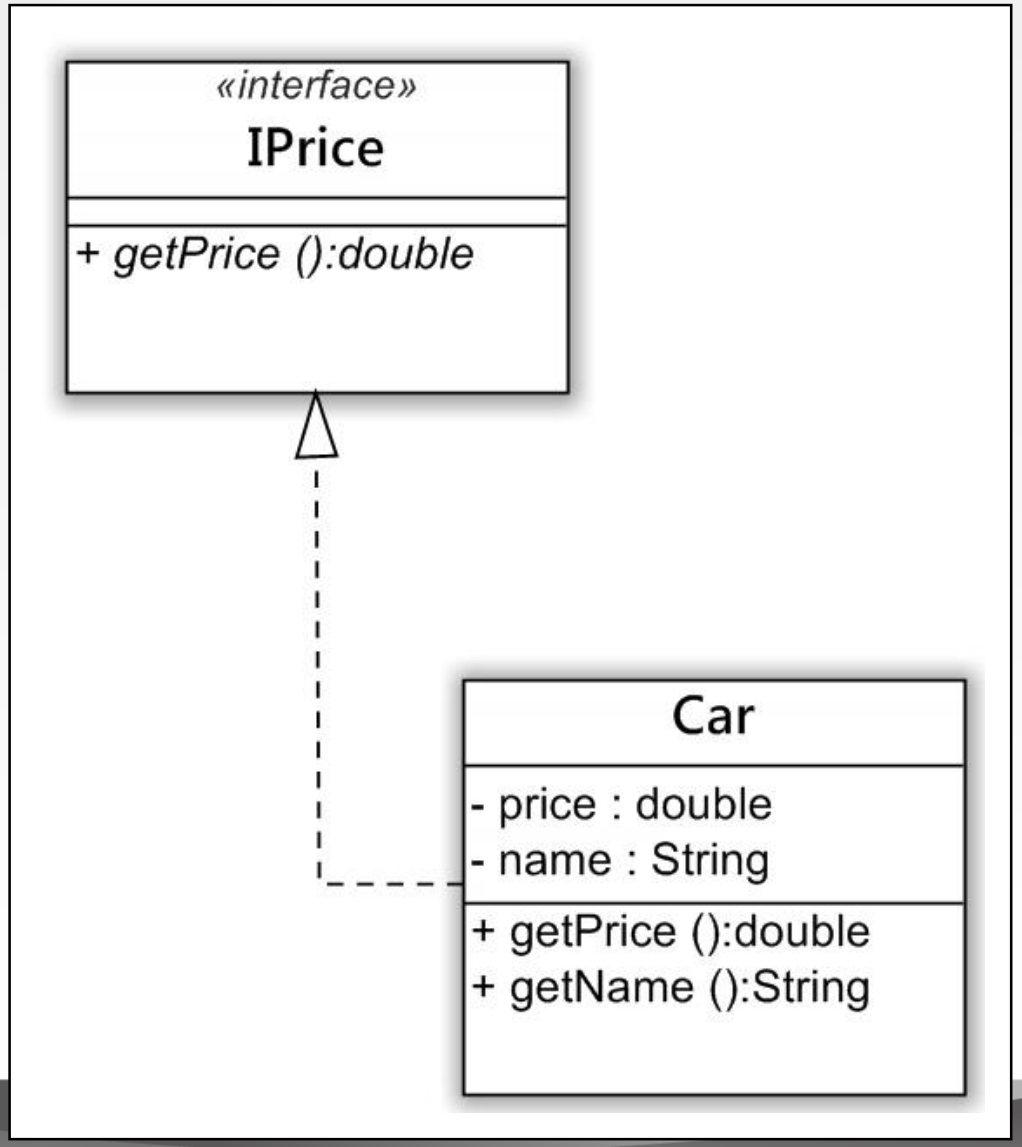


9-3-10 實現關係-說明

- 實現關係（Realization）是類別和介面之間的關係，介面（Interface）是一種特殊的類別型態。UML類別圖是使用虛線和空心三角來表示實現關係，因為實現關係類似一般關係，所以符號也相似，只是連接線由實線改為虛線。
- 介面（Interface）的類別型態可以用來定義行為，然後透過介面替其他類別提供共同行為的定義，就算類別之間沒有任何關係（有關係也可以），一樣可以擁有共同行為的介面。換句話說，介面是用來定義不同類別之間的一致行為。

9-3-10 實現關係-介面

- UML類別圖是在類別名稱上方使用<<interface>>指明為介面，其語法類似模版（Stereotype），但不是模版，稱為類別化（Classifier），如右圖所示：



9-4 物件圖

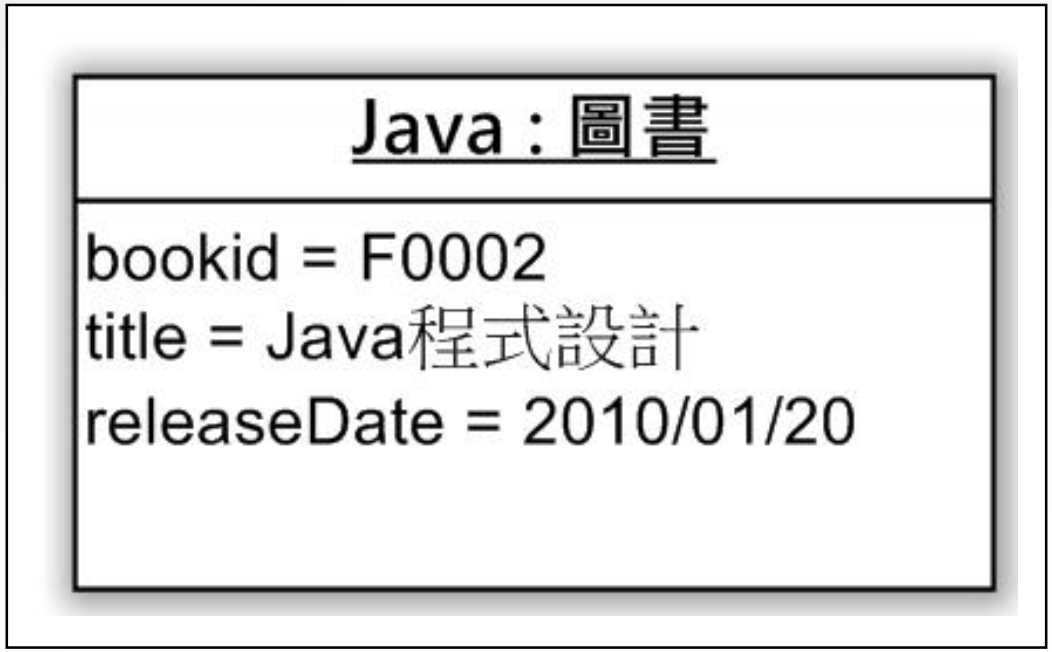
- 9-4-1 物件圖的符號
- 9-4-2 物件關係
- 9-4-3 驗證類別圖的多重性

9-4 物件圖

- 物件圖（Object Diagram）類似類別圖，可以描述特定時間點物件集合和之間的關係。其差別在於物件圖是顯示實例（Instances），而不是類別的藍圖。
- 物件圖呈現的是物件導向軟體系統執行時期的物件狀態，可以實際將我們設計的類別轉換成物件，描述特定集合的物件、屬性值與之間的關係，和這些物件如何通力合作來完成特定的工作。

