

第7章 活動圖

- 7-1 活動圖的基礎
- 7-2 基本的活動圖
- 7-3 進階的活動圖
- 7-4 活動圖的開始與結束
- 7-5 繪製活動圖
- 7-6 綜合練習



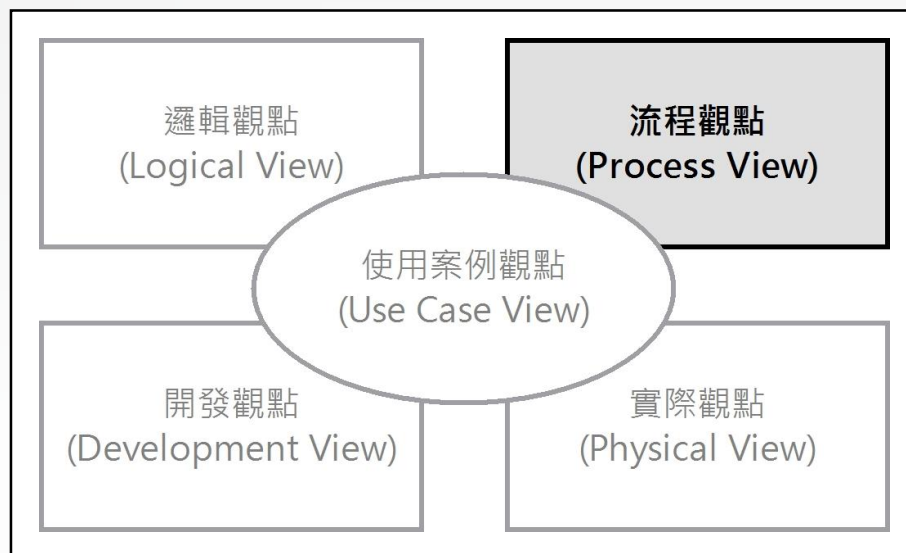


7-1 活動圖的基礎-說明

- 「活動圖」（Activity Diagram）是UML眾多圖形之中最容易了解的圖形，因為就算是軟體系統的利益相關者（Stakeholders），也可以看懂類似流程圖的活動圖，換句話說，在需求階段我們可以使用活動圖來與客戶進行溝通。
- 在UML 1.x版的活動圖只能算是狀態機圖的一種特例，可以用來定義進入某狀態時發生的過程和執行的功能。UML 2.x版清楚切割活動圖和狀態機圖，大幅擴充舊版活動圖的功能和視野，如今的活動圖不再只是特殊版本的狀態機圖，而是能夠繪出任何流程圖的UML圖形。

7-1 活動圖的基礎-目的

- 活動圖的主要目的是描述系統的商業流程，使用案例（**Use Case**）情節的流程和操作步驟，即工作流程和所需的作業和活動，其應用範圍十分廣大，在物件導向分析和設計階段都可以使用活動圖來增強塑模的效益。活動圖可以呈現**4+1**觀點軟體系統模型的流程觀點，如下圖所示：



7-1 活動圖的基礎-用途

■ 使用案例可以顯示系統需要做什麼，活動圖能夠描述系統如何完成所需做的事。在UML塑模過程的活動圖可以用來描述：

- 商業規則（**Business Rules**）的商業流程，這是一組協調完成商業目的的工作，例如：訂單流程。
- 單一使用案例情節的流程，例如：訂票、設定時間和借書等。
- 一系列複雜的使用案例。
- 並行處理的操作。
- 行程或執行緒的流程。
- 軟體流程和邏輯控制結構。

7-1 活動圖的基礎-活動與動作

- 活動（**Activity**）和動作（**Actions**）常常被使用者誤用，在進入本章主題的活動圖前，我們需要先區分清楚之間的差異，如下所示：
 - 動作：動作是完成一個過程中的步驟，可能是一項計算，或一件工作或作業。
 - 活動：活動是我們描述的一個完整過程，動作只是其中的步驟。





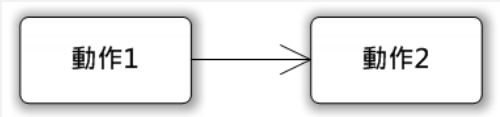
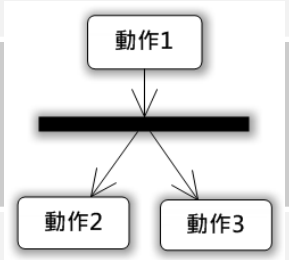
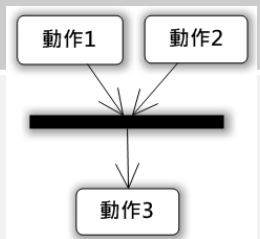


7-2 基本的活動圖

- 7-2-1 活動圖的基本符號
- 7-2-2 數學運算流程
- 7-2-3 使用案例的流程
- 7-2-4 系統的商業流程



7-2-1 活動圖的基本符號

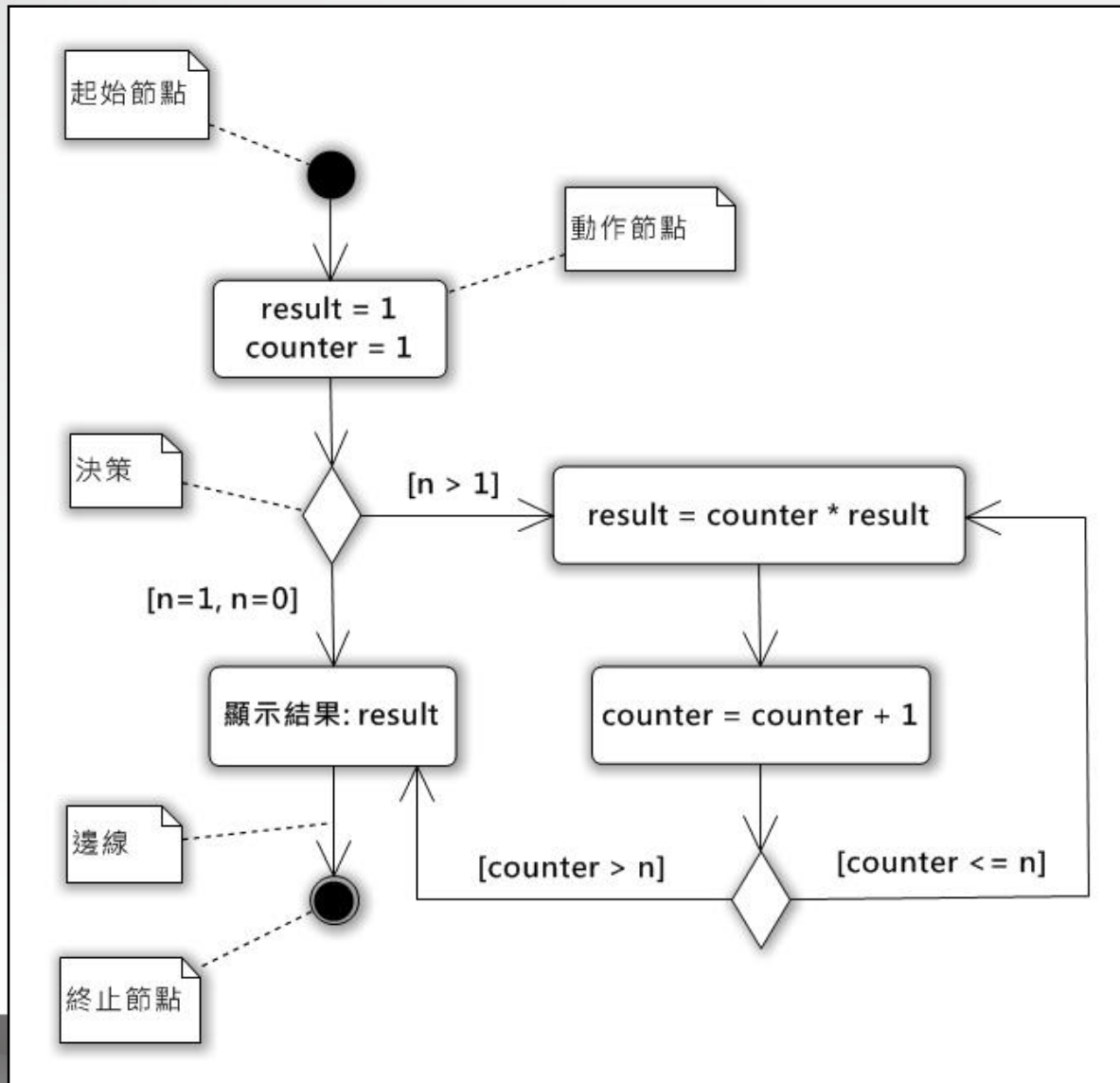
符號	說明
	實心圓形代表活動起始節點
	同心圓形代表活動終止節點
	圓角長方形表示動作節點，也就是執行的操作或作業
	菱形代表決策或合併節點
	連接節點的箭頭線稱為邊線或路徑，標示節點之間控制流程的轉換，進入節點稱為進入邊線（Incoming Edge）；離開節點稱為離開邊線（Outgoing Edge）
	路徑從一條分成多條稱為分岔（Fork）
	路徑從多條結合成一條稱為結合（Join）

7-2-2 數學運算流程-說明

- 活動圖可以取代流程圖建立複雜的程式流程，或數學運算過程，例如：數學階層函數 $n!$ 的運算流程，如下所示：

$$n! = 1 * 2 * \dots * (n-2) * (n-1) * n$$

7-2-2 數學運算流程-活動圖



7-2-3 使用案例的流程-說明

- 使用案例圖只是使用圖形表示系統的服務與功能，每一個完整使用案例除了圖形外，還包含描述的故事，稱為情節（**Scenarios**），我們可以使用活動圖描述情節的流程（**Flow**），或稱工作流程（**Workflow**），進一步說明請參閱第8章。

7-2-3 使用案例的流程-主流程

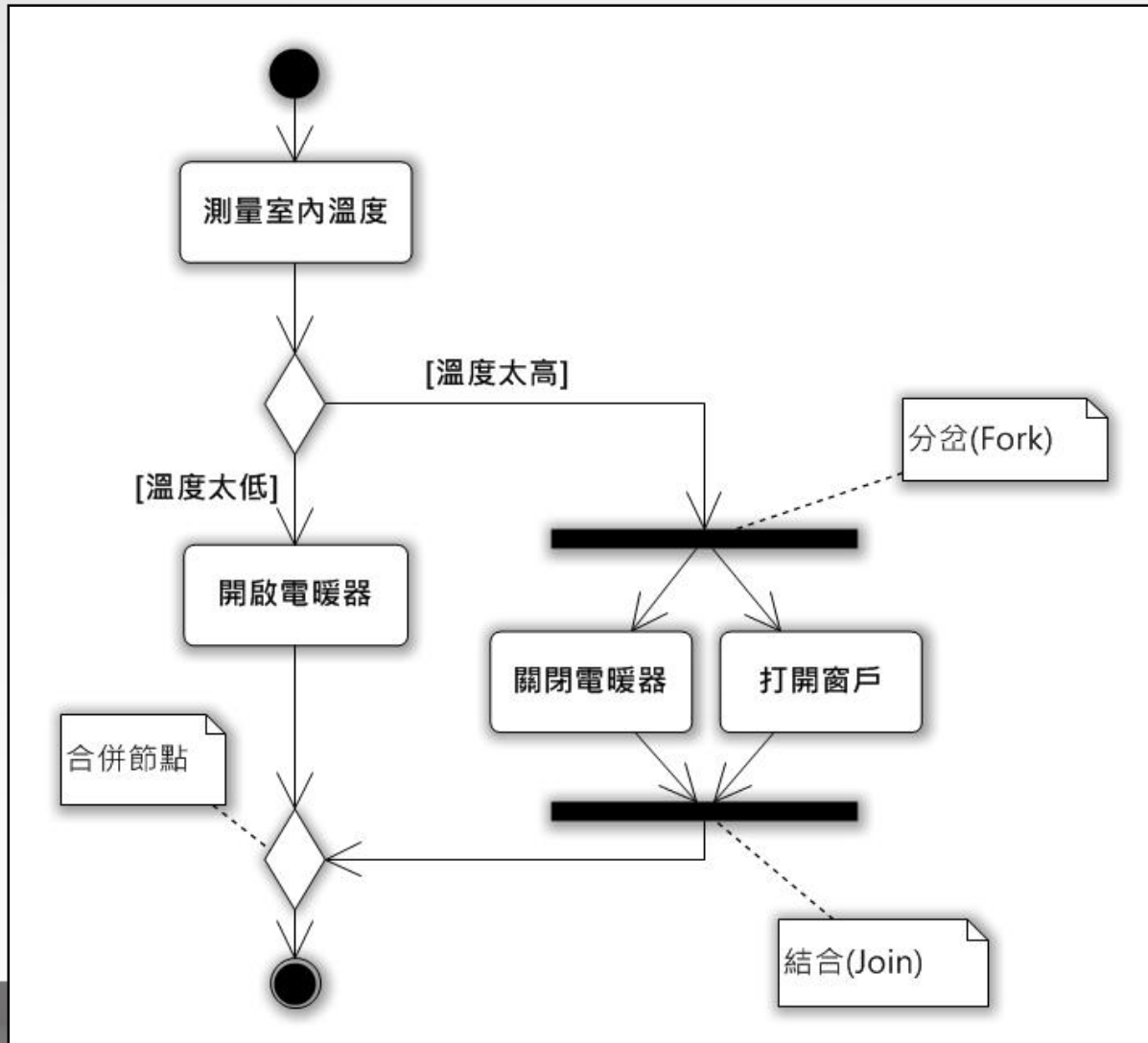
調整室內溫度使用案例的主流程

- 調整室內溫度使用案例可以在夏天維持室內溫度在一定範圍之內，其主流程如下所示：

主流程

1. 系統測量室內溫度。
2. 如果溫度太高：
 - 2.1. 關閉電暖器和打開窗戶。
3. 否則，溫度太低：
 - 3.1. 開啟電暖器。

7-2-3 使用案例的流程-活動圖



7-2-3 使用案例的流程-平行路徑

- 平行路徑（**Concurrent Paths**）的分岔（**Fork**）與結合（**Join**），表示這些路徑的動作可以同時處理，並不會互相影響，其說明如下所示：
 - 分岔（**Fork**）：將一條路徑使用黑色粗線的同步棒（**Synchronization Bar**）分成多條不同路徑，以此例是2條【關閉電暖器】和【打開窗戶】路徑，表示2條路徑是平行執行，因為各條路徑中的動作不會相互依賴，關閉電暖器不會影響打開窗戶。
 - 結合（**Join**）：將多條路徑使用黑色粗線的同步棒合併成一條路徑，表示多條路徑都已經完成後，才合併進行下一個動作，以此例是完成【關閉電暖器】且完成【打開窗戶】動作。

