



國立臺北科技大學

進階 C 語言實務

Homework 2

老師：蔣政諺

班級：電機碩一

學號：111318133

姓名：魏千竣

日期：112/04/21

1. Program Description

1. 匯入所需的標頭檔和使用命名空間 std。

```
#include <iostream>
#include <cstring>
#include <vector>

using namespace std;
```

2. 定義一個 static 的成員變數 Count，用於計算建立的帳戶數量。

```
static int Count = 0;
```

3. 定義一個類別 Account。這個類別包含私有成員變數，只能在該類別被存取，包括帳戶名稱 name 和帳戶餘額 balance。"char name[100]" 是一個字符陣列，用來儲存使用者的名稱，最大長度為 100 個字符。

```
class Account {
private:
    char name[100];
    int balance;
```

4. Account 類別還包含一些公有成員函數，用於存款、提款、取得帳戶名稱、取得帳戶餘額及取得帳戶數量等操作。

- 首先建立預設建構子，用於初始化一個新物件的數據成員，設置所有帳戶的初始餘額為 0，並將帳戶名稱設置為空字串。及輸入建構子，來接受兩個參數：初始餘額、帳戶名稱用於賦予一個新物件的數據成員特定的值。

```
public:
    // 預設建構子
    Account() {
        balance = 0;
        name[0] = '\0';
        Count++;
    }

    // 輸入建構子
    Account(int initialBalance, const char* accName) {
        balance = initialBalance;
        strcpy(name, accName);
        Count++;
        cout << Count << endl;
    }
```

- 建立存款(Deposit) 將傳入的 amount 加到 balance 帳戶餘額中。及提款(Withdraw) 判斷帳戶餘額是否足夠提款，足夠就將傳入的 amount 從 balance 中減去，若帳戶餘額不足，則輸出 "Insufficient balance!"。

```
// 存款函數
void Deposit(int amount) {
    balance += amount;
}

// 提款函數
void Withdraw(int amount) {
    if (balance >= amount) {
        balance -= amount;
    } else {
        cout << "Insufficient balance!" << endl;
    }
}
```

- 再來建立取得帳戶(GetName)及取得餘額(GetBalance)來取得 Account 物件的兩個成員變數使用。兩個函數都是 const member function，這表示這些函數不會修改物件的狀態，因此在函數的參數列表最後面加上 const 關鍵字，避免對物件的修改。

```
// 取得使用者名稱
const char* GetName() const {
    return name;
}

// 取得帳戶餘額
int GetBalance() const {
    return balance;
}
```

- 接著建立取得帳戶數量(GetCount)的靜態成員函數，使其傳回儲存的帳戶數量。及最後的解構子，由於 Account 類別沒有需要動態分配的資源，所以在解構時是一個空函數。

```
// 取得帳戶數量
static int GetCount() {
    return Count;
}

// 解構子
~Account() {
}

};
```

5.主程式:

- 首先使用 vector 類型的 accounts 變數來儲存所有帳戶。及定義 name。

```
int main() {
    // 儲存所有帳戶
    vector<Account> accounts;
    char name[100];
```

- 使用 while 循環來提示使用者輸入名稱和初始餘額，創建一個 Account 對象，並利用 push_back 函式，將對象添加到 accounts 向量中，且中間有判斷，當輸入為"quit"時，要跳出這個迴圈。

```

while (true) {
    cout << "Enter your name (or enter 'quit' to exit): ";
    cin >> name;

    if (strcmp(name, "quit") == 0) {
        break;
    }

    int balance;
    cout << "Enter initial balance: ";
    cin >> balance;

    Account acc(balance, name);
    accounts.push_back(acc);
}

```

- 接著使用 while 迴圈，提示使用者選擇要執行的操作，包括存款、提款和退出。在這個循環中，使用 if 判斷來根據使用者的選擇，執行對應的操作。

```

while (true) {
    cout << "\n";
    cout << "1. Deposit\n";
    cout << "2. Withdraw\n";
    cout << "3. Quit\n";
    cout << "Choose an operation: ";
    int choice;
    cin >> choice;

    if (choice == 1) {
        int amount;
        cout << "Enter deposit amount: ";
        cin >> amount;
        acc.Deposit(amount);
    } else if (choice == 2) {
        int amount;
        cout << "Enter withdrawal amount: ";
        cin >> amount;
        acc.Withdraw(amount);
    } else if (choice == 3) {
        break;
    } else {
        cout << "Invalid choice. Please choose again." << endl;
    }
}

```

- 當輸入為 3 時，會跳出迴圈，並利用 "GetName" 及 "GetBalance" 印出這個帳戶的名字及存款。當最後所有帳戶都輸入完後，會利用 Account 裡的 GetCount 印出總共有幾個帳戶在這個 ATM 系統裡。

```

    cout << acc.GetName() << "'s balance is " << acc.GetBalance() << endl;
}

cout << "There are " << Account::GetCount() << " accounts in this ATM." << endl;
cout << "-----" << endl;

```

- 最後利用 for 迴圈，對於 accounts 中的每個元素，用變數 acc 代表該元素，並執行後面的程式碼，使用 "acc.GetName"、"acc.GetBalance" 依序取得所有帳戶，並印出帳戶名稱及金額。

```

for (const auto& acc : accounts) {
    cout << acc.GetName() << "'s balance is " << acc.GetBalance() << endl;
}

return 0;
}

```

2. Result Display

```
PS C:\Users\503503\Desktop\AdvC> ./hw2.exe
Enter your name (or enter 'quit' to exit): aaaa
Enter initial balance: 9999
1
1. Deposit
2. Withdraw
3. Quit
Choose an operation: 1
Enter deposit amount: 1

1. Deposit
2. Withdraw
3. Quit
Choose an operation: 2
Enter withdrawal amount: 500

1. Deposit
2. Withdraw
3. Quit
Choose an operation: 3
aaaa's balance is 9500
Enter your name (or enter 'quit' to exit): bbbb
Enter initial balance: 1234
2
1. Deposit
2. Withdraw
3. Quit
Choose an operation: 2
Enter withdrawal amount: 5555
Insufficient balance!

1. Deposit
2. Withdraw
3. Quit
Choose an operation: 5
Invalid choice. Please choose again.

1. Deposit
2. Withdraw
3. Quit
Choose an operation: 3
bbbb's balance is 1234
Enter your name (or enter 'quit' to exit): quit
There are 2 accounts in this ATM.
-----
aaaa's balance is 9999
bbbb's balance is 1234
```