

ASSIGNMENT 1

SHORT REPORT

1.EXPLANATION OF CLASSES DEFINED IN THE PROGRAM

1.1 CLASS OF ACCOUNT

The Account class is defined in account.php file. The main purposes of defining this class are:

- Record the ID and the balance of each account. IDs and balances are set to be properties of Account class. In order to gain good encapsulation, they are defined as private members.
- Setter and getter (class methods) are defined to access private properties.
- Define several methods to do account operation in terms of transaction validation, withdrawing and depositing.

```
//method of transaction processing
public function transactionValidation($transactionType, $transactionAmount){
    // if transaction is succeed, return true
    if($transactionType === 'W'){
        return $this->withdraw($transactionAmount);
    }else if($transactionType === 'D'){
        return $this->deposit($transactionAmount);
    }else{
        return false;
    }
}
```

transactionValidation method is defined to validate transaction.

Transaction type and transaction amount are passed in as arguments. If transaction is invalid it returns false.

```
//method of depositing money
public function deposit($transactionAmount){
    if($transactionAmount<0){
        return false;
    }else {
        $this->balance+=$transactionAmount;
        //echo 'account '.$this->id.' deposit amount:'.$transactionAmount.' balance:'.
        return true;
    }
}
```

deposit method is defined to validate transaction amount and deposit money into account.

```
//method of withdrawing money
public function withdraw($transactionAmount){
    if($transactionAmount<0 || $transactionAmount>($this->balance)){
        return false;
    } else {
        $this->balance-=$transactionAmount;
        //echo 'account '.$this->id.' withdraw amount: '.$transactionAmount.' balance:
        return true;
    }
}
```

Withdraw method is defined to validate transaction amount and withdraw money from account.

- there is no obvious limitation of these properties and methods

1.2 CLASS OF TRANSACTION

The Transaction class is defined in a1.php file. The main purposes of defining this class are:

- Record the line in the file, validation information, the account ID, the transaction type and the transaction amount of each transaction. All of these stuffs are set to be properties of Transaction class. In order to gain good encapsulation, they are defined as private members.
- Setter and getter (class methods) are defined to access private properties.
- Constructor is defined to initialize transaction class properties.
- there is no obvious limitation of these properties and methods

2. THE ALGORITHM USED IN A1.PHP

There are four sections of the whole algorithm in a1.php.

A. PROCESS ACCT.TXT

open file(acct.txt)

while loop(file)

read one line each time

spilt line

- create new instance of account class
- store account info into account instance
- store account instance into account array

close file

////////////////////////////////unit test////////////////////////////////

B.PROCESS TRANZ.TXT

- open file(tranz.txt)
- read and ignore first line

while loop(file)

- operating counters
- read one line each time
- split line
- create new instance of transaction class
- store transaction info into transaction instance
- store transaction instance into transaction array

close file

////////////////////////////////unit test////////////////////////////////

C.TRANSACTION OPERATION LOGIC

Foreach loop(transaction array)

- If (id of current transaction exists in account array)

- Operating accounts with current transaction

- If(invalid transaction)

- Store current transaction in invalid Transaction Array

- operating invalid counters

Else

Store current transaction in invalid Transaction Array

operating invalid counters

////////////////////////////////unit test////////////////////////////////

D.OUTPUT UPDATED ACCOUNT INFORMATION TO UPDATE.TXT

Open file (update.txt)

foreach loop(account array) write into update.txt

////////////////////////////////unit test////////////////////////////////

3. POSSIBLE SUGGESTIONS/CRITICISM/EXTENSIONS TO THE PROBLEM

There may exist some productive ways to improve the program and the problem:

- Create a file manager class to process all file related operations in terms of reading file, writing file. This action can separate file process and other processes, in order to achieve better objective-oriented environment.
- Use more safe way (functions) to get strings from file.
- Use file get contents function in each file process to be more effective.
- Use switch to replace may if and if else to be more effective.
- Use PHPUnit to unit test php files.
- May add a client-side transaction. User can type in account id, transaction type and amount.
- May contains front-end index web page for the atm. User can do transactions on a web page.
- May contain a hidden staff login page and a hidden admin login page.
- May contains a database for storing account details.
- May express more detailed invalid messages for invalid transactions.
- May contains a function that can print out the receipt.

- May contains authentication before user to do transactions(e.g. login function).