7-2-4 系統的商業流程-流程描述

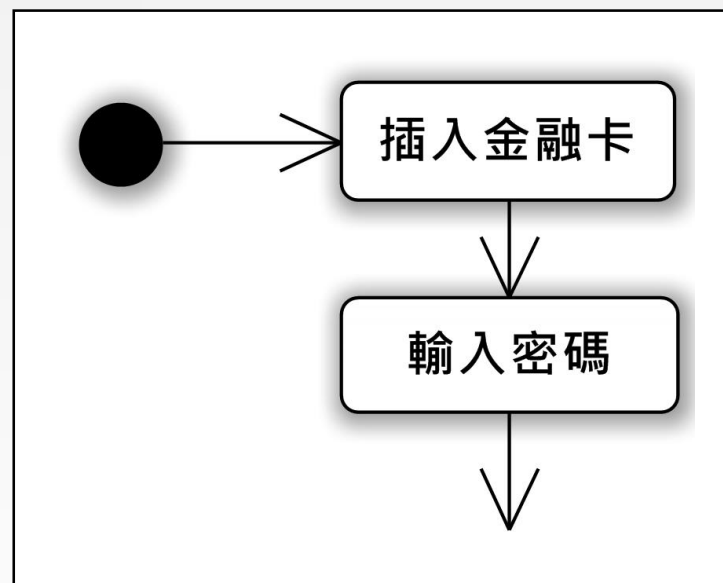
ATM自動櫃員機的操作流程

- 在建立ATM自動櫃員機系統的需求階段，我們取得自動櫃員機提款流程的描述，如下所示：

客戶將金融卡插入自動櫃員機後，輸入卡片密碼，如果密碼正確，就可以選擇提款功能，然後輸入提款金額，當帳戶餘額足夠時，就吐出鈔票，接著退出和取出卡片，和列印收據，完成整個提款流程。

