

《資訊系統與分析》

試題評析

今年資訊系統與分析的考題都是觀念性的問題，需要觀念很正確才能答的好。第一題考傳統的檔案與資料庫之優缺點比較，第二題是問物件導向技術中的繼承機制之優缺點，第三題考的是在系統分析與設計中建立模型之目的，第四題問的是系統的功能性與非功能性需求，第五題則是考逐步精緻化及其在結構化系統分析之應用。綜合而言，今年的考題對於普通的考生約可得40 55分，程度較好者可望拿55 70分。

一、所有資訊系統皆將產生並保養以及使用資料，這些資料以檔案（file）或資料庫（database）的型態儲存，試比較這兩種儲存資料的方式有何優缺點，並舉例說明。（20分）

答：

(一)使用檔案之優點：

- 1.作業成本較低。
- 2.意外狀況之復還較容易。
- 3.資訊系統開發人員有較高的自主能力。
- 4.不必培訓資料庫管理師。

(二)使用檔之缺點：

- 1.資料重複性高。
- 2.資料完整性難維持。
- 3.資料共用性差。
- 4.難以建立標準。
- 5.程式生產力低。
- 6.程式維護成本高。
- 7.資料管制困難。

(三)使用資料庫之優點：

- 1.減少資料的重複。
- 2.避免資料不一致。
- 3.資料的整合性高。
- 4.資料的共同性佳。
- 5.資料的獨立性強。
- 6.利於應用程式發展。
- 7.易建立資料的安全機制及作業之標準化。

(四)使用資料庫之缺點：

- 1.意外狀況易造成無法彌補的傷害。
- 2.優秀的資料庫管理師（DBA）難以培訓。
- 3.很難設計出具有完善整合控制能力的資料庫管理系統。
- 4.作業成本提高。

二、在物件導向技術中繼承(inheritance)是最重要的機制之一，試論這項機制的優缺點，並舉例說明。（20分）

答：

(一)繼承是一種超類別和子類別之間的關係，例如員工類別中包含有姓名、電話、地址等屬性及工作、吃飯、休息等方法（運作），可定義已婚員工為員工之子類別（員工則為已婚員工之超類別），此時已婚員工除可繼承員工的屬性和方法外，亦可加入新的屬性（如配偶）和新的方法（如養育子女）。

(二)繼承的優點：

- 1.程式碼可重複使用，減少重新撰寫程式之工作。

- 2.降低製作時之複雜性。
- 3.增加軟體之可靠性。
- 4.降低成本。
- 5.提高軟體發展效率。
- 6.提高系統品質。
- 7.減少軟體製作之風險。
- 8.減少撰寫程式及測試之時間。

(三)繼承的缺點：

- 1.不當或過度使用繼承，易增加程式及測試之複雜度。
- 2.不當使用繼承將造成不穩定且不易維護之類別階層。
- 3.使用多重繼承時，若無良好規劃設計，易導致整體架構雜亂且執行缺乏效率。

三、系統分析師以模式(model)來描述從真實世界中所蒐集的資料，例如結構化技術中的資料流程圖(dataflow diagram)模式，或如物件導向技術中顯示系統功能的使用個案模式(use case model)等，為何系統分析與設計者要以模式來描述系統的需求？(20分)

答：

所謂建立模型(modeling)就是產生一個能夠表達觀念及流程的圖型使得系統開發人員能夠分析、測試以及修改。系統分析師藉由一套對企業、資料、物件、網路及流程的模型就能夠描繪並簡化一個資訊系統。所謂企業模型(business model)或稱為需求模型(requirement model)被用來描述一個資訊系統所必須支援的企業功能。而所謂的資料模型(data model)則描述資料的結構與設計。物件模型(object model)描述結合了資料與流程的物件。網路模型(network model)則描述了通訊連結的設計及協定。處理工作模型(process model)則描述了程式設計師用來發展程式碼模組的系統邏輯與處理方式。雖然這些模型在某些層面似乎有重疊，實際上他們只是同時從不同的觀點來描述同一個環境。

建立模型需要許多不同的技術，如資料流程圖(DFD, data flow diagrams)、實體關聯圖(ERD, entity relationship diagrams)、使用個案(use cases)，以及統一模型語言(UML, unified modeling language)。

四、需求(requirements)敘述系統以及其與四周環境，諸如使用者與其他系統之間的交互作用(interaction)，系統分析師蒐集需求時主要蒐集系統的功能需求(functional requirements)與非功能需求(nonfunctional requirements)，何謂功能與非功能需求？試舉例說明。(20分)

答：

(一)功能性需求包括：

- 1.系統功能(system function) - 必須由某些輸入資料製作出某些特定輸出資料之功能。
- 2.外部介面(External interface) - 系統軟體必須與外部硬體、資料庫或其他軟體系統銜接之介面限制。
- 3.設計限制(Design constraints) - 根據系統與應用領域之特性，在設計階段必須滿足之特殊限制條件。

(二)非功能性需求包括：

- 1.績效(Performance) - 系統在執行狀態中必須滿足之績效條件，如執行速率、容量、精確度等。
- 2.品質屬性(Quality attributes) - 系統必須滿足之其他可測量之品質，如可靠性、可維護性、正確性等等。
- 3.用戶介面(User interface) - 系統用戶介面與使用者溝通時必須具備之需求，此友善性、簡單易學等。

五、「逐步精緻化」(stepwise refinement)是解決問題(problem-solving)的技術基礎，尤其從事系統分析時，這項技術的運用更為重要，試解釋何謂逐步精緻化，並論述其在DeMarco或Gane/Sarsen的結構化系統分析(structured system analysis)時的應用。(20分)

答：

(一)逐步精緻化可重覆地將較抽象或較高層次的概念細分成一些較具體且較低階層的想法，它是整個自上而下設計過程中所採用的不二法則。逐步精緻化可由下面幾個步驟所組成的：

- 1.確定此設計工作必須作的決策是什麼。
- 2.將這些設計決策按重要性排列優先等級。
- 3.從事結構化的設計。
 - (1)先決定重要的決策，愈細微的決策拖到愈晚決定愈好。
 - (2)作劃分工作。每一次的精緻化過程都應該把問題劃分成幾個獨立的子問題，這些子問題在程式中都應自成一個段

落。

(3)在選擇每一階層的決策架構時，儘量簡化這個架構所能決定的事項，以便為下面階段的決策設計保留最大的選擇彈性。

(二)逐步精緻化在結構化系統分析之應用：

- 1.進行資料流程圖之階層化，其可避免因系統太大或太複雜而無法在一張紙上描繪出來。因此資料流程圖需要階層化，使之具有容易閱讀和管理之優點。
- 2.階層化是將系統加以細分之過程，細分到無法或無需再分的基本功能為止，即形成一組階層化的資料流程圖，從最簡要的到最詳細的資料流程圖都包括在內。因此一組階層化的資料流程圖由上往下展開後，即成為一張大的資料流程圖。