9-4-1 物件圖的符號-說明

- 物件圖符號對比類別圖簡單一些，在長方形符號分為上下兩部分：物件名稱與屬性值清單，如下圖所示：



9-4-1 物件圖的符號-物件名稱

- 在物件圖上方的底線名稱是物件完整名稱，其基本語法如下所示：

  物件名稱：類別名稱

- 上述語法在「:」符號之前是物件名稱，之後是藍圖的類別名稱。例如：圖書類別的物件圖名稱，如下所示：

  Java：圖書

  C#：圖書

9-4-1 物件圖的符號-屬性值清單

- 在物件圖下方是屬性值清單，這就是源自類別藍圖的屬性清單，而且只有屬性值，其基本語法如下所示：

  屬性 = 值

- 上述語法的每一個屬性值是一個指定敘述，前為屬性名稱，等號後是目前的屬性值。例如：【Java:圖書】物件圖的屬性清單，如下所示：

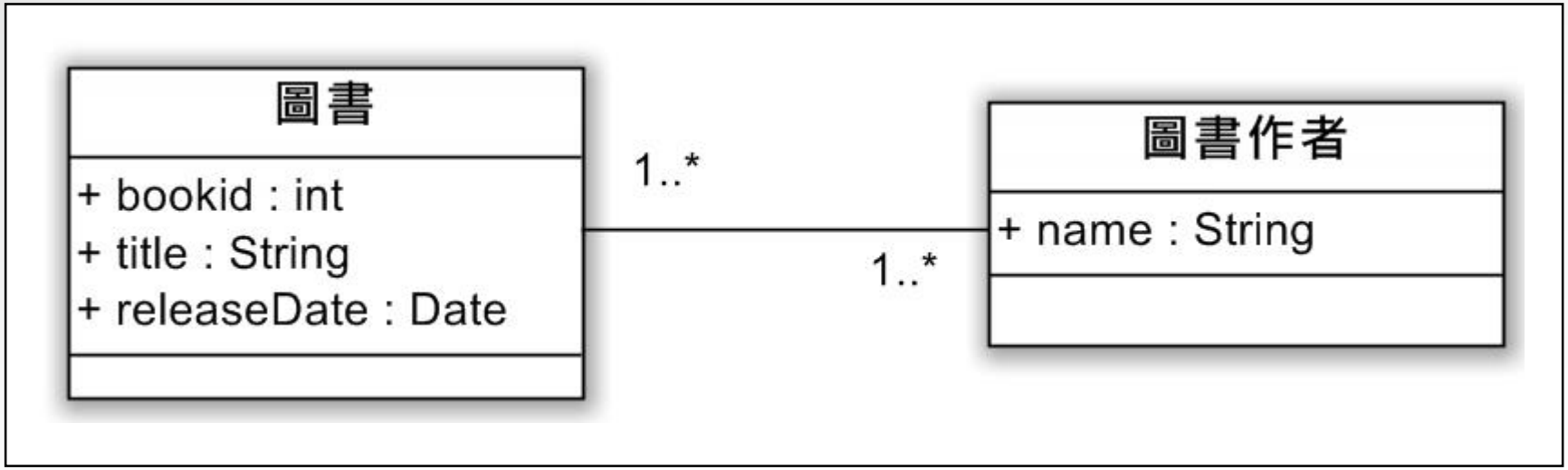
  bookid = F0002

9-4-1 物件圖的符號-匿名物件

- 匿名物件（Anonymous Objects）的物件圖只有類別名稱，沒有指明物件名稱，通常是因為物件名稱並不重要，如下圖所示：

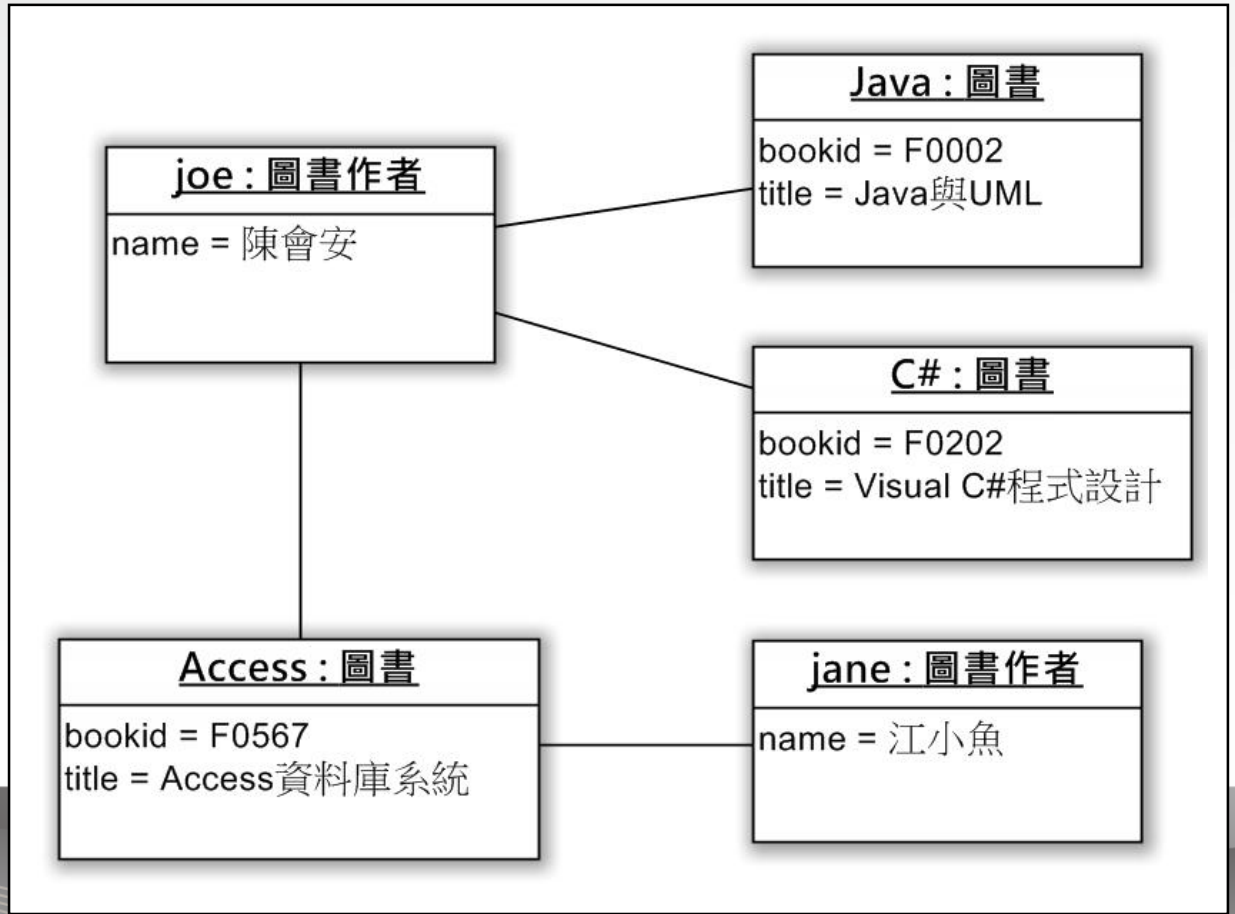