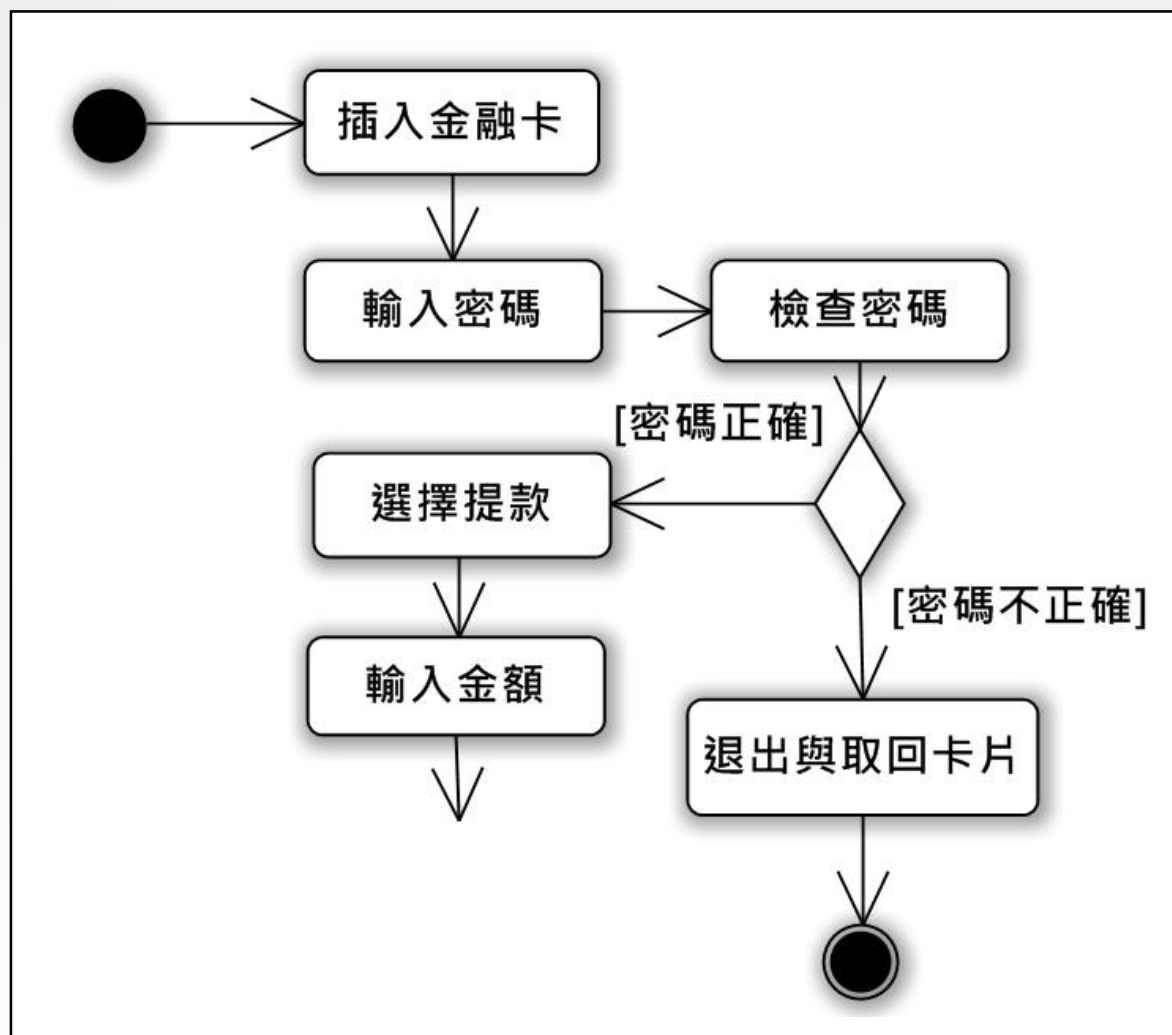
7-2-4 系統的商業流程-步驟一

- Step 1：從起始節點開始，插入金融卡和輸入密碼，如右圖所示：



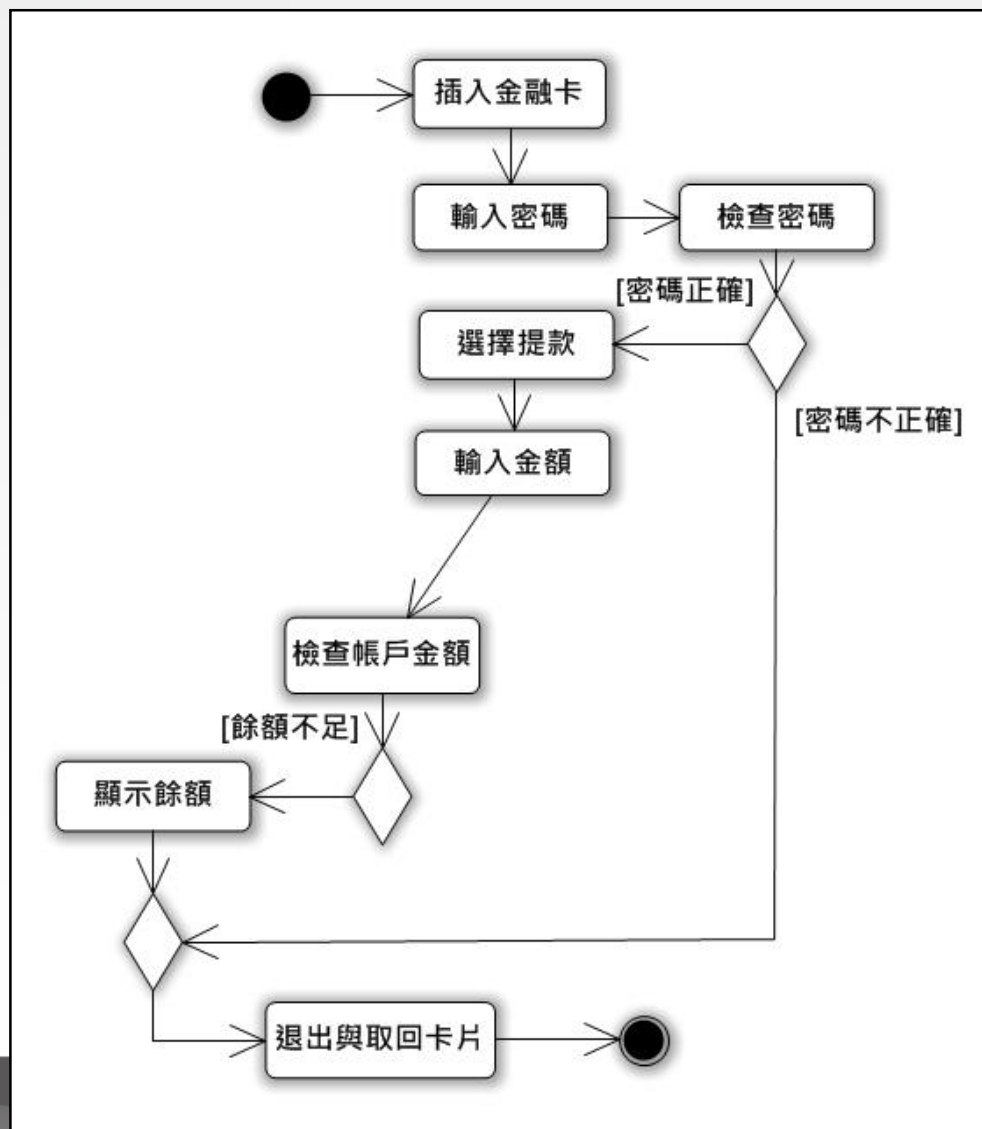
7-2-4 系統的商業流程-步驟二

- Step 2：檢查密碼是否正確，不正確就退出與取回卡片，然後結束；正確選擇提款功能和輸入提款金額，如右圖所示：



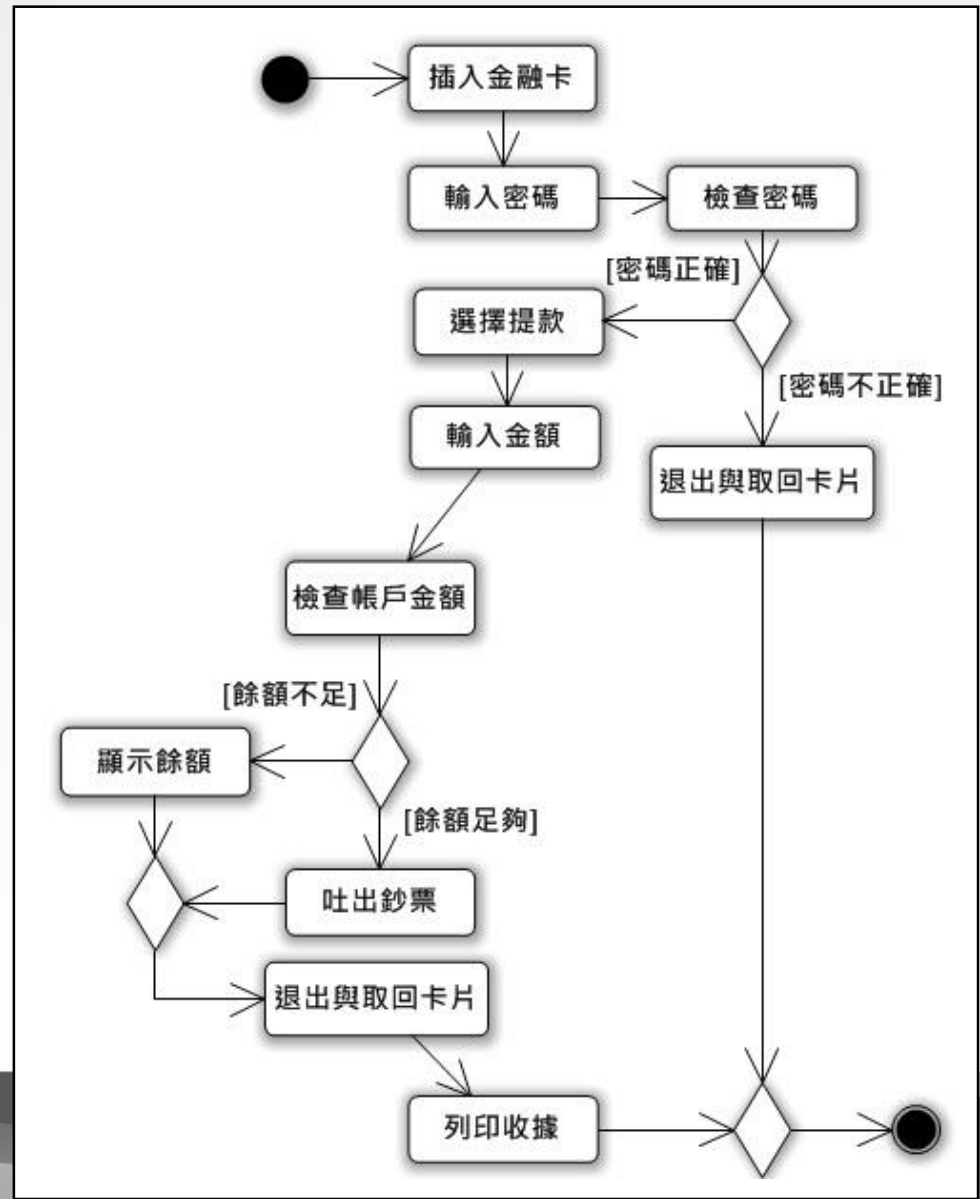
7-2-4 系統的商業流程-步驟三

- Step 3：在輸入提款金額後，檢查帳戶餘額是否足夠，餘額不足，自動櫃員機會顯示帳戶餘額和退出與取回卡片，如右圖所示：



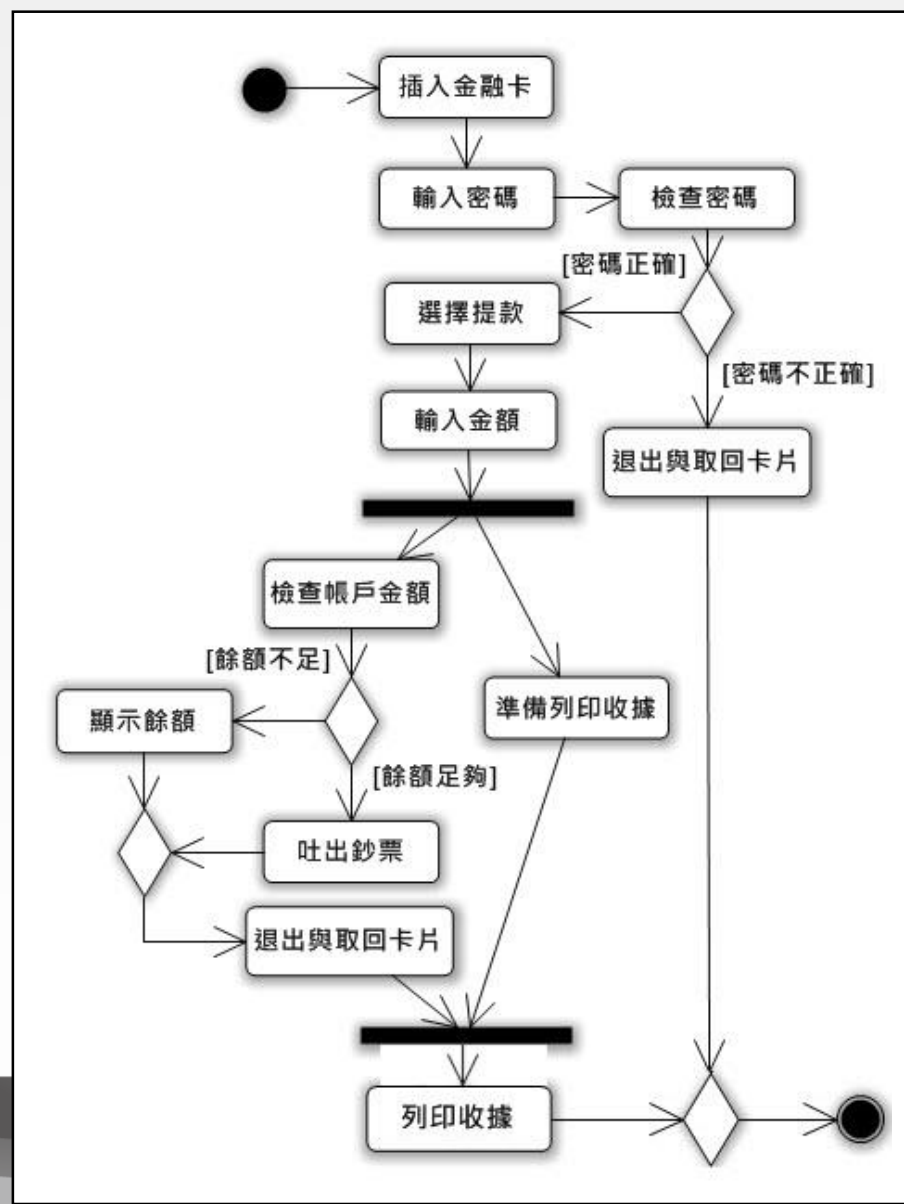
7-2-4 系統的商業流程-步驟四

- Step 4：餘額足夠，就從自動櫃員機吐出鈔票，在退出與取回卡片後，列印收據，如右圖所示：



7-2-4 系統的商業流程-步驟五

- Step 5：收據是直到客戶取回卡片後，才會列印，所以準備列印收據和提款過程使用分岔和結合的同步處理，如右圖所示：





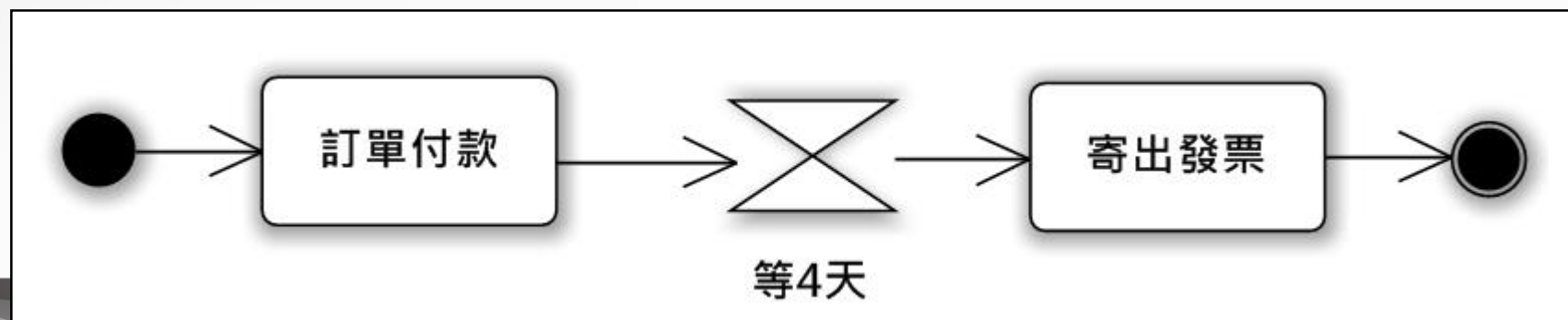
7-3 進階的活動圖

- 7-3-1 時間事件動作
- 7-3-2 呼叫其他活動
- 7-3-3 物件流程轉換
- 7-3-4 送出或接收訊號
- 7-3-5 泳道
- 7-3-6 連接器
- 7-3-7 擴展區與擴展節點



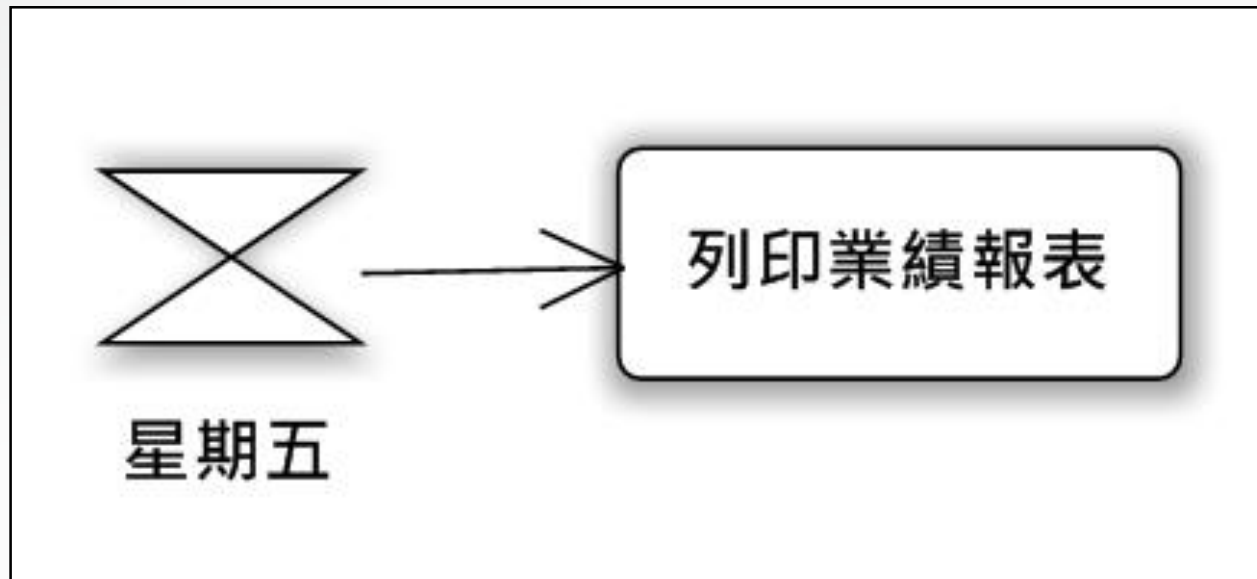
7-3-1 時間事件動作-等待一段時間

- 時間事件（Time Event）是在流程中標示等待時間，因為有時我們需要等待一段時間後，才會進行下一個動作，或周期定時執行一些動作，例如：在訂單付款後，等待4天寄出發票；每個星期五列印業績報表等。
- 在活動圖是使用沙漏（Hourglass）符號表示時間事件動作，如下圖所示：




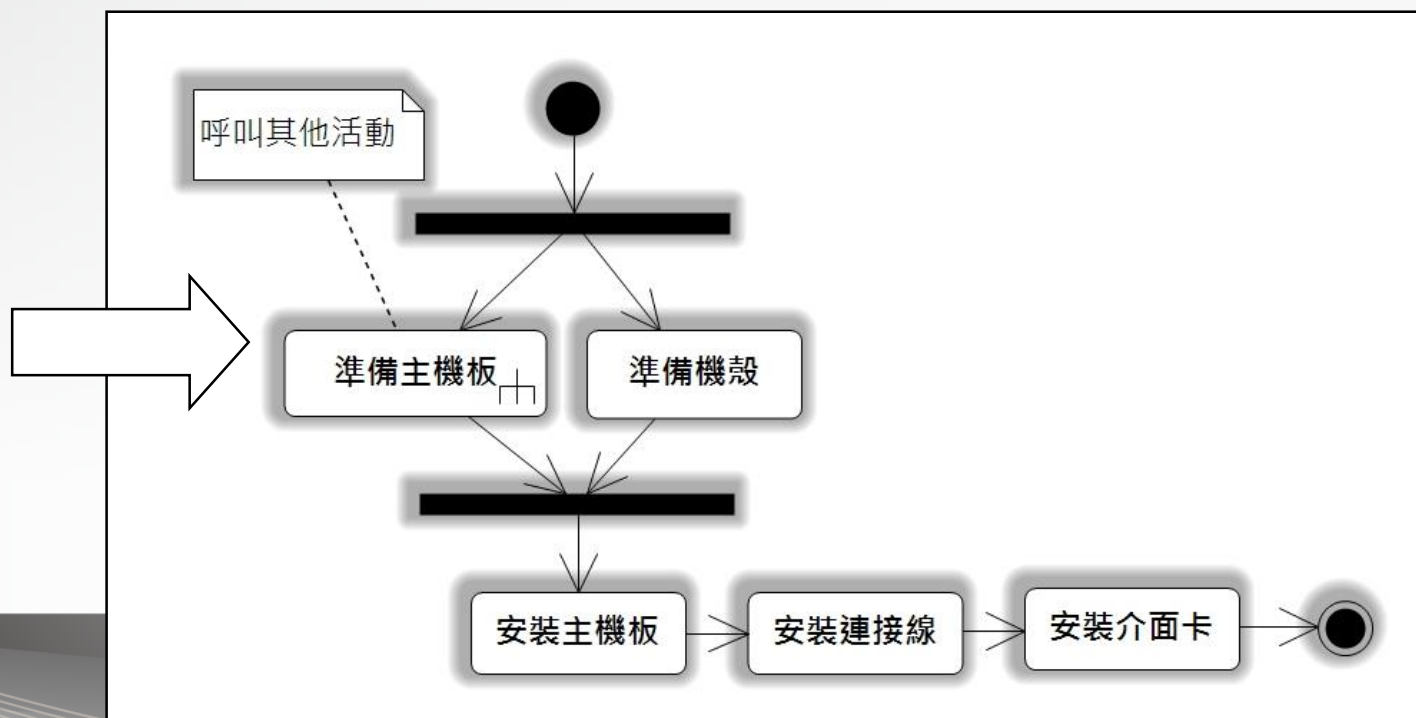
7-3-1 時間事件動作-重複操作的周期動作

- 時間事件動作節點可以取代起始節點，即沒有進入節點的進入邊線（**Incoming Edge**），此時，時間事件動作代表重複操作的周期動作，例如：每個星期五都會列印業績報表，如下圖所示：



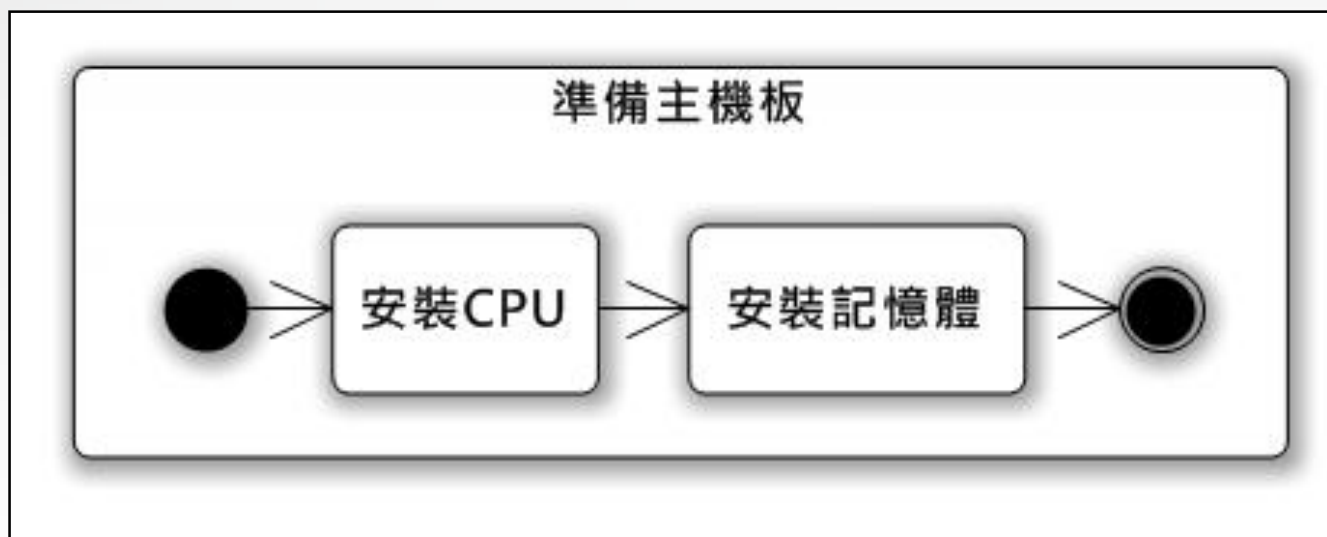
7-3-2 呼叫其他活動-呼叫

- 活動圖的流程如果描述太詳細，有可能會讓活動圖尺寸變的太大，我們可以將一些相同流程的重複動作建立成獨立活動，然後在動作節點呼叫其他活動，使用的是「」符號，如下圖所示：



