

104 年特種考試地方政府公務人員考試試題

等 別：三等考試

類 科：資訊處理

科 目：系統分析與設計

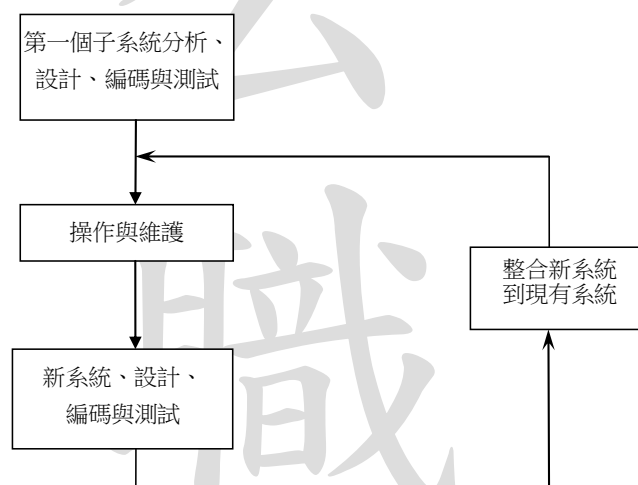
一、漸增模式 (Incremental Model) 是一種常見的軟體開發方法：

(一)何謂漸增模式 (請說明其執行程序與原則及強調重點為何)？

(二)請說明其與瀑布模式在應用上之差異？

【擬答】：

(一)漸增模式把需求分成部分，然後將每個「部分需求」開發訂為一個開發週期，每個週期可依序或平行開發。每個週期之階段清楚定義要做哪些工作及交付哪些文件，每個階段循序進行且僅循環一次。漸進模式是基於下列的構想而設計：將一個軟體系統分割成數個子系統，每個子系統執行一部分功能。設計一個彈性、開放的系統架構，使後續的功能能夠加入，而不需修改基本的架構。後續的子系統陸續與已完成的部分整合在一起。



(二)漸增模式與瀑布模式不同地方：

- 1.系統被分成幾個子系統或功能，各子系統可獨立依序開發；而瀑布模式則是各個子系統需同時開發。
- 2.系統開發可由多個週期完成，每個週期表示不同版本之系統，因為每個週期均有程式編輯及上線實施，使用者均有參與，故漸增模式之風險較低。

二、開發一套新系統，例如甲資訊公司承接乙公司的電子商務系統 (含訂購、付款、出貨等)，需經歷需求分析、系統分析與設計等步驟：

(一)請至少舉出常見的五種需求擷取方式。

(二)進行需求分析時，應以那些需求塑模工具、語言或圖形來表達那些不同性質的需求 (例如作業、規則、表單、資訊等)？

【擬答】：

(一)常見的五種需求擷取方式

1. 查閱文件

係指研究企業的內部文件，例如工作說明書、企業表單與手冊等，是瞭解企業運作邏輯之初步工作。以此方式蒐集之資訊常有過時之慮。

2. 觀察

所獲得資料之正確性會比查閱文件為高，亦能驗證所蒐集資料之正確性及補充不完整的部分，透過實地觀察也可獲得第一手資料。觀察時，可選擇正常與例外情況之時機或對象來做觀察，以便獲得各種可能的資料。但觀察仍無法完整地反映出組織的真實情況與需求，例如被觀察者的行為可能改變。

3. 訪談

是需求擷取方式中最有效且最普遍的資料蒐集方法。訪談期間，系統分析師蒐集到的可能是事實、選擇或推測，並可觀察到人們的肢體語言、情緒和他們對於現行系統之觀感。

4. 問卷

當潛在使用者太多或分布太廣時，可考慮以問卷之方式擷取需求。適合於大型企業或公共資訊系統的設計，因為問卷所涉及的作業範圍或對象太廣，系統分析師無法逐一親自調查，故利用問卷方式來蒐集使用者需求較為可行。因為問卷上的問題是以文字靜態地表達出來，因此問題之語意與邏輯必須很清楚且有條理。設計問卷時也可用各種不同的方法來問同一個問題，以觀察各種可能的答案。