9-4-2 物件關係-類別圖

- 在物件圖一樣可以顯示物件之間的結合關係，我們可以使用連接線標示物件之間的關係，例如：圖書與圖書作者的類別圖，如下圖所示：



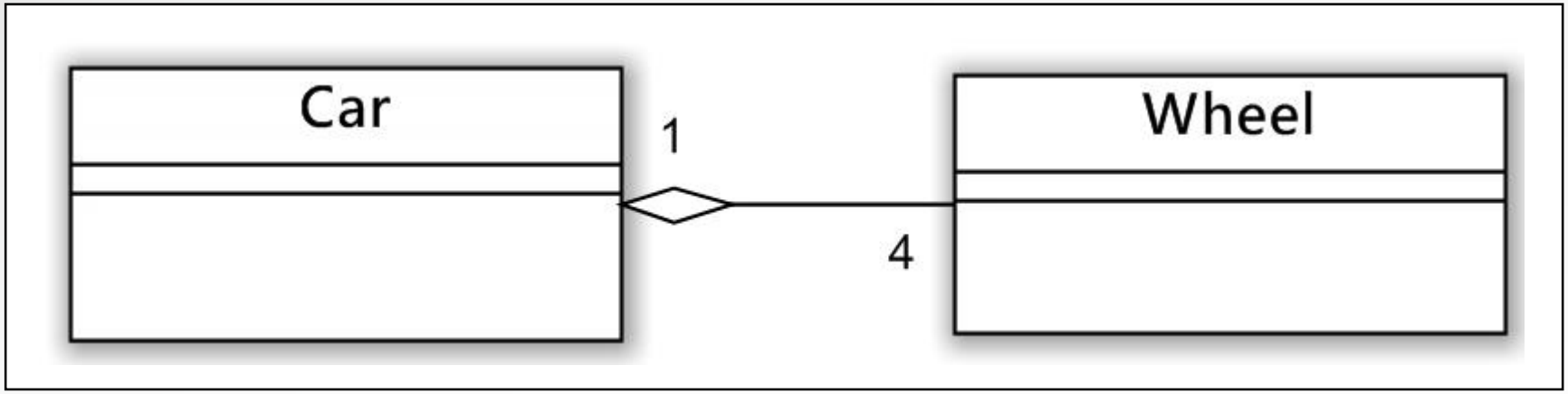
9-4-2 物件關係-物件圖

- 從上述類別圖，可以建立一位作者有多本著作；一本圖書有多位作者的物件圖，如下圖所示：



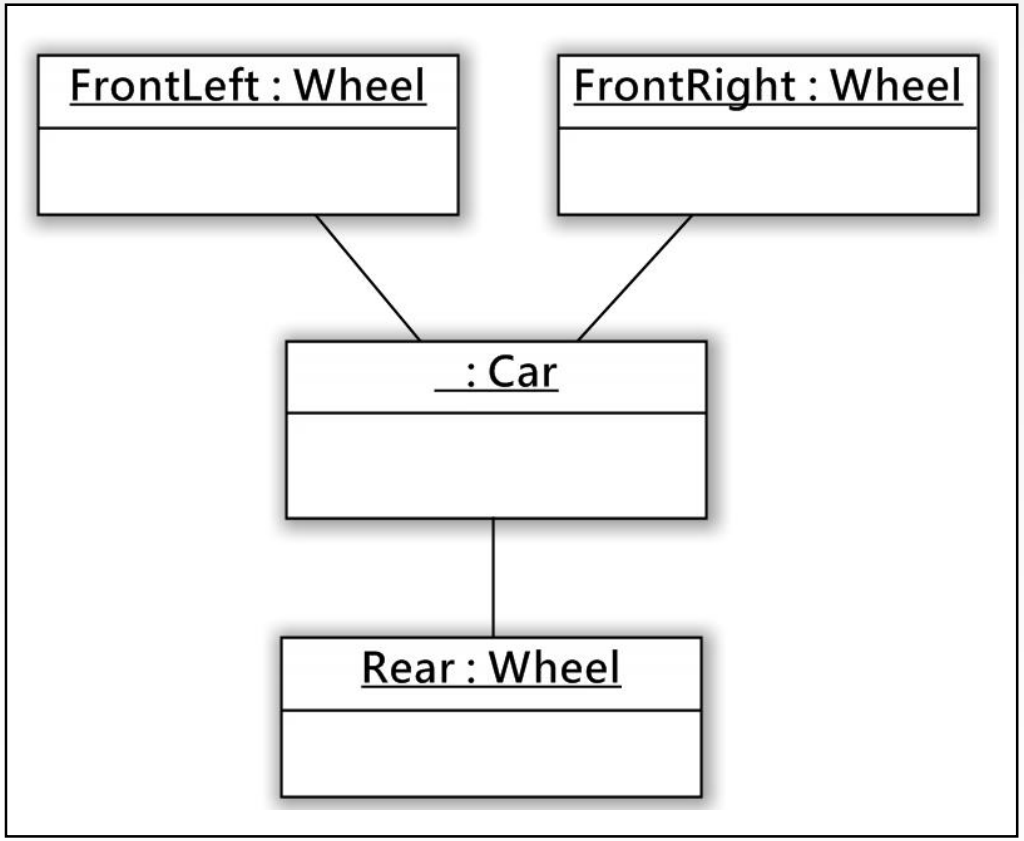
9-4-3 驗證類別圖的多重性-類別圖

- 物件圖常常用來作為類別圖的測試案例，幫助我們驗證類別圖多重性設計是否正確，例如：一輛車有4個輪子，UML類別圖如下圖所示：



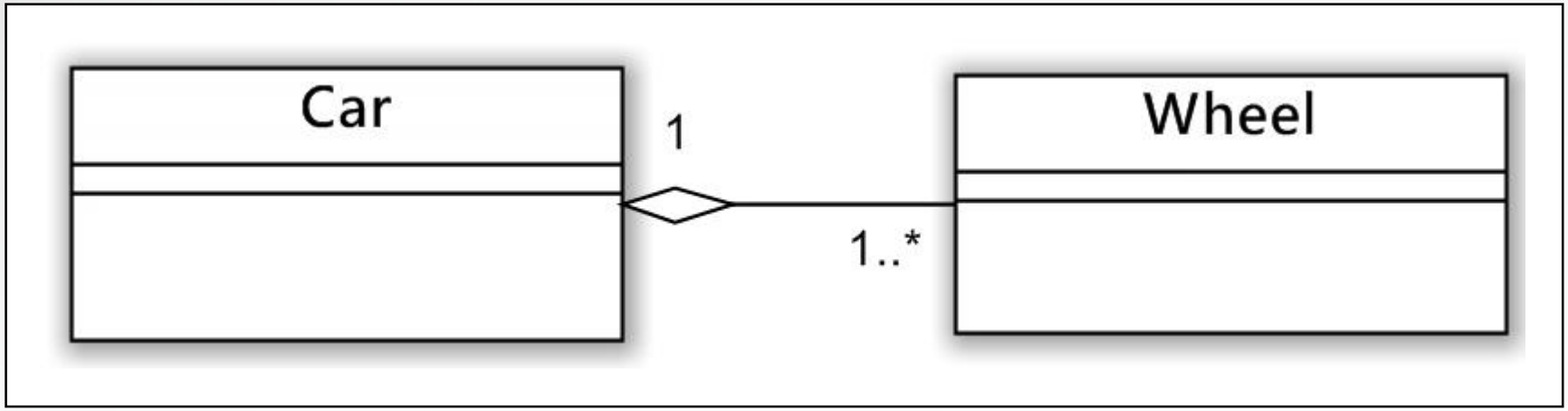
9-4-3 驗證類別圖的多重性-物件圖

- 聚合關係的成品與零件表示一輛Car車有4個Wheel輪子。但是，當我們將它繪成物件圖後，發現並無法建立三個輪子的車輛，如右圖所示：



9-4-3 驗證類別圖的多重性-更正的類別圖

- 車輛只有3個輪子，之前的類別圖是將輪子固定成4個，有問題，所以，應該將多重性改為1..*，如下圖所示：

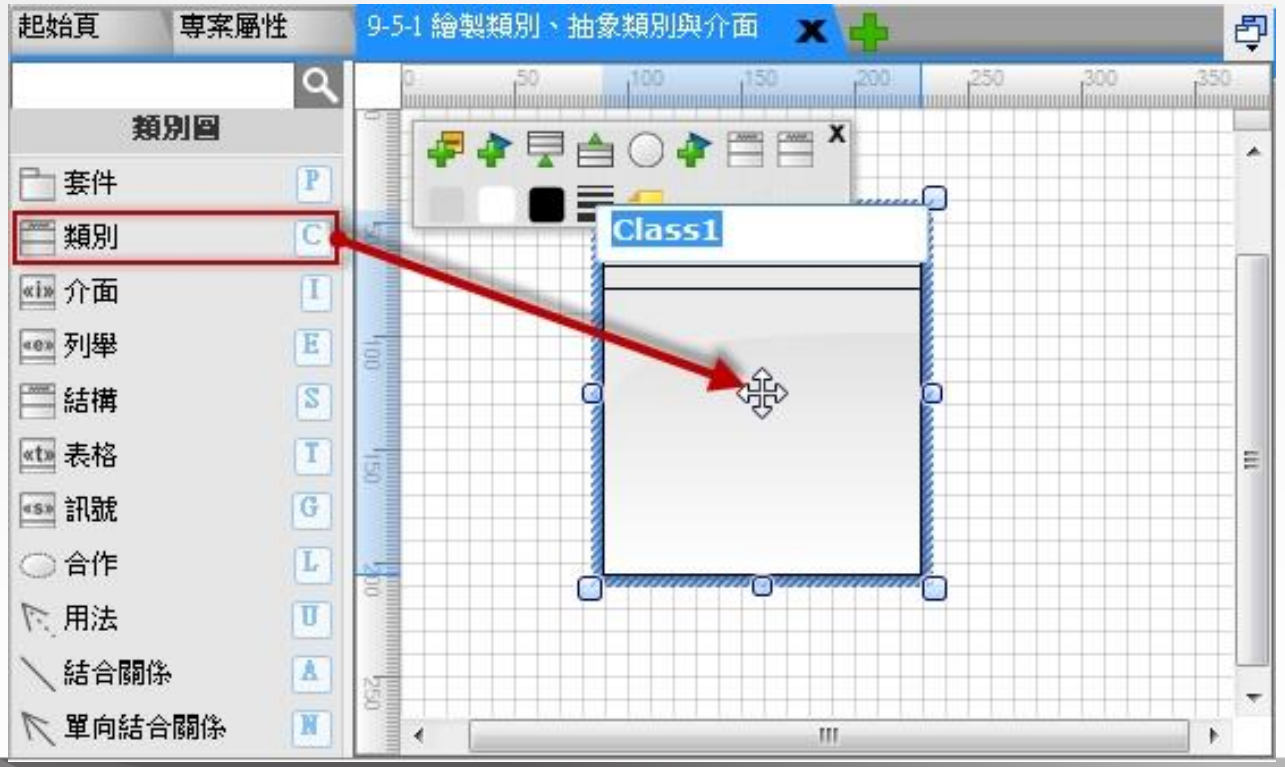