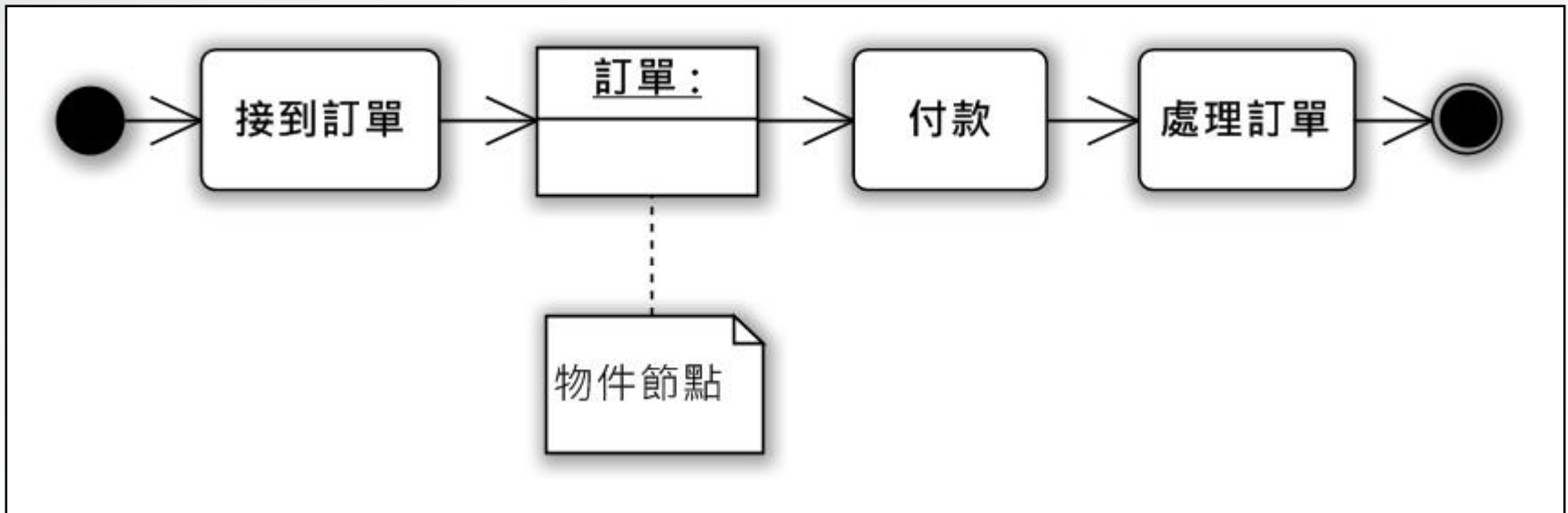
7-3-2 呼叫其他活動-被呼叫

- 活動圖的【準備主機板】動作之前有「 \pitchfork 」符號，表示它是呼叫名為【準備主機板】的活動，如下圖所示：



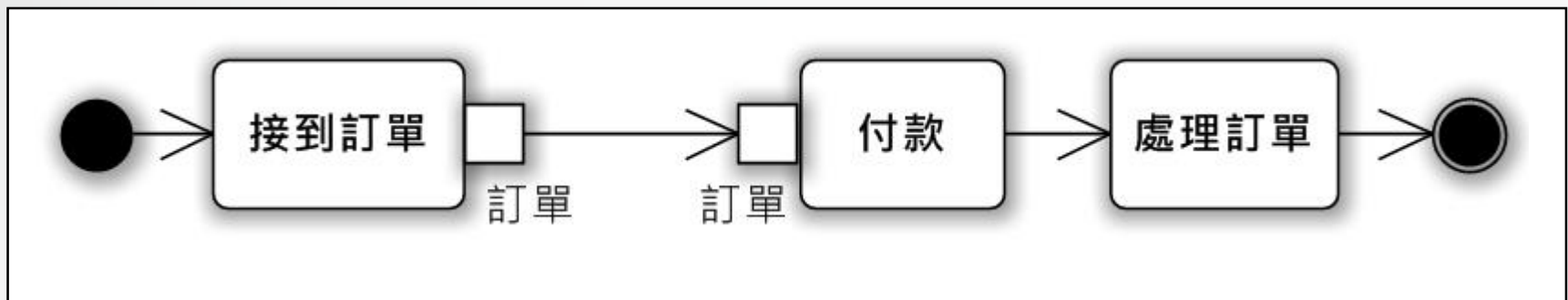
7-3-3 物件流程轉換- 在動作之間傳遞物件

- 在活動圖的動作之間可以傳遞物件節點，這是使用長方形符號表示的節點，例如：訂單處理活動圖中的訂單節點是一個物件節點，如下圖所示：



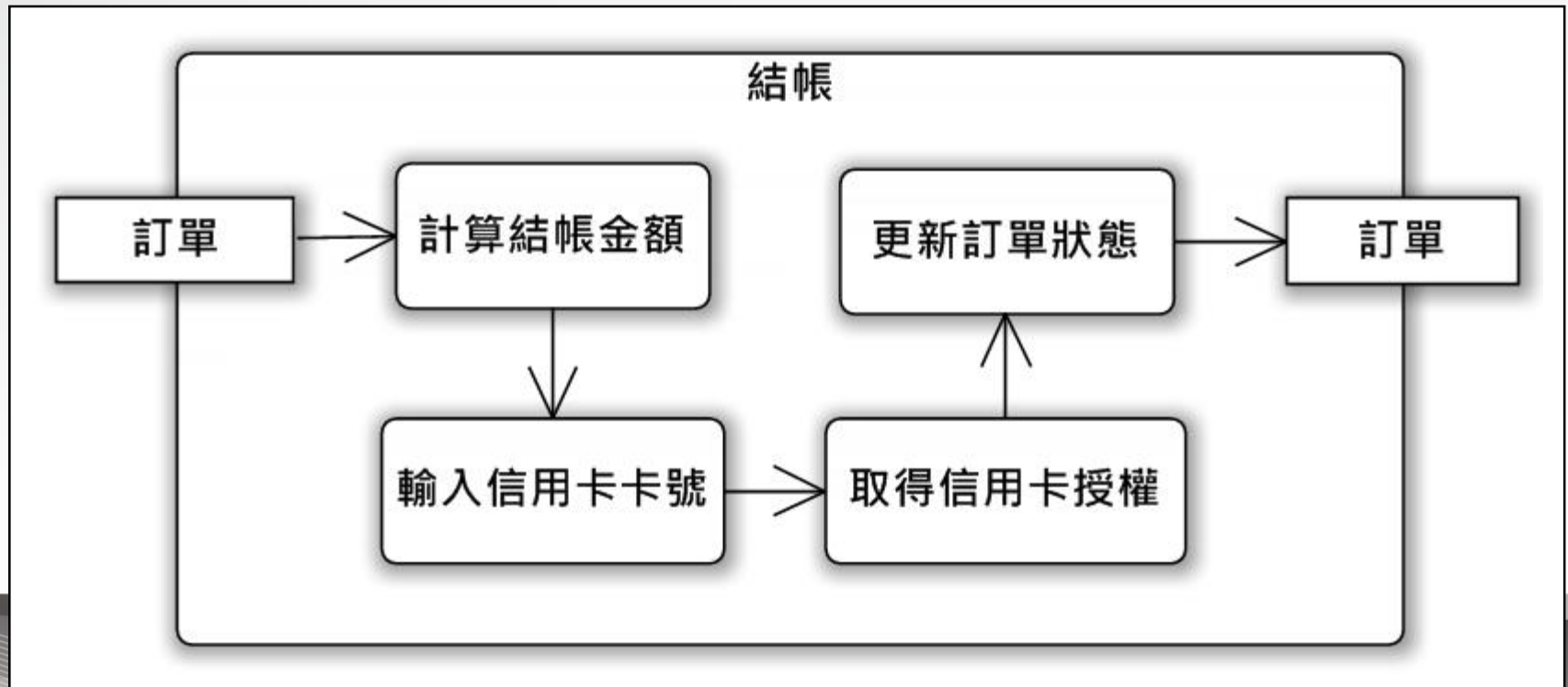
7-3-3 物件流程轉換- 動作的輸出與輸入物件

- 活動圖除了使用物件節點外，也可以使用腳位（**Pins**）來明顯標示動作的輸出與輸入，例如：將前述訂單處理活動圖改為腳位，如下圖所示：



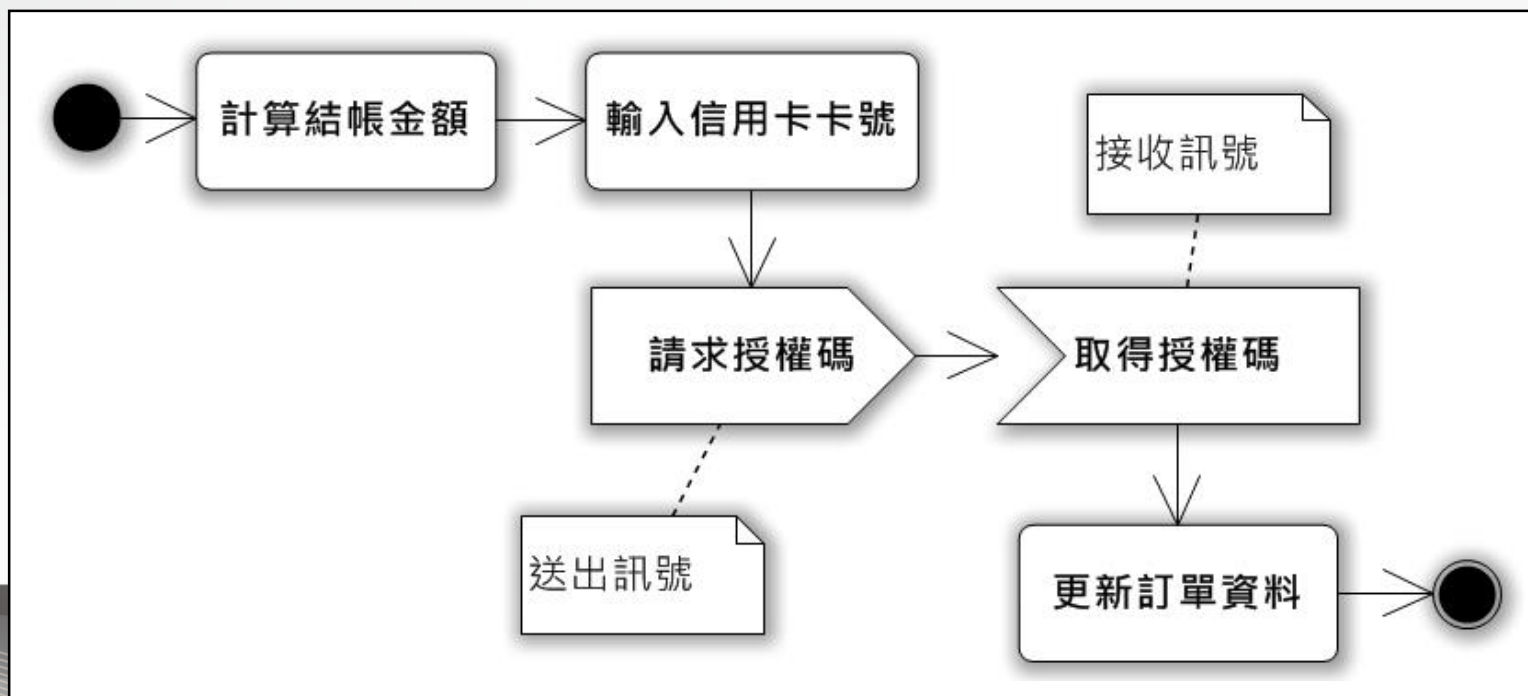
7-3-3 物件流程轉換- 活動參數節點

- 活動參數節點（Activity Parameter Node）是活動的輸入與輸出物件（物件節點是動作的輸入與輸出），可以在活動框架的邊界上明確標示活動的輸入與輸出物件，例如：網路商店結帳流程的活動圖，如下圖所示：