5. 開會討論

是一種很有效率的資料蒐集方式。使用者代表與系統開發人員齊聚一堂，將所知道的事實、觀念說出，讓與會人員一起相互溝通意見。

(二)

1. 環境圖：主要表達系統與所在環境之關係。用來表達系統所在之環境及其與環境間之關係，這包括與系統有關之外部實體及系統與外部實體間之互動，例如資訊之輸出入與處理等。常用於表達系統之巨觀範圍，以幫助我們瞭解系統所在之環境及兩者間之互動關係，包括下列圖形：

(1) 系統

以圓形來表示系統，並於其中註明系統名稱。

(2) 外部實體

外部實體可以是任何組織、物件或相關系統，是環境中與系統有互動或交換訊息之任何人或物。以矩形表示外部實體，且於其中註明外部實體之名稱。

(3) 處理與資訊流

處理與資訊流是以箭頭連結系統與外部實體，表示資訊之輸出入處理或事件之方向。

2. 流程圖：主要表達實體之作業程序及所需資訊。為一圖形化表達工具，用以描述企業作業流程，其可分為系統流程圖及程式流程圖兩類。系統流程圖(System Flow Chart)用以描述整個工作系統中，各單位之間的作業關係。程式流程圖(Program Flow Chart)則用以表示程式中的處理過程。有下列元件：

(1) 作業處理(Processing)

為真實世界的一個動作處理、一組動作程序、或是可執行的一段副程式。一個作業處理是流程的基本單位，不能再被分解且此動作之執行不能被中斷。作業處理以矩形表示，內部標示流程名稱。

(2) 決策(Decision)

是用於表達有多個選擇路徑，但僅能依條件選擇其中一個路徑執行，條件通常包含 Yes/No 或 True/False 這類的問題。決策是以一菱形外加一條流入菱形之箭頭與多條流出菱形之箭頭表示。

(3) 報告(Report)

可用於描述某一個流程之輸出入資訊。

(4) 流程方向(Flow of Control)

用以表達控制流。

(5) 接點(Connector)

當流程圖過於龐大或複雜時，可使用接點(Connector)將流程圖轉到另一頁，此舉可避免流程方向之箭頭交叉，亦可避免流程太過冗長，增加流程圖之易讀性。

(6) 註解(Annotation)

用以表達流程圖之補充與說明。

3. 事件：表示外部實體所啟動且系統必須回應之刺激，可分為：

(1)資料流導向事件

是系統藉由接收到資料之輸入而知道事件已發生。例如系統收到客戶輸入代號時，啟動驗證是否為有效客戶之事件。

(2)時間導向事件

指預設之時間到時，該事件被啟動。例如在系統內建一時鐘，當時間到達時，系統便會自動啟動簽發支票事件

(3)控制導向事件

是由非預設時間之某些刺激或狀態所引發。例如系統之開或關等。

4.處理描述：表示流程圖中作業處理之執行程序與規則、相關之資料輸出入資訊以及處理之限制與備註。

5.藍圖：主要表示流程圖中資訊之展示格式與內容。用於表達流程圖中，有關之表單、介面等各項資訊需求之名稱、展示位置、格線、圖表與說明等。這些資訊常無法在流程圖上具體地表達，因此須另以藍圖做進一步地表示。

6.資料詞彙：主要描述藍圖內資訊之詳細內容與規則，如資訊之長度、型態、格式、公式、規則、範圍與限制等，並分別舉例說明之。資料詞彙中，規則、格式、範圍、公式等資訊可使用資料字典的格式來表達。