9-5 繪製類別圖與物件圖

- 9-5-1 繪製類別、抽象類別與介面
- 9-5-2 新增類別的屬性清單
- 9-5-3 新增類別的操作清單
- 9-5-4 繪製類別關係
- 9-5-5 繪製物件圖

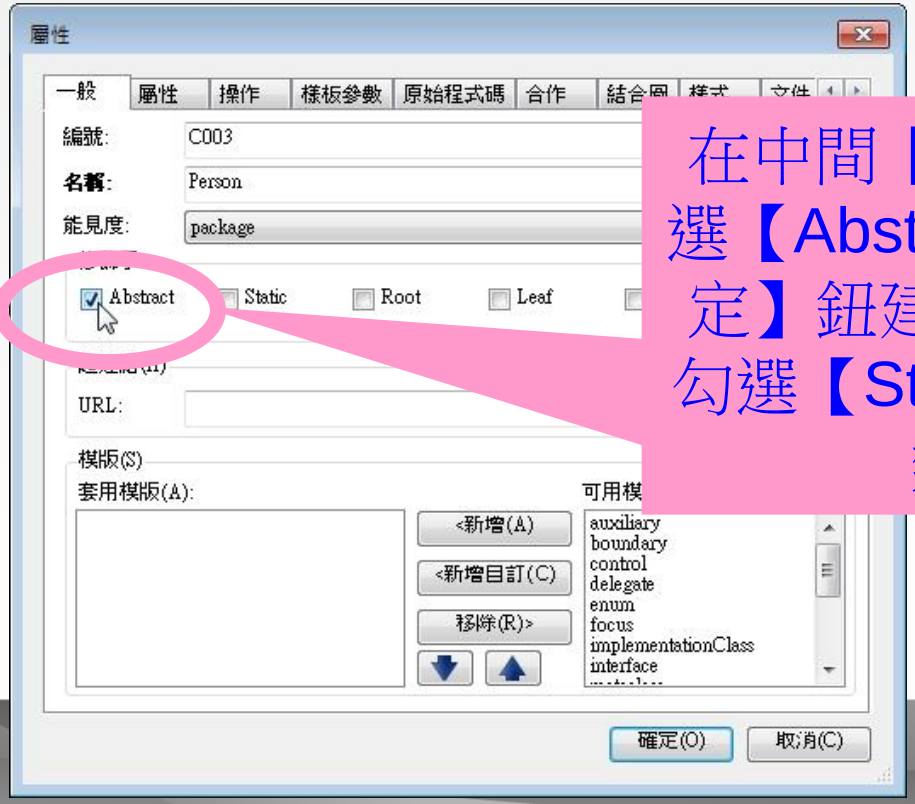
9-5-1 繪製類別、抽象類別與介面-新增類別與介面

- 在「工具箱」視窗拖拉【類別】至編輯區域後，即可新增類別符號，然後輸入類別名稱，如下圖所示：



9-5-1 繪製類別、抽象類別與介面-新增抽象類別

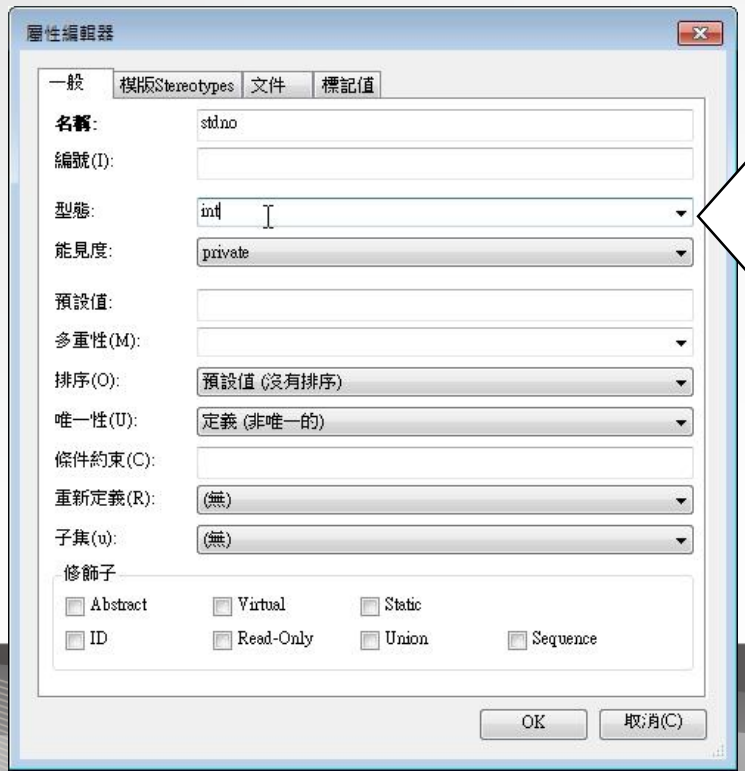
- 請先新增名為Person的類別符號後，選取此符號，執行【右】鍵快顯功能表的【屬性】指令，可以看到「Person – 屬性」對話方塊。



在中間【修飾子】欄勾選【Abstract】，按【確定】鈕建立抽象類別；勾選【Static】建立靜態類別。

9-5-2 新增類別的屬性清單-新增屬性

- 請新增名為Student的類別和選取類別符號，執行【右】鍵快顯功能表的【屬性】指令開啟「屬性」對話方塊。選【屬性】標籤，按右邊【新增】鈕，可以開啟「屬性編輯器」對話方塊。