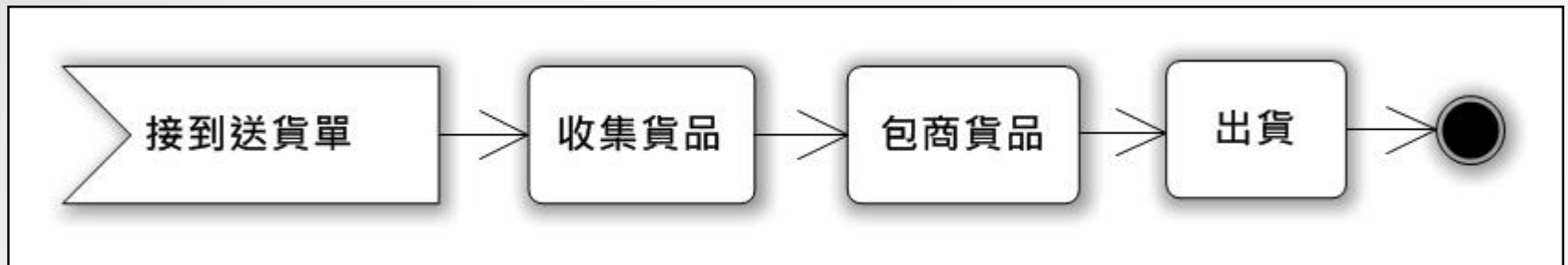
7-3-4 送出或接收訊號-送出訊號

- 活動圖的訊號（**Signals**）代表外部參與者的人員、系統或流程，分為送出或接收訊號兩種。訊號是一個訊息送給外部參與者，然後接收到外部參與者的回應訊號後，再進行下一步動作。例如：網路購物的結帳流程需要取得信用卡公司的授權，如下圖所示：



7-3-4 送出或接收訊號-接收訊號

- 接收訊號可以取代起始節點作為流程的開始，例如：倉庫出貨流程，如下圖所示：

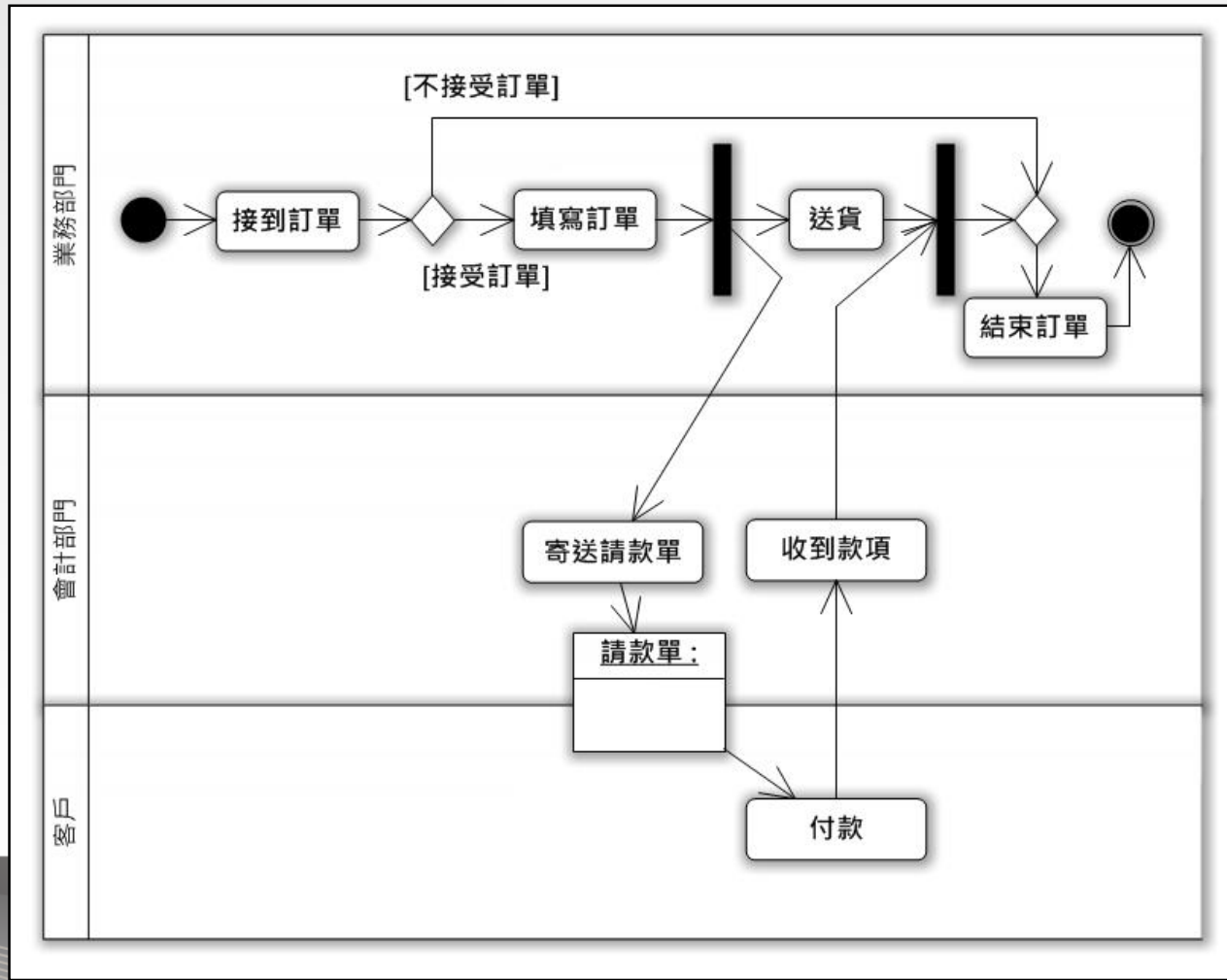


7-3-5 泳道-說明

- 在活動圖中如果有組織或公司的不同部門或角色參與，例如：公司的業務、會計部門和客戶都參與訂單處理，使用者、**ATM**和銀行參與自動提款機的提款流程等。泳道（**Swimlane**）的主要目的就是清楚分割不同參與角色的動作。
- **UML**活動圖可以如同游泳池的水道一般分割成多個水平或垂直區段，稱為泳道（**Swimlane**），在每一個泳道可以顯示角色名稱，表示泳道的內容是此角色進行的相關動作。

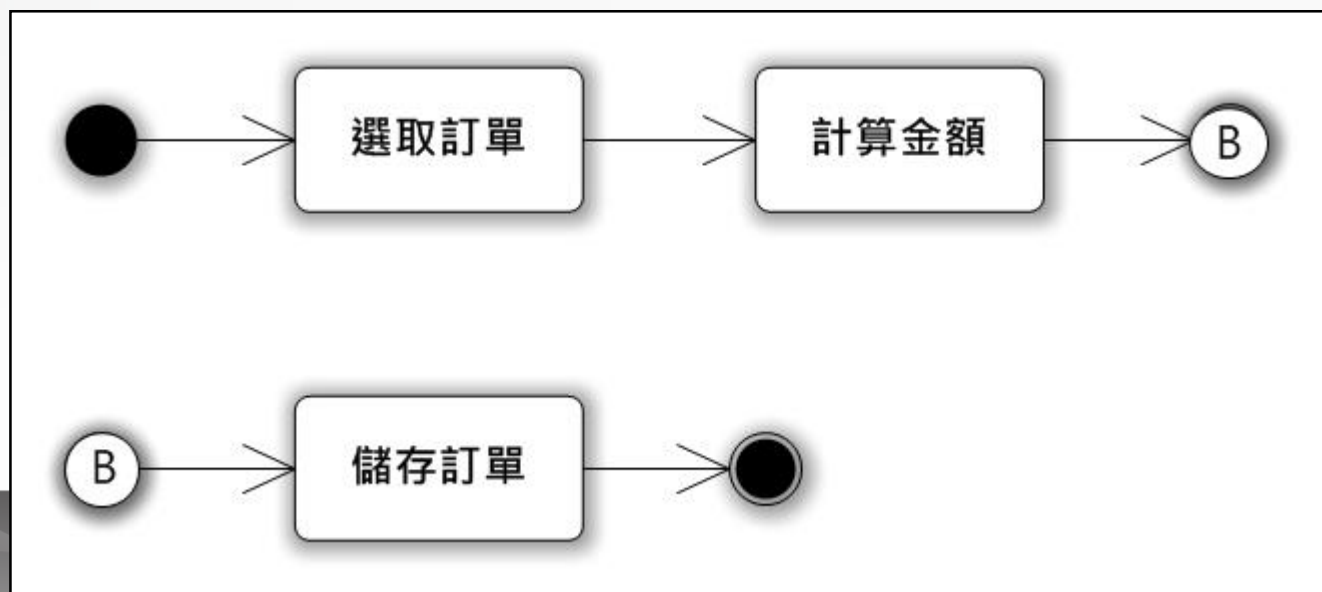
7-3-5 泳道-範例

■ 例如：公司訂單處理的活動圖，如下圖所示：



7-3-6 連接器

- 連接器（Connectors）是當活動圖太大或太複雜時，因為邊線可能跨過多條其他邊線，造成閱讀上的困擾，我們可以使用連接器來分割邊線。
- 連接器是使用小圓形符號中間有名稱來表示，名稱通常是使用單一英文字母，而且連接器是成雙成對的，因為我們是比對相同名稱連接器，表示邊線是應該連在一起的，如下圖所示：

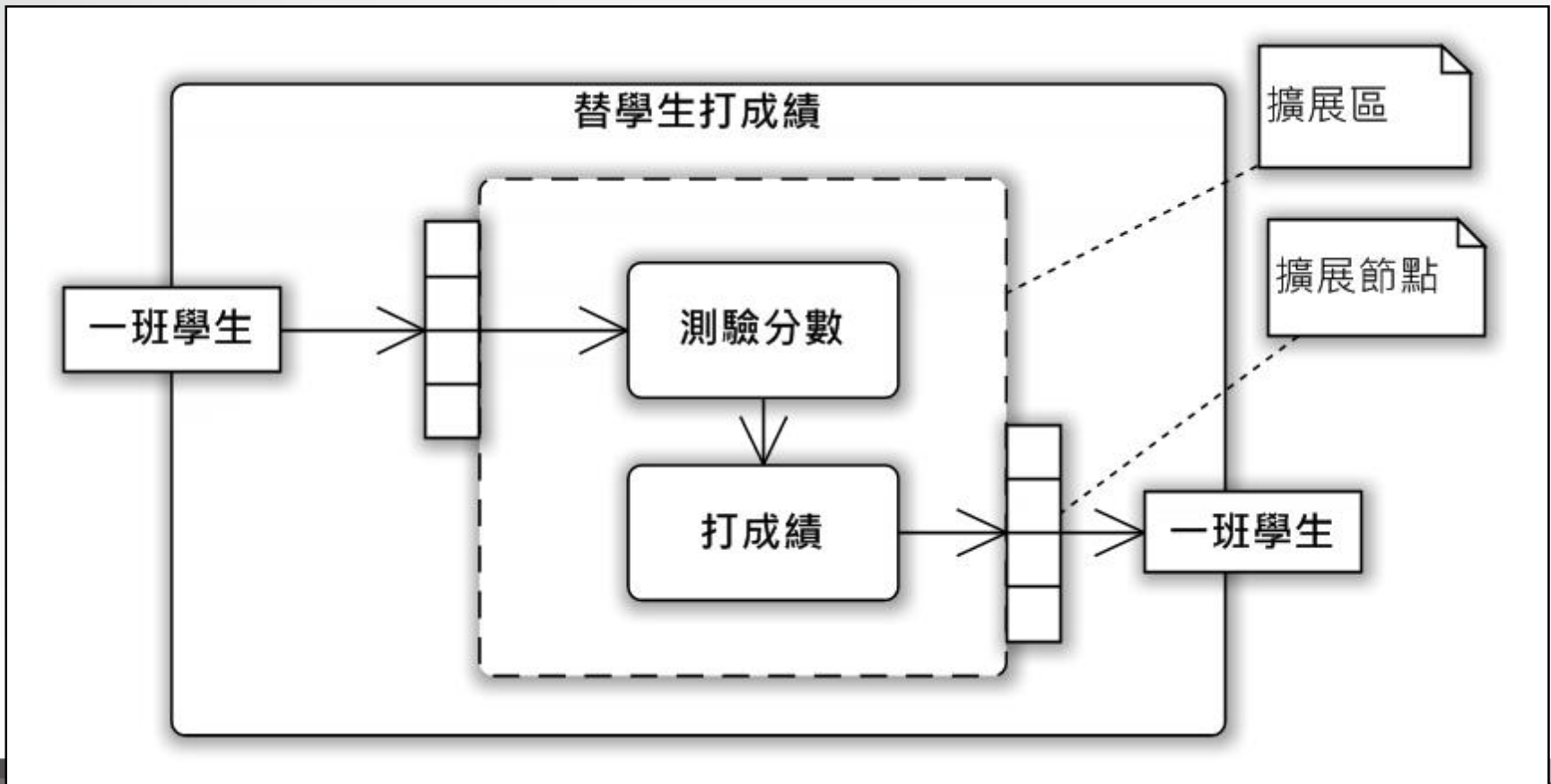


7-3-7 擴展區與擴展節點-說明

- 擴展節點（Expansion Node）可以明顯呈現擴展區（Expansion Region）是如何處理集合物件的每一個項目，例如：使用擴展區描述函數如何取得一組檔案清單，然後在每一個檔案清單中搜尋關鍵字。
- 擴展節點是一組物件集合，其符號類似一維陣列圖形的表格，可以表示集合物件是如何進入和離開擴展區，擴展區符號是一個虛線圓角長方形。