三、一個系統開發專案在不同階段會有不同性質之參與者參與，例如使用者、系統分析師、系統整合者，測試者、技師與專案管理者等，不同性質的參與者在專案發展過程中，會在不同的時間點從不同的角度來看系統，例如使用個案觀點(Use Case View)、設計觀點(Design View)、流程觀點(Process View)、實施觀點(Implementation View)與部署觀點(Deployment View)。以應用物件導向技術為例，在實施觀點與部署觀點，主要以元件圖與部署圖進行系統元件與結構的塑模。

(一)請分別說明元件圖與部署圖主要表達什麼？

(二)請說明元件圖與部署圖在應用上之差異？

【擬答】：

(一)

1. 元件圖(Component Diagram)用以說明系統設計過程各類別與物件的配置，以及敘述軟體元件間的組織架構和相依關係。元件是開發和執行過程之實際物件類別，將可分解的實際基本單位模組化，這些基本單位包括模組並擁有特性和明確定義的介面。包含下列元件：

(1)元件(Component)是系統的實體模組，也是系統中可被替換的部分，適合提供給一群介面做適當的實現(實作)。

(2)介面(Interface)是許多操作(不包含實作)的集合，用來說明類別或元件所提供的服務，而其他類別或元件可以透過介面來實作這個類別或元件。

(3)關係是描述元件與元件，或元件與介面間的結合關係，關係之意義與表達符號均與類別圖相同。

2. 部署圖(Deployment Diagram)用來說明系統各軟硬體(例如處理器、處理元件)元件的配置與關聯，或是硬體及網路此類基礎建設之拓撲，例如系統是由哪些硬體所組成，硬體上面安裝了哪些軟體，而硬體間使用了哪些中介軟體來做連結。包含下列元件：

(1)產出(Artifacts)

代表現實生活中具體的資訊，為軟體使用或執行的實體檔案。例如原始碼檔(.java 或.cpp 檔案)、可執行檔(.exe 或.jar 檔案)、程式庫檔(.dlls 檔案)等。其表達方式為一矩形，其右上表有一折角之文件圖示，並在其名稱上方標示構造型 <<artifact>>。

(2)節點 (Nodes)

代表著一種計算資源，也可說是一個硬體。例如，可以是一個單純的設備或感應器，也可能是一部大型主機等。因此，一台 Web Server、一組硬碟陣列、一台電腦或一台印表機等皆可設為一個節點。

(3)連接(Connections)

表示節點、元件或產出間的溝通方式或路徑，也表示其間之關係。

(二)系統元件與結構塑模主要是以元件圖視覺化地表達系統中實體元件間的靜態結構關係，以及說明其建構細節。系統元件與結構塑模主要是以部署圖視覺化的表達系統中，一群實體節點與實體節點間的靜態結構關係及說明其建構細節。

四、資料庫若沒有適當的設計，於資料維護時可能造成異常現象，包括新增異常、刪除異常及修改異常。請分別舉例說明這三種異常。？

【擬答】：

例如下圖的資料庫設計及果中，因為FD 集合的左邊中有主鍵(S#,P#)的子集合，如S#→CITY，因此此一關連式並非 2NF。不符合 2NF 的關連式會造成下列異常現象：

FIRST

S#	P#	QTY	CITY	STATUS
		↑	↑	↑

(一)新增異常：輸入時必須等主鍵完整輸入才可進行，例如 FIRST 中我們不能單單描述一個供應商 S9 住在 HK，一直要到這個供應商開始供應某一零件才能將上列資料輸入。(因為實體整合規則)

(二)刪除異常：刪除時會把過多資訊刪掉，例如 FIRST 中我們將一個供應商 S9 供應零件 P9 的值組刪除時，我們同時也刪掉 S9 住在 HK 的資訊。

(三)修改異常：修改時需要一起修改許多相關值組，例如 FIRST 中我們將一個供應商 S9 住在 HK 改為住在 Taipei，則需要將其中許多 S#=S9 的值組一起修改，否則資料會不一致。