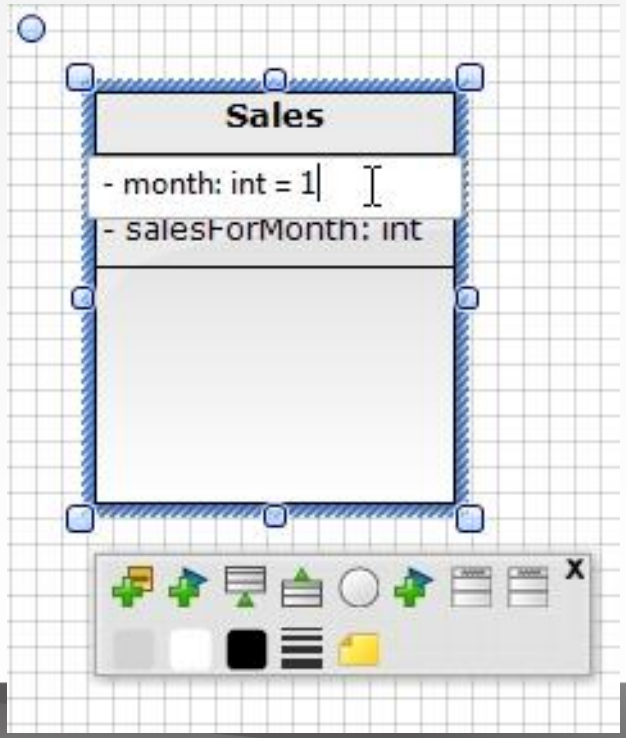


9-5-2 新增類別的屬性清單-設為靜態屬性

- 在類別新增屬性清單後，對於指定屬性，我們可以按【編輯】鈕開啟「屬性編輯器」對話方塊，在最下方「修飾子」框勾選【Static】，將屬性改為靜態屬性。

9-5-2 新增類別的屬性清單-屬性的初值

- 在類別圖上直接點選欲指定初值的屬性，就可以進入屬性的編輯模式，請直接編輯屬性內容加上等號的屬性初值，例如：- month: int = 1，如下圖所示：



9-5-3 新增類別的操作清單-新增操作

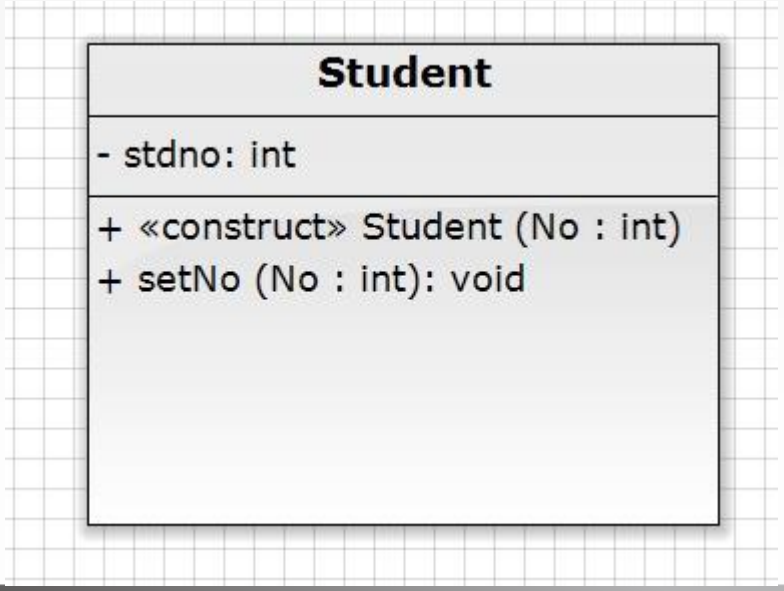
- 請選取Student類別和開啟「屬性」對話方塊，選【操作】標籤，按右邊【新增】鈕，可以開啟「操作編輯器」對話方塊。

9-5-3 新增類別的操作清單-指定靜態操作

- 在「屬性」對話方塊開啟指定操作的「操作編輯器」對話方塊，勾選下方【Static】，就可以將操作改為靜態操作。

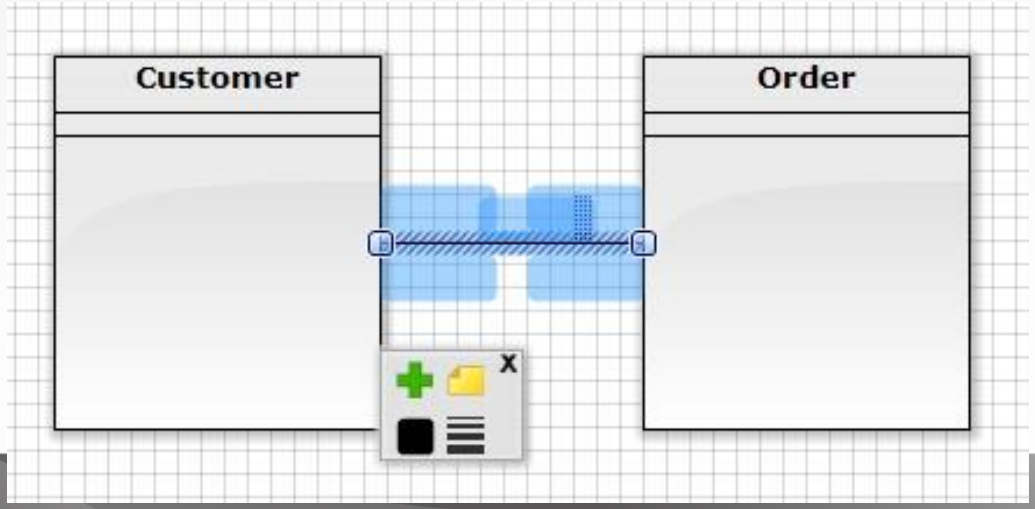
9-5-3 新增類別的操作清單-指定建構子的模版

- 如果新增的是建構子Student()方法，我們需要在「操作編輯器」對話方塊的【模版Stereotypes】標籤，新增名為construct的模版，請按中間【<新增自訂】鈕，可以看到「模版」對話方塊。



9-5-4 繪製類別關係-建立關係

- SIM在類別圖建立關係和使用案例圖相似，我們只需在「工具箱」視窗選擇類別關係的種類，就可以建立類別之間的關係，其中可導覽的結合關係就是【單向結合關係】。
- 例如：選【結合關係】後，移動滑鼠游標，在開始類別按一下，接著拖拉移動滑鼠游標至另一個類別，放開滑鼠按鍵就可以建立之間關係的連接線，如下圖所示：