7-3-7 擴展區與擴展節點-範例

■ 替一班學生打成績的活動圖，如下圖所示：





7-4 活動圖的開始與結束

- 7-4-1 活動圖的開始
- 7-4-2 活動圖的結束

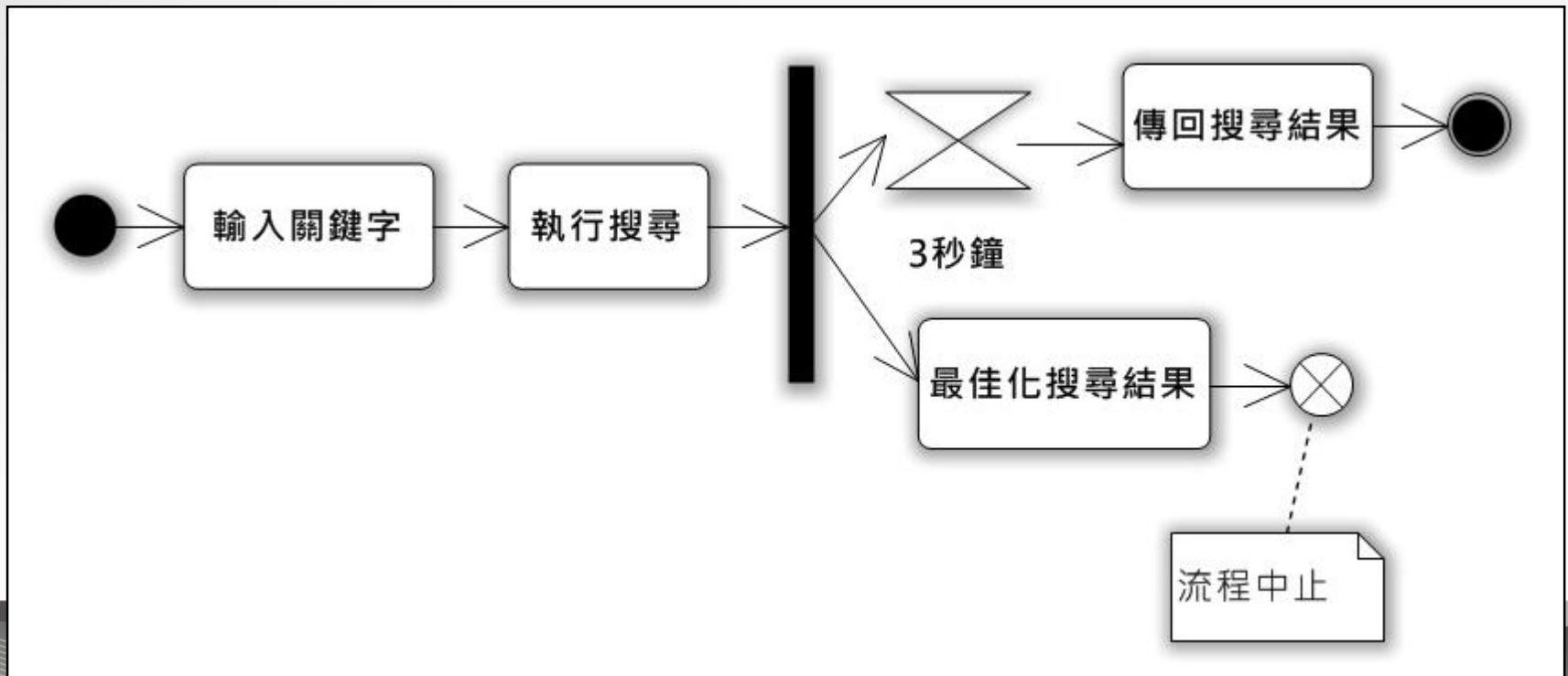


7-4-1 活動圖的開始

- 活動圖的開始是整個流程的開始，我們可以有多種方式來開始活動圖的流程，如下所示：
 - 活動圖最常用的方式是使用起始節點來開始。
 - 使用時間事件動作來周期起始活動圖，詳見第7-3-1節。
 - 使用接收訊號取代起始節點作為流程的開始，簡單的說，這是使用外部參與者的人員、系統或流程送出的訊號來喚醒活動，詳見第7-3-4節。
 - 使用活動參數節點來開始活動圖，詳見第7-3-3節。

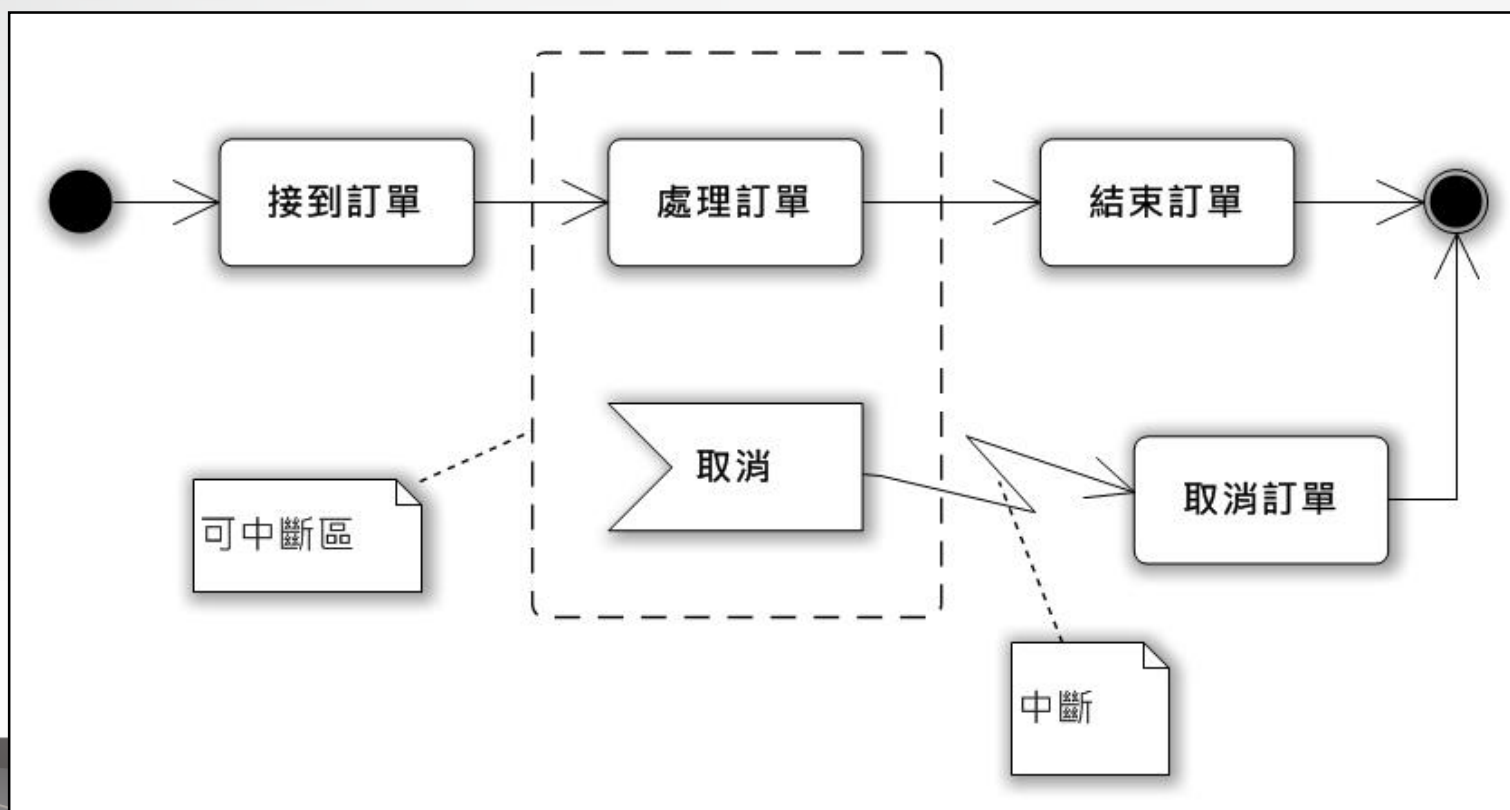
7-4-2 活動圖的結束-流程中止

- 流程中止（**Flow Final**）並不是結束整個活動，而是其中一條路徑，所以不會影響其他路徑的執行。UML是使用小圓形在中間加上大X符號來表示，例如：搜尋引擎執行搜尋，如下圖所示：



7-4-2 活動圖的結束-中斷活動

- 當活動圖的流程需要被事件中斷，通常是使用者自行取消操作，例如：取消訂單，在活動圖是在可中斷區（**Interruption Regions**）顯示中斷的連接線，如下圖所示：





