

ĐỀ KIỂM TRA KẾT THÚC HỌC PHẦN
Tên học phần: Cấu trúc dữ liệu và giải thuật
Thời gian làm bài: 90 phút
Mã đề: 003

Câu 1:

Viết chương trình sử dụng một trong các cấu trúc dữ liệu đã học trong học phần Cấu trúc dữ liệu và giải thuật để quản lý danh sách các tài khoản ngân hàng với thông tin của từng tài khoản như sau:

Thông tin tài khoản:

Số tài khoản: tối đa 16 ký tự, ví dụ 0111222233334444
Tên chủ tài khoản: ví dụ Nguyễn Văn An
Loại tài khoản: 001 hoặc 002 (001- Tài khoản Thanh Toán, 002-Tài khoản Tiết kiệm)
Số dư tài khoản: ví dụ 200 triệu VNĐ
Tình trạng tài khoản: 0 hoặc 1 (0 – Active, 1 – Block)

- Cho biết cấu trúc dữ liệu bạn chọn cho bài toán này là gì? Giải thích lý do.
- Viết hàm thực hiện tính năng nộp tiền vào một tài khoản ngân hàng do người dùng nhập vào từ bàn phím.

Các bước thực hiện như sau:

- Nhập vào số tài khoản.
- Kiểm tra tài khoản có tồn tại và có Active hay không, đồng thời phải là Tài khoản thanh toán. Nếu đúng các điều kiện trên sẽ cho phép tài khoản đó thực hiện tới bước tiếp theo.
- Người dùng nhập vào mật khẩu, nếu mật khẩu sai sẽ yêu cầu nhập lại. Nếu nhập sai quá 3 lần thì tình trạng của tài khoản trở thành trạng thái Block(giả sử mật khẩu set cố định là 123456).

- Người dùng thực hiện tạo tác nộ tiền(giả sử số tiền nộ được đọc từ một file txt). Nếu nộ tiền thành công cần cập nhật lại số dư tài khoản.
- c. Viết hàm thực hiện tính năng kiểm tra tài khoản Active/Block đồng thời có thể active lại tài khoản với các bước như sau:
 - Người dùng nhập vào 1 số tài khoản
 - Kiểm tra số tài khoản đó đang ở trạng thái Active hay Block. Xuất ra màn hình
 - Nếu tài khoản đang ở trạng thái Block thì sẽ hiển thị thêm câu hỏi có muốn active lại tài khoản này không? Nếu người dùng muốn active sẽ chọn Yes. Khi đó cần cập nhật trạng thái tài khoản đó thành Active.

Câu 2:

Cho dãy các số sau:

70, 18, 40, 20, X, -10, 10, 19, 35, 45, 60, 55

Với X là 2 chữ số cuối trong MSSV(nếu X trùng với 1 trong các số trong dãy thì +1 đến khi nào hết trùng).

- a. Vẽ cây nhị phân tìm kiếm(BST) với key của mỗi nút lần lượt là các số từ dãy trên.
Lưu ý: Ghi kết quả từng bước khi thêm từng nút.
- b. Vẽ cây nhị phân tìm kiếm cân bằng (AVL) với key của mỗi nút lần lượt là các số từ dãy trên. Sau đó thực hiện xóa nút X ra khỏi cây AVL vừa tạo. Lưu ý: ghi rõ từng bước cân bằng cây khi thêm/xóa nút(ví dụ: cây mất cân bằng loại nào, kết quả sau khi cân bằng).
- c. Viết chương trình cài đặt cây nhị phân tìm kiếm (BST) với các tính năng sau đây:
 - Nhập vào một số Y cần tìm.
 - Xác định giá trị Y có tồn tại trong cây hay không. Nếu có cho biết nút trong