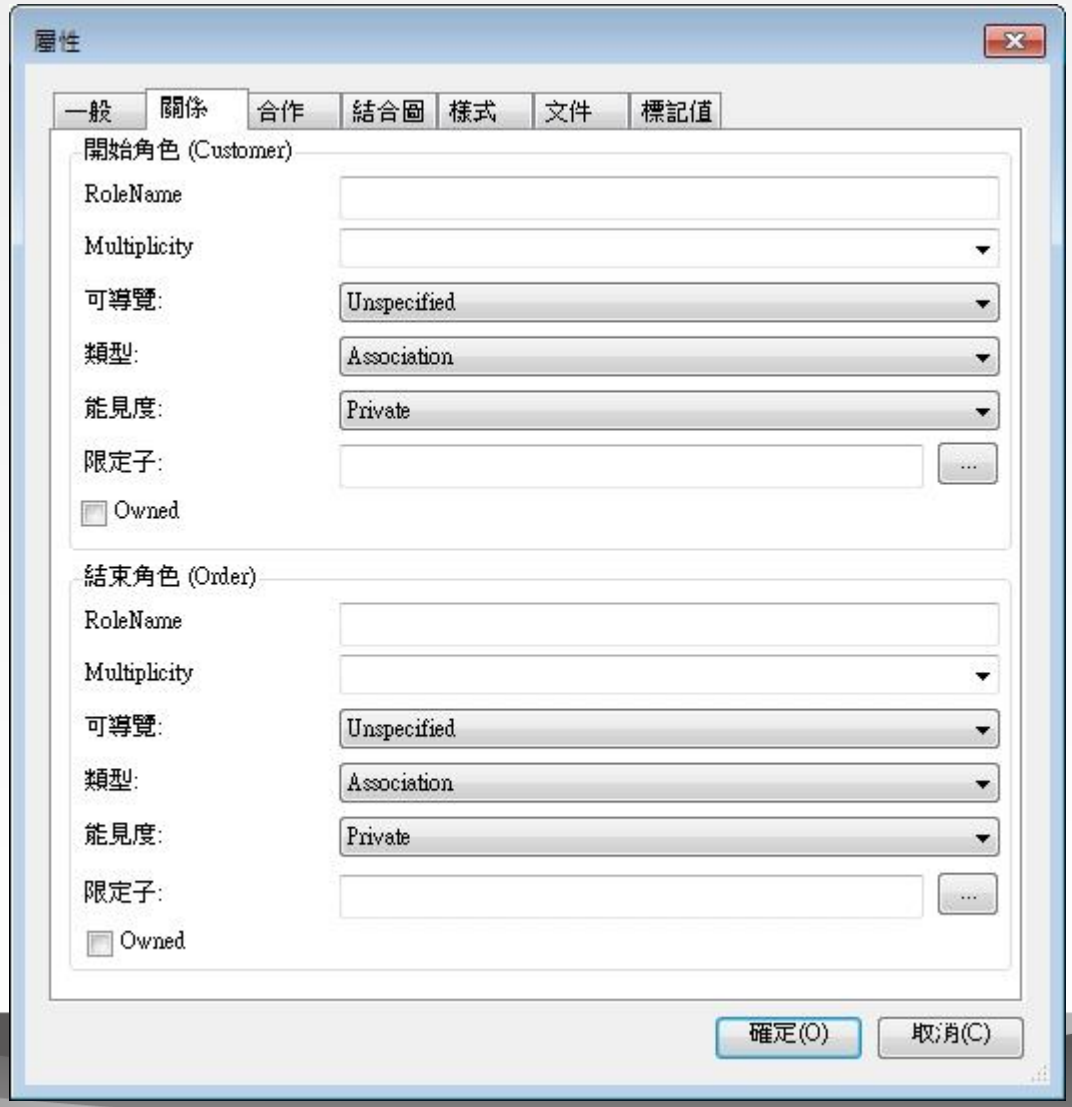


9-5-4 繪製類別關係-建立反身關係

- 反身關係的連接線是連向自己，請在「工具箱」視窗選擇類別關係的種類後，在類別符號上按一下，拖拉且仍停留在同一個類別範圍內，放開滑鼠按鍵就可以建立反身關係。

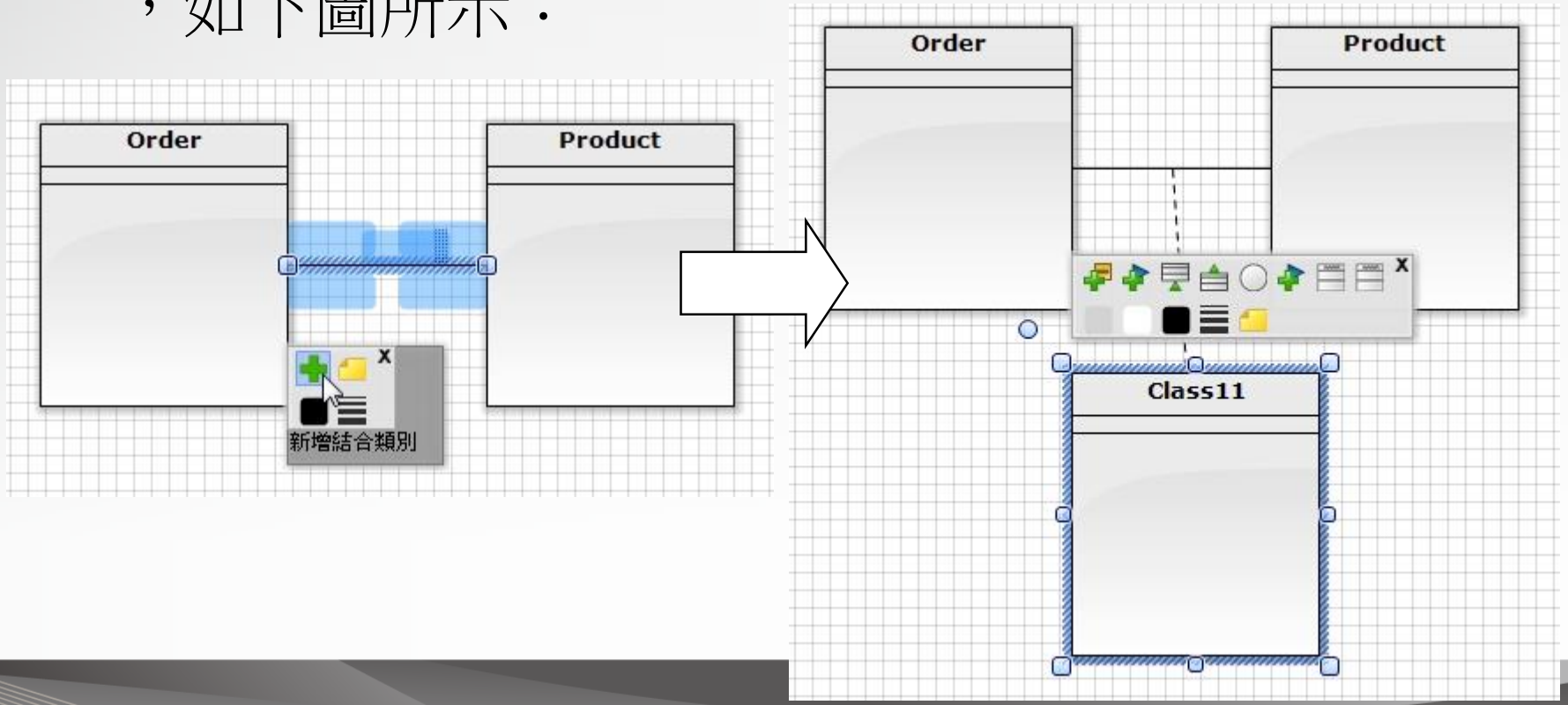
9-5-4 繪製類別關係-設定多重性與角色名稱

- 在建立類別之間的關係後，請選取連接線，執行【右】鍵快顯功能表的【屬性】指令，可以看到「屬性」對話方塊。選【關係】標籤，就可以指定開始與結束的角色名稱和多重性，如右圖所示：



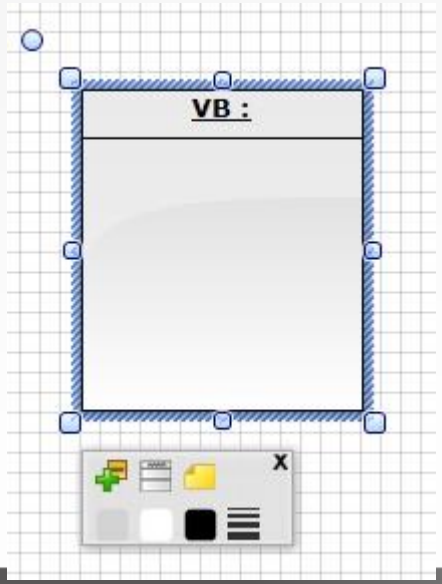
9-5-4 繪製類別關係-建立結合類別

- 對於結合類別（Association Class），SIM是在類別連接線上提供按鈕來建立2個類別之間的結合類別，如下圖所示：



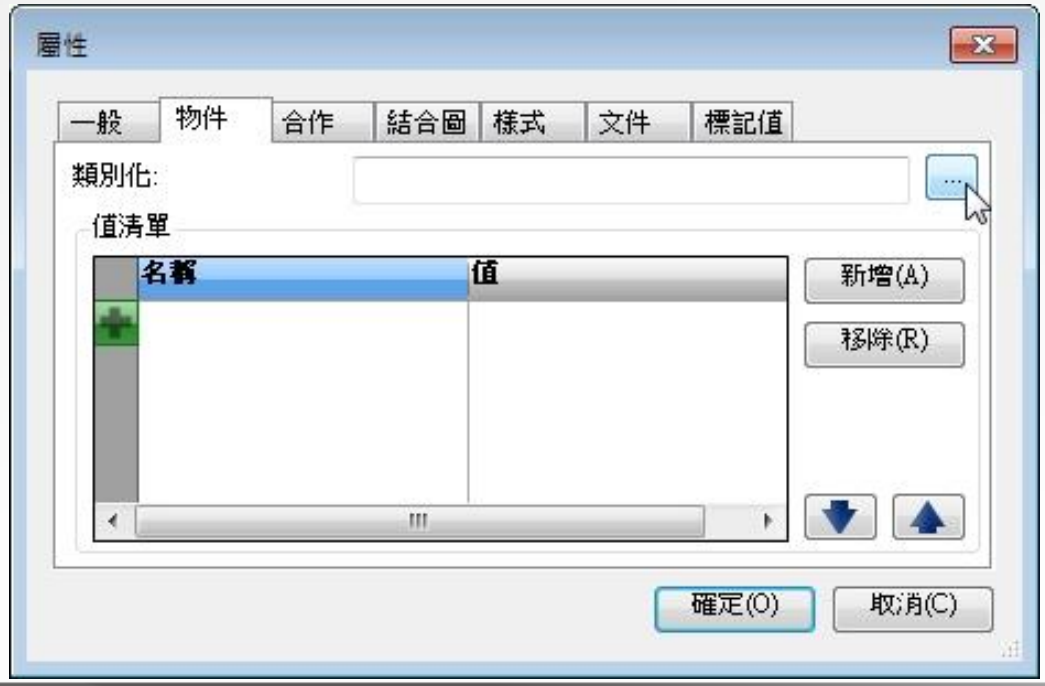
9-5-5 繪製物件圖-新增

- 物件圖的建立和類別圖相同，主要差異在於物件名稱包含類別，而且屬性清單有屬性值。請在「工具箱」視窗拖拉【物件】建立物件符號後，就可以直接輸入物件名稱，如下圖所示：