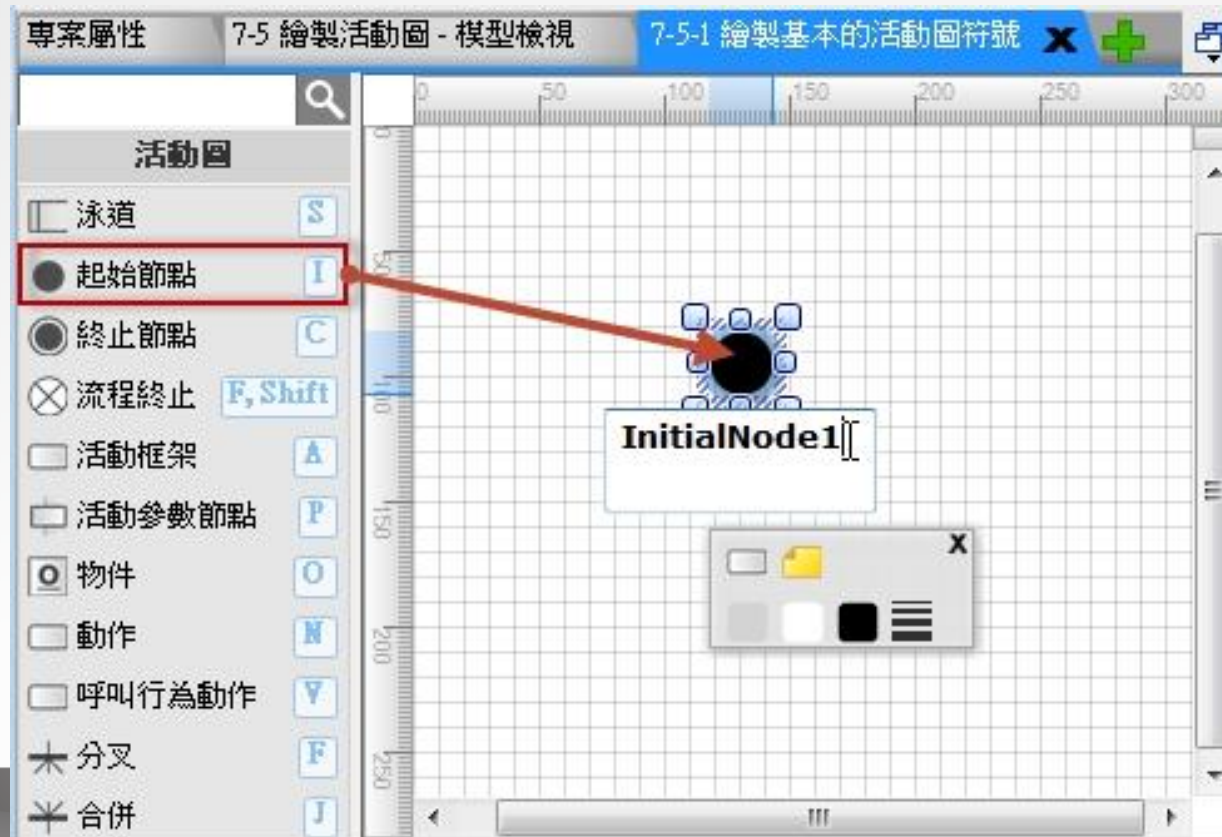
7-5 繪製活動圖

- 7-5-1 繪製基本的活動圖符號
- 7-5-2 繪製進階的活動圖符號



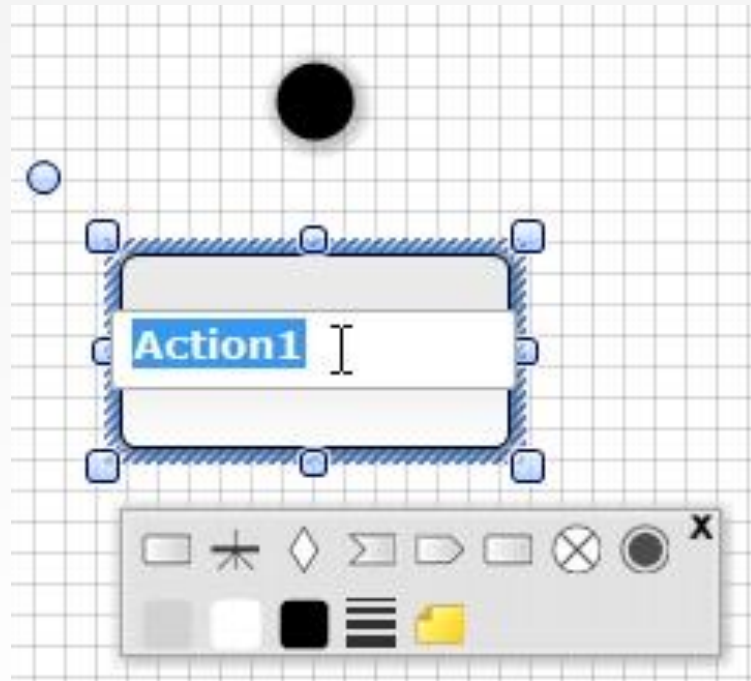
7-5-1 繪製基本的活動圖符號- 新增起始節點

- 在「工具箱」視窗拖拉【起始節點】至編輯區域後，即可新增起始節點符號，如下圖所示：



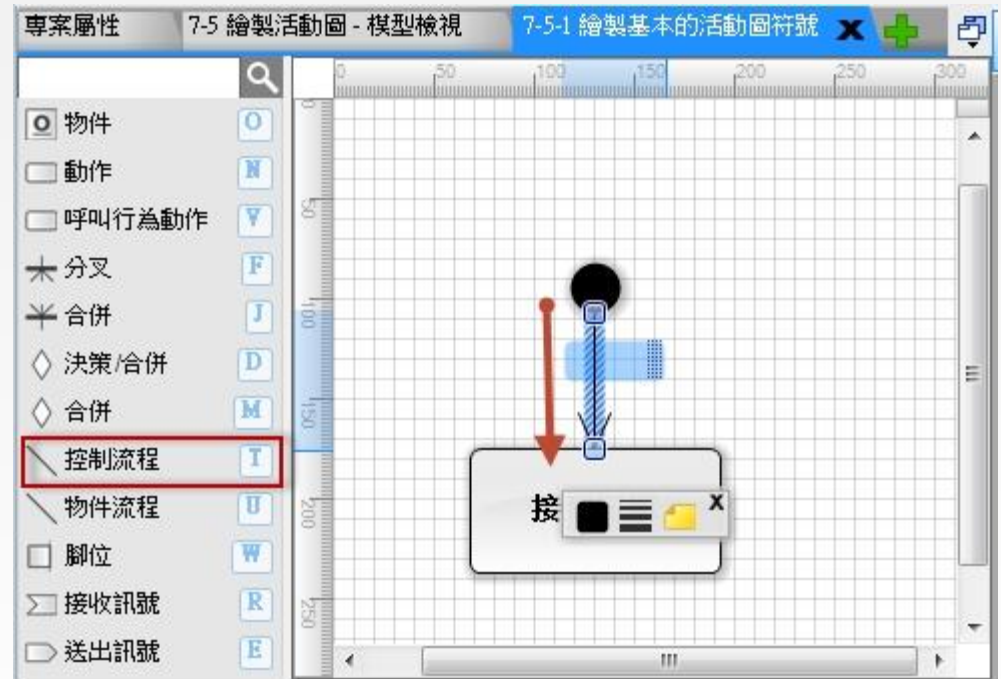
7-5-1 繪製基本的活動圖符號- 新增動作節點

- 同樣方式，從「工具箱」視窗拖拉【動作】，就可以在編輯區域新增動作節點符號，和輸入動作名稱，如下圖所示：



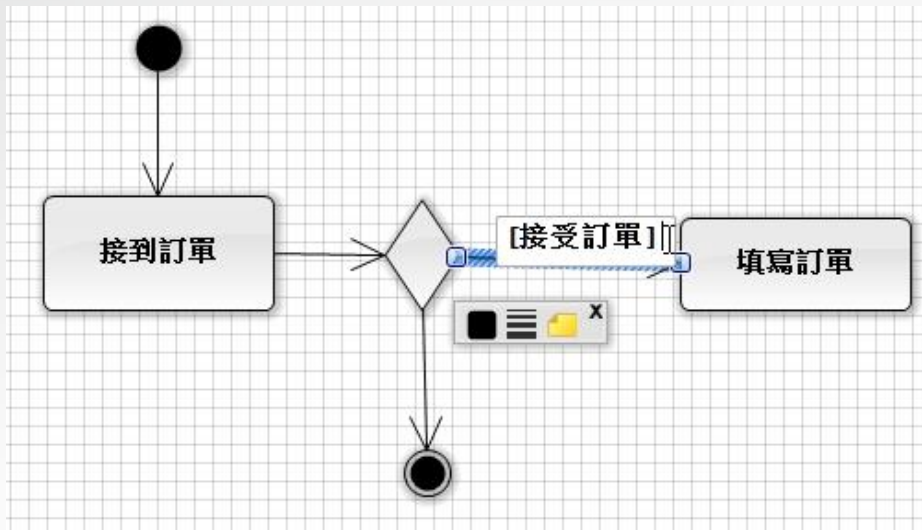
7-5-1 繪製基本的活動圖符號- 新增連接線

- 在活動圖的節點之間建立連接線是【控制流程】，請在「工具箱」視窗選【控制流程】，然後在來源節點（起始節點）上按一下，拖拉至目的節點（動作節點）上，放開滑鼠按鍵就可以建立連接線，如右圖所示：



7-5-1 繪製基本的活動圖符號- 在連接線上輸入防衛條件

- 在新增菱形決策節點（如果需要，可以縮小菱形尺寸）後，就可以在離開此節點的2條邊線上輸入防衛條件，如下圖所示：

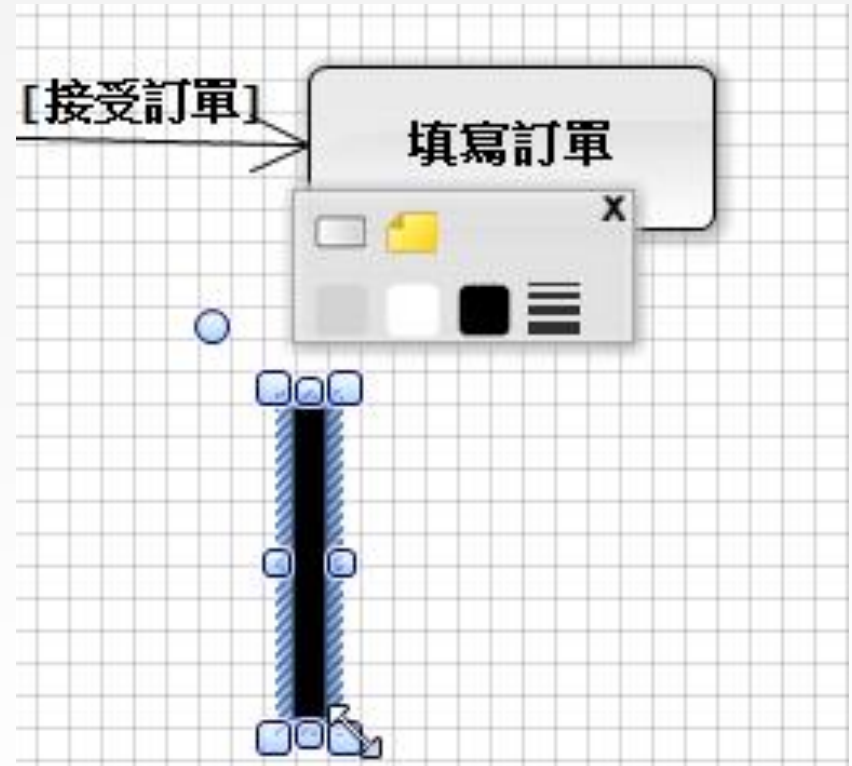
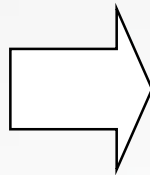
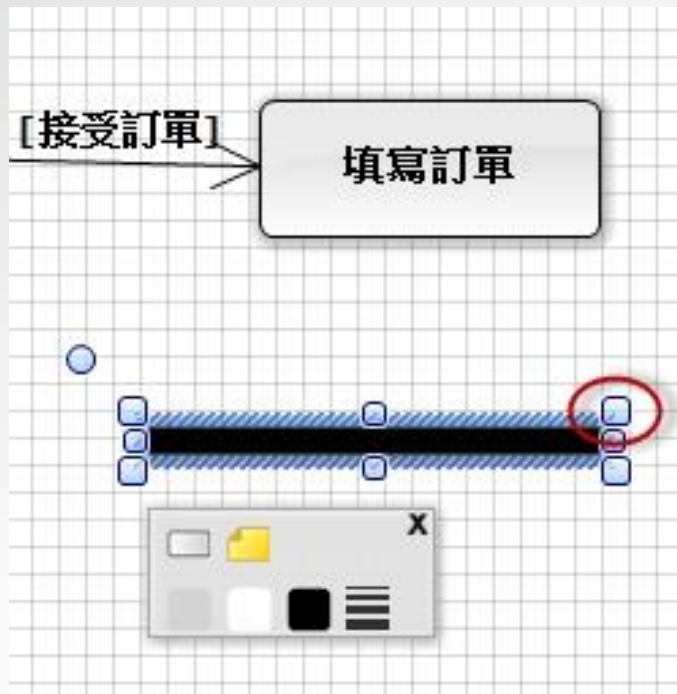


The image shows the "屬性" (Properties) window for a UML element, specifically a decision node. The window has tabs for "一般" (General), "控制流程" (Control Flow), "合作" (Collaboration), "結合圖" (Merge Diagram), "樣式" (Style), "文件" (File), and "標記值" (Marked Value). The "一般" tab is active. The "編號" (ID) field contains "ELEM006". The "名稱" (Name) field contains "[接受訂單]". The "能見度" (Visibility) dropdown is set to "package". Under the "修飾子" (Stereotypes) section, there are checkboxes for "Abstract", "Static", "Root", and "Leaf", all of which are unchecked. The "超連結(H)" (Hyperlink) section has a "URL:" field. The "模版(S)" (Template) section has a "套用模版(A):" (Apply Template) list and a "可用模版(V):" (Available Templates) list, with buttons for "<新增(A)" (Add), "<新增目訂(C)" (Add Custom), "移除(R)>" (Remove), and arrows. At the bottom right are "確定(O)" (OK) and "取消(C)" (Cancel) buttons.

7-5-1 繪製基本的活動圖符號-

將分岔與結合的同步棒改為垂直顯示

- SIM預設建立的同步棒（Synchronization Bar）是水平顯示，如下圖所示：



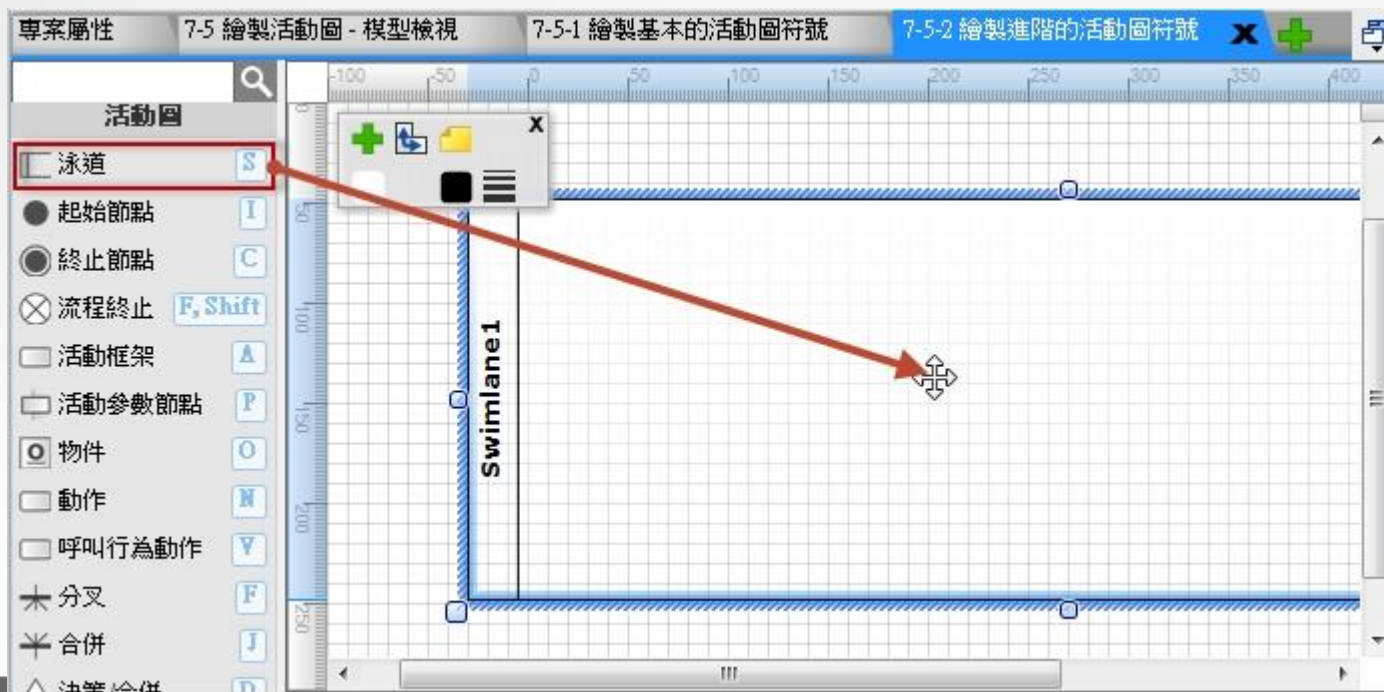
7-5-2 繪製進階的活動圖符號- 建立呼叫其他活動的動作節點

- 在動作節點呼叫其他活動，請在「工具箱」視窗選【呼叫行為動作】符號，就可以建立呼叫其他活動的動作節點，如下圖所示：



7-5-2 繪製進階的活動圖符號- 建立泳道

- SIM支援的泳道只能水平顯示，無法調整成垂直顯示，我們只需從「工具箱」視窗拖拉【泳道】就可以建立一道泳道，如下圖所示：



7-5-2 繪製進階的活動圖符號- 圖層的处理

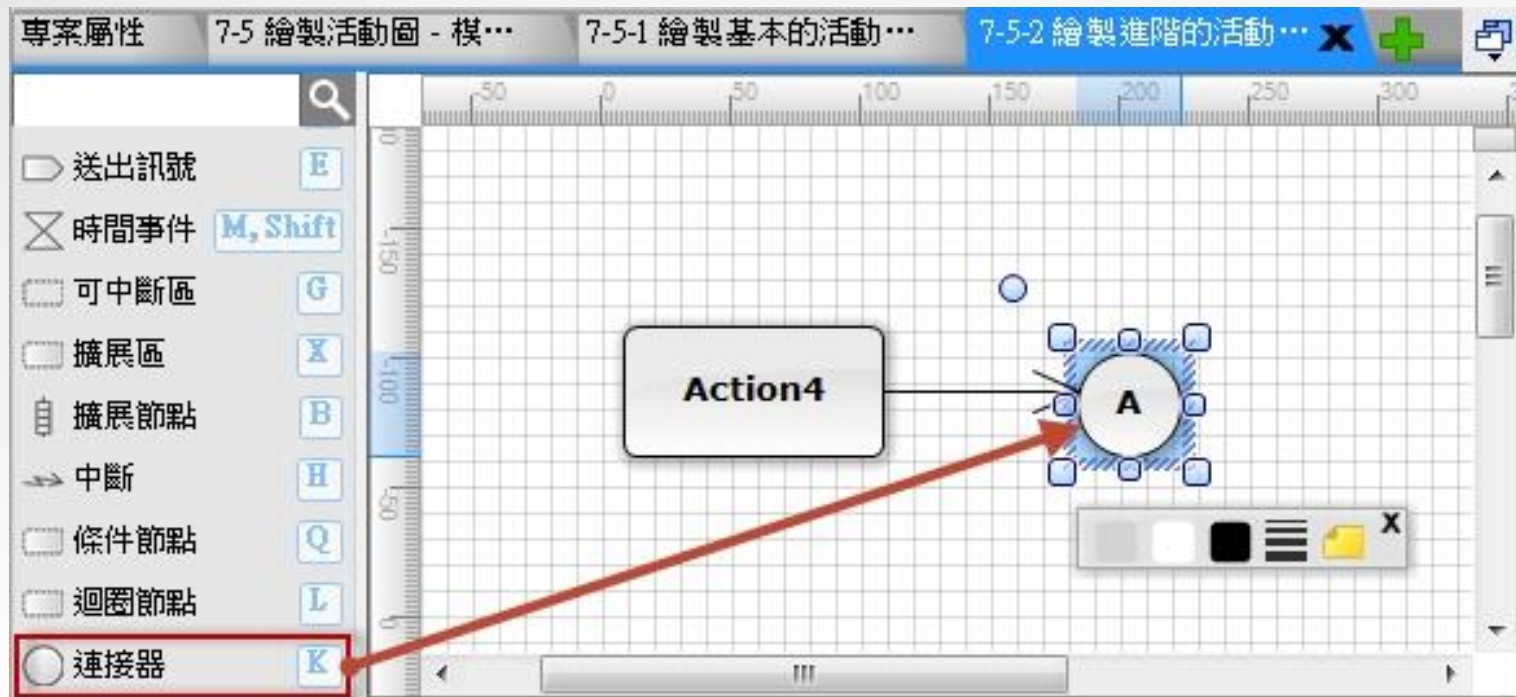
- 如果在新增其他符號後，才加上泳道、活動框架、可中斷區和擴充區等容器符號（建議先新增這些容器符號後，再在其中新增符號），此時就需要考量圖層問題，否則有可能無法建立節點之間的連接線。
- 因為SIM編輯區域新增的符號預設是位在舊符號的上一層圖層，我們可以執行「排列/順序/移至最下層」或「排列/順序/提到最上層」指令，將容器符號移至內容符號的下一層。

