

Banking Application

Skills to be learned:

- use of simple linear linked list
- use of binary files
- write a complete application including a menu

Bank Transaction

A transaction is a money transfer between two accounts (debit → credit).

Data types

- date: day, month, year (3 ints)
- account : account number, opening date, owner name, balance,
- transaction: transaction number, transaction date, number of account to credit to, number of account to debit from, amount to be transferred

Data Representation

A simple linear linked list of each type

Part I

1. Define the 3 data types: date, account, transaction.
2. Write the functions: getDate(), putDate(), datecmp(), getAccount(), putAccount(), getTransaction(), putTransaction().
3. Write the function addAccount() that adds an account to the list of accounts. Add at head → easier. Account information is given as arguments. Similarly, write the function addTransaction().
4. Write the function displayAccounts() that displays the contents of the account linked list, an account per line. Similarly, write the function displayTransactions().
5. Write the function deleteAccounts() that deletes the account linked list elements from the heap. Similarly, write the function deleteTransactions().

6. Write a function `amountPerDay()` to calculate and return the total amount of all the transactions that occur on a given date passed as parameter. This function will deal with the linked list of transactions. Remember that transactions are sorted from most recent (head) to least recent (tail).
7. Write a function `errTrans()` to return whether a given transaction is erroneous or not. A transaction is erroneous if debit account number or creditor account number does not belong to accounts, or if no enough money exists the debit account.
8. Write the function `update()` to update the list of accounts according to a transaction, takes into parameter the transaction. The update is done by withdrawing the transaction amount of the debit account, and depositing this amount in the creditor account. Before updating you must call `errTrans` to make sure the update is possible. If not, the transaction node must not be added to linked list of transactions no update should be made.

File System

A binary file of accounts "accounts", contains information about accounts. A binary file of transactions "transactions", contains information transactions.

Part II

9. Write the function `saveAccounts()` that copies the list of accounts to the file "accounts". Similarly, write the function `saveTransactions()` that copies the list of transactions to the file "transactions".
10. write the function `loadAccounts()` that constructs the list of accounts out of the content of the file "accounts", use the function `addAccount()`, but push at tail using a local reference in order to keep the most recent opened document at head of list. (i.e. push at tail). Similarly, write the function `loadTransactions()`.

Part III

11. Write a main function with a menu to choose to perform any of the actions you wrote as functions, inspire from queue lab, it had a menu. Remember that everytime the user chooses to add a transaction, the function `update()` must also be called !
12. Try your banking system now!

The End