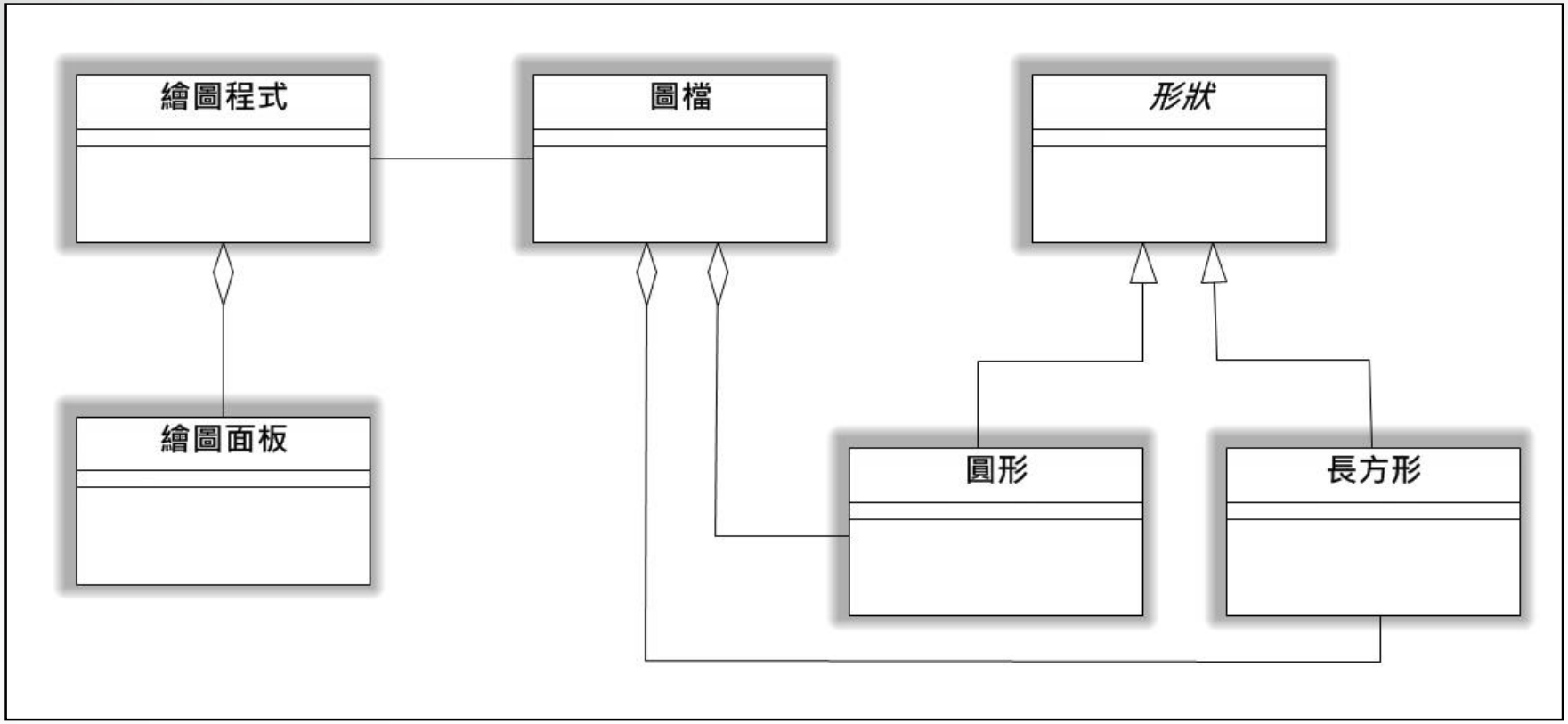


9-5-5 繪製物件圖-指定屬性值的屬性清單

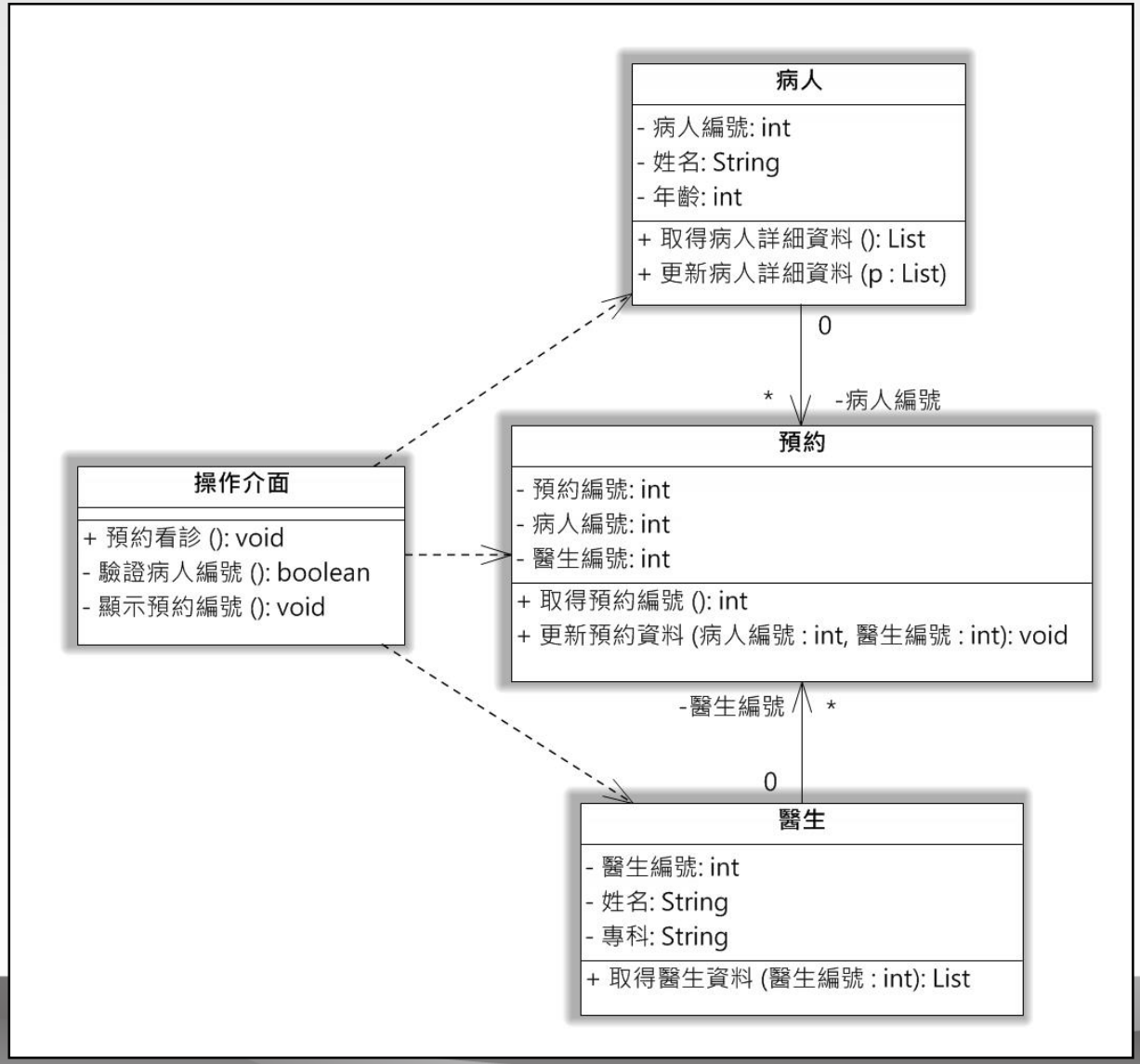
- 請選擇模型中的類別，以此例是【圖書】，按【確定】鈕，可以看到下方列出可以指定屬性值的屬性清單，如下圖所示：