7-5-2 繪製進階的活動圖符號- 腳位與參數節點名稱

- 在第7-3-3節建立動作輸出與輸入物件的腳位時，**SIM**腳位名稱位置預設是在小長方形符號的上方，請自行拖拉調整其位置。同理，參數節點名稱是位在長方形符號的外面，也請自行拖拉調整至長方形之中。

7-5-2 繪製進階的活動圖符號- 新增連接器符號

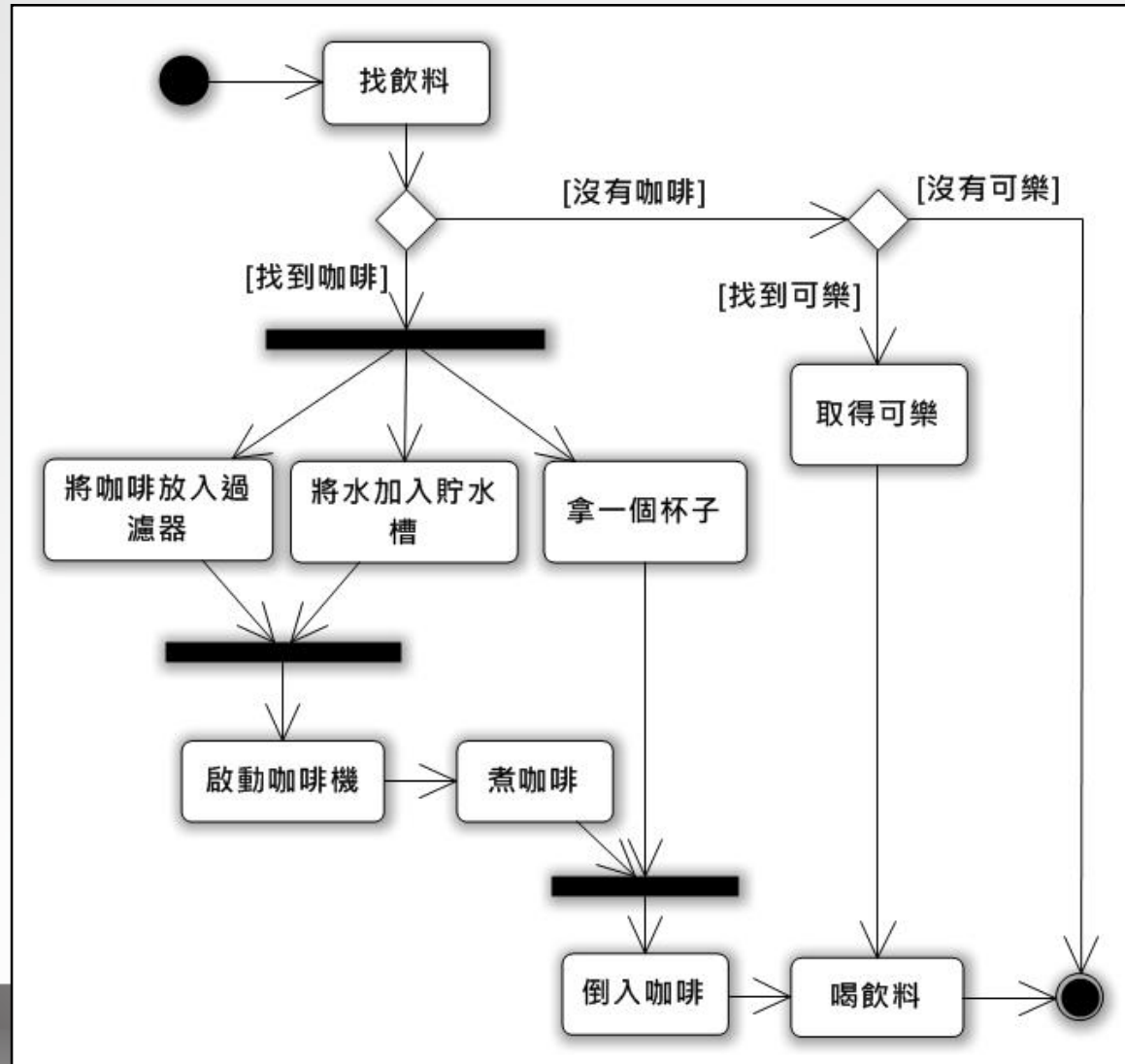
- 請在「工具箱」視窗拖拉【連接器】符號，就可以新增連接器符號和輸入名稱，如下圖所示：



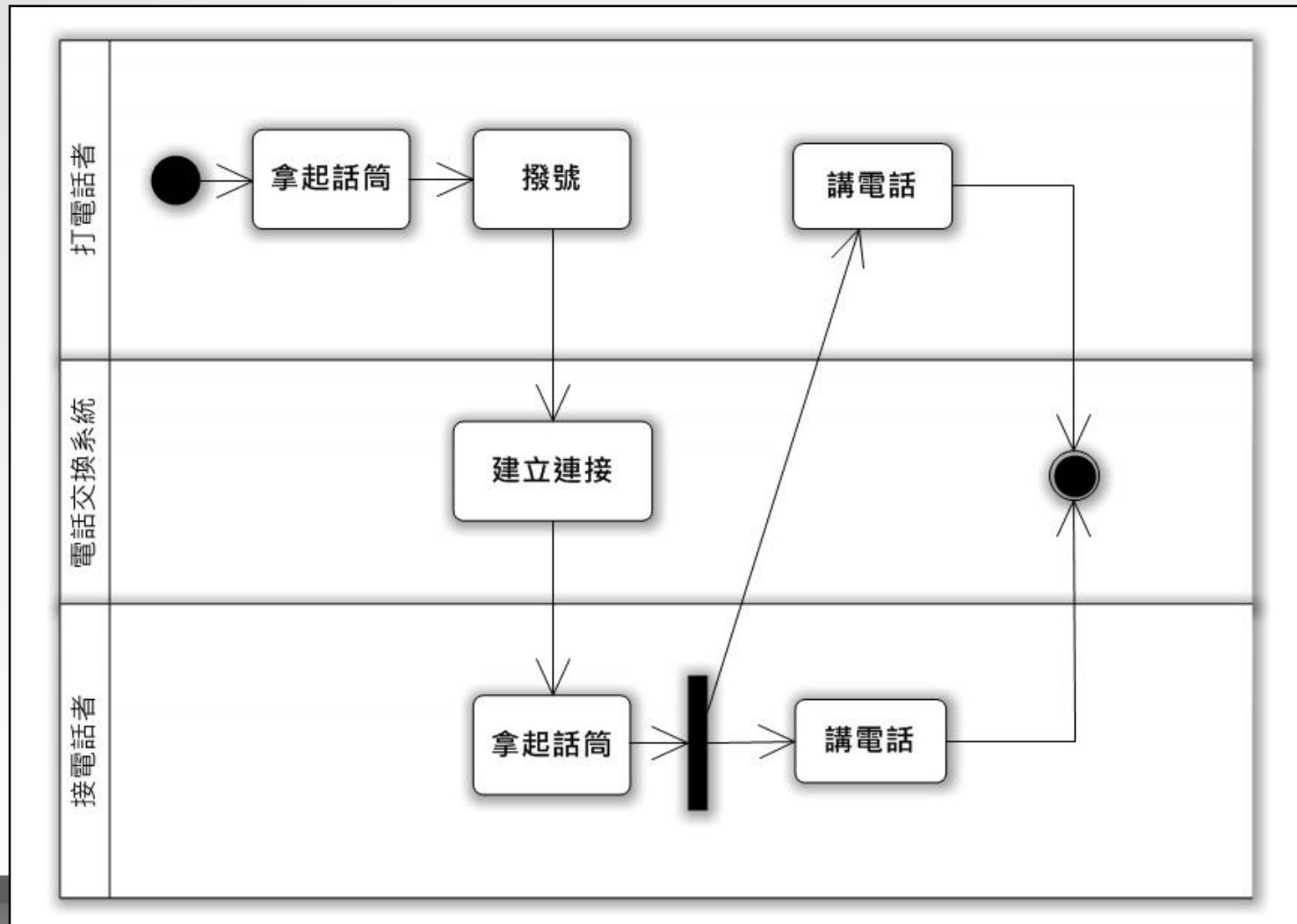


7-6 綜合練習-

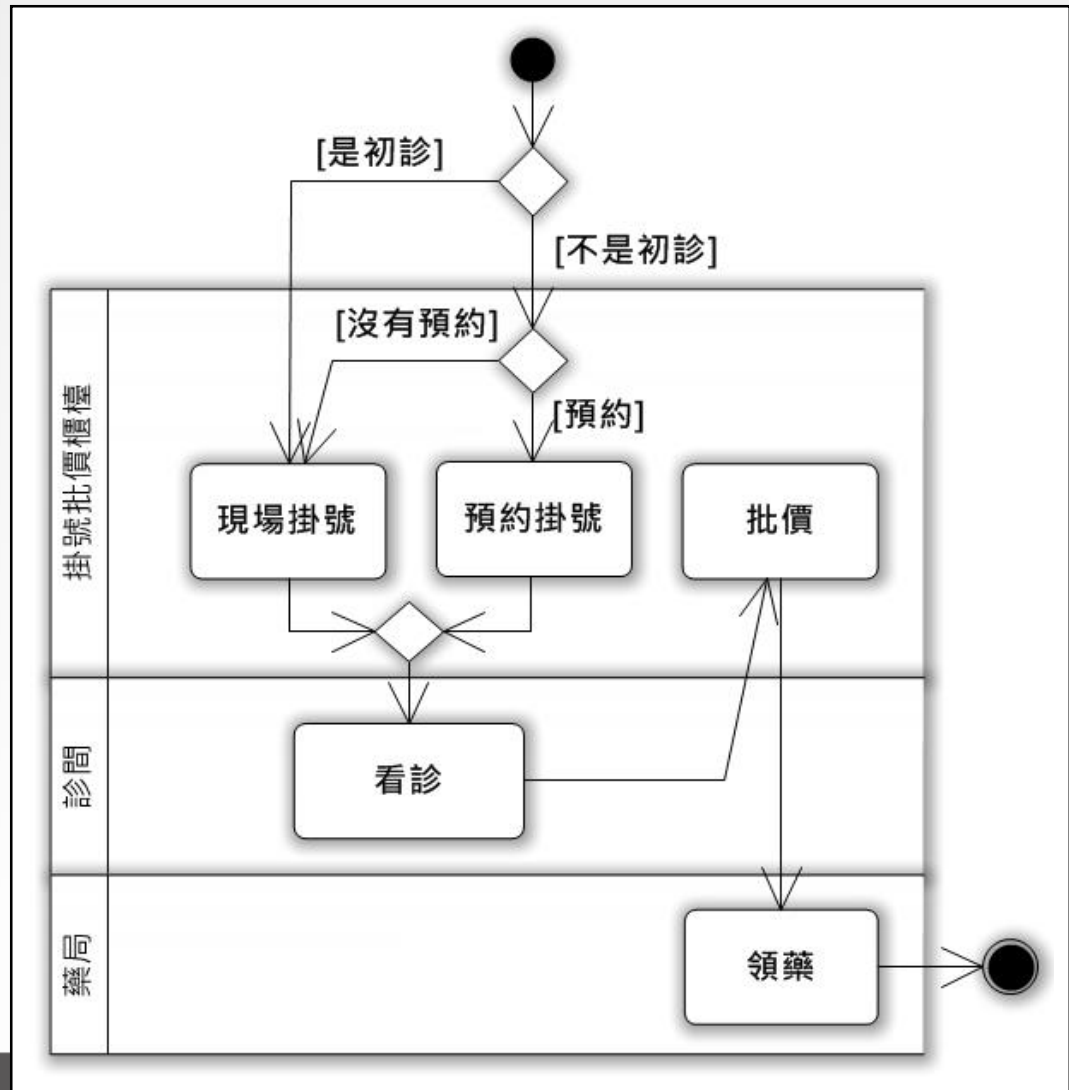
在公司休息室喝飲料的流程



7-6 綜合練習- 打室內電話的流程



7-6 綜合練習- 醫院掛號看診流程



End

