

專題

學生管理系統

- **涵蓋的概念:**
 - 類和對象
 - 屬性和方法
 - 建構
 - 數據類型和變量
 - 有條件的敘述
 - 數組和集合（列表,地圖）
 - 繼承和多形
 - 例外處理
- **項目描述:** 創建一個系統來管理學生信息。該系統應允許用戶添加,更新和刪除學生記錄,包括名稱,ID,年齡和成績等屬性。使用 Student 具有適當屬性和方法的類。實施不同類型的學生（例如本科生,研究生）以展示繼承和多態性。使用列表存儲學生記錄和地圖,按學生 ID 進行快速查找。處理重複 ID 或無效數據條目等情況的例外情況。

銀行管理系統

- **涵蓋的概念:**
 - 類和對象
 - 屬性和方法
 - 建構
 - 數據類型和變量
 - 有條件的敘述
 - 數組和集合（列表,地圖）
 - 繼承和多形
 - 例外處理

項目描述: 創建一個系統來管理銀行帳戶信息。該系統應允許用戶添加、更新和刪除帳戶記錄，包括帳戶號碼、持有者姓名、餘額等屬性。使用 BankAccount 類

來定義適當的屬性和方法。實施不同類型的帳戶（例如儲蓄帳戶、支票帳戶）來展示繼承和多態性。系統使用 List 來存儲所有帳戶記錄，並使用 Map 以帳戶號碼進行快速查找。處理重複帳戶號碼或無效數據輸入的異常情況。

這個系統將展示物件導向的核心概念，包括類別與物件、繼承、多態性、集合操作和異常處理。

屬性與方法:

EX:

- **class BankAccount**：定義一個代表銀行帳戶的類別，包括帳戶號碼、帳戶持有者姓名和餘額三個屬性。
- **private**：這些屬性是私有的，只能在該類內部直接訪問，保護資料不被外部直接修改。
- **建構子 (public BankAccount)**：當創建一個 BankAccount 物件時，會將帳戶號碼、帳戶持有者姓名和初始餘額賦值給物件的屬性。
- **getAccountNumber()、setAccountNumber()**：這些方法用來獲取和修改帳戶號碼。
- **getAccountHolderName()、setAccountHolderName()**：這些方法用來獲取和修改帳戶持有者姓名。
- **getBalance()**：此方法用來獲取當前帳戶的餘額。
- **deposit(double amount)**：此方法用來進行存款操作，當存入的金額大於 0 時，將該金額加到帳戶餘額中。
- **withdraw(double amount)**：此方法用來進行提款操作，當提款金額大於 0 且不超過帳戶餘額時，從帳戶餘額中扣除相應金額。如果不符合條件，則會拋出異常。
- **toString()**：此方法覆寫了 Object 類中的 toString 方法，返回該銀行帳戶的詳細資訊，包括帳戶號碼、帳戶持有人姓名和餘額。

- 使用 List、Map、ArrayList 和 HashMap 等類別，用來儲存和管理帳戶。
- 主類別，用來管理所有的銀行帳戶。
- 屬性：
 - accounts：這是一個 List，用來存放所有的 BankAccount 物件。
 - accountMap：這是一個 Map，將帳戶號碼(String)與對應的 BankAccount 物件進行映射，方便快速查找。
- addAccount 方法：
 - 用來添加新帳戶。如果帳戶號碼已經存在，則拋出異常。
- removeAccount 方法：
 - 根據帳戶號碼移除帳戶。如果帳戶不存在，則拋出異常。
- getAccountByNumber 方法：
 - 根據帳戶號碼查找帳戶。如果找不到對應的帳戶，則拋出異常。
- displayAllAccounts 方法：
 - 用來顯示目前系統中所有的帳戶。
- main 方法：
 - 程式的執行入口。這裡初始化系統，創建一些樣本帳戶（儲蓄帳戶和支票帳戶），將它們添加到系統中，執行存取款操作，並顯示帳戶資訊。