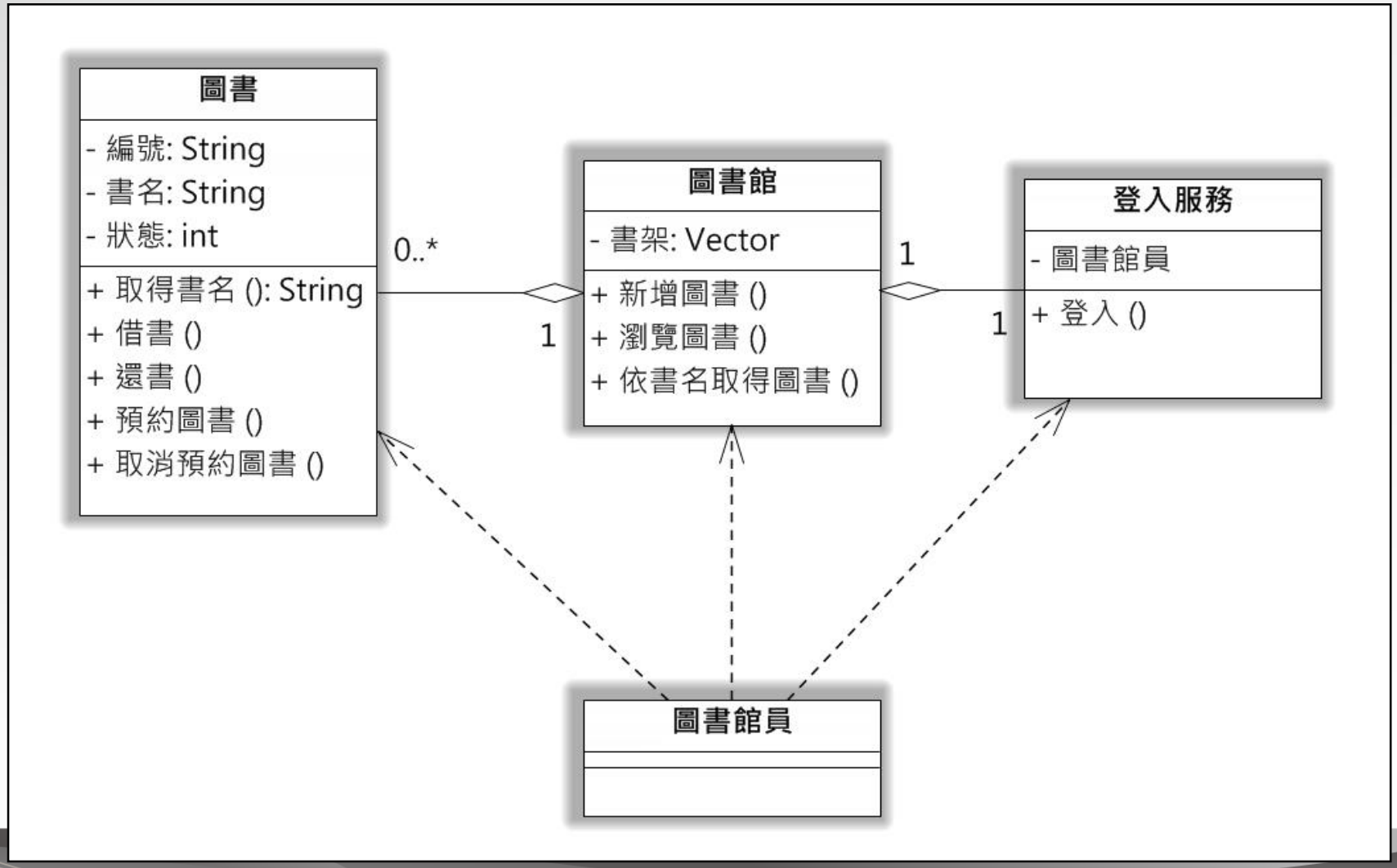
9-6 綜合練習-繪圖程式的類別圖



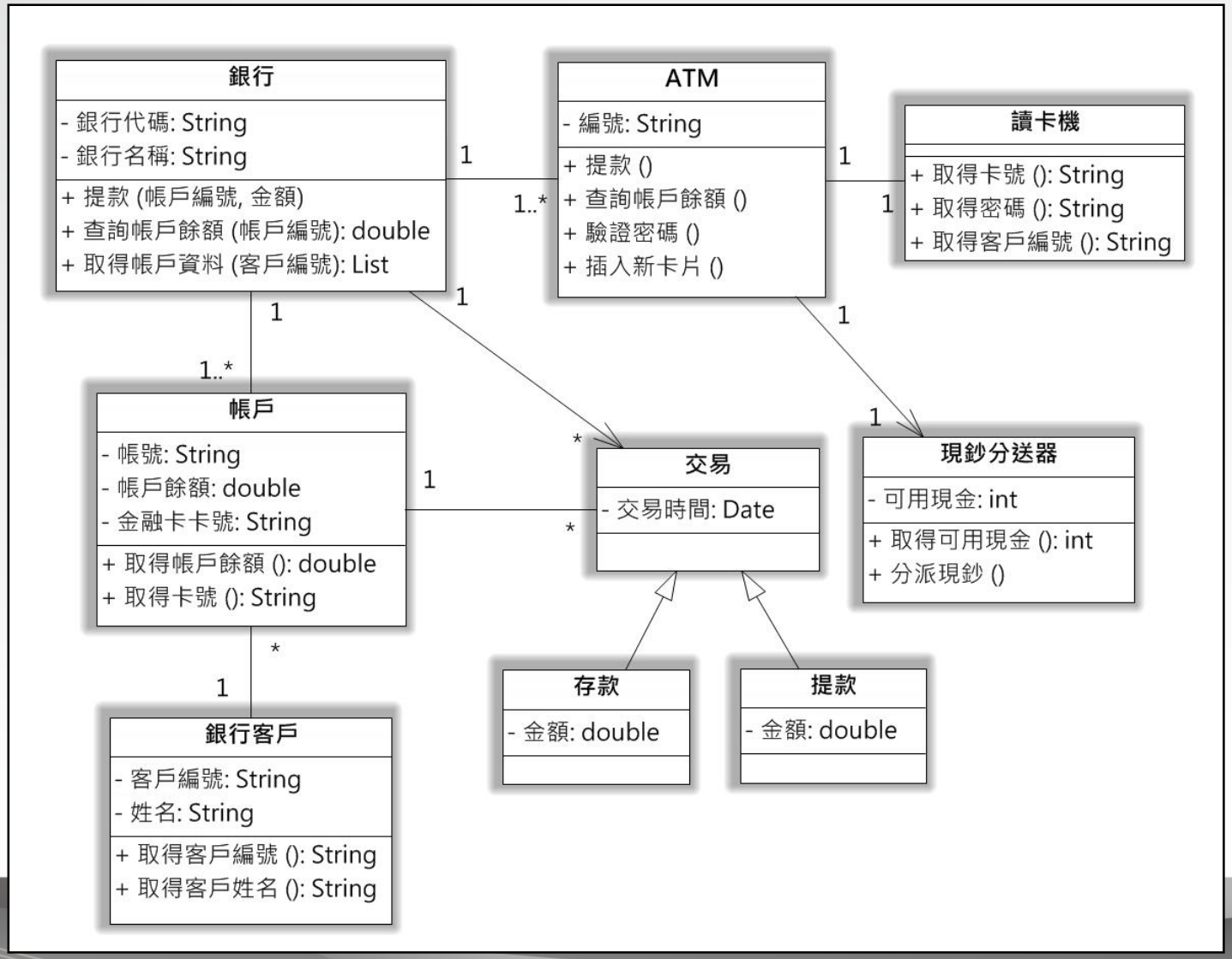
9-6 綜合練習-醫院管理系統的類別圖



9-6 綜合練習-圖書館系統的類別圖



9-6 綜合練習-ATM自動櫃員機系統的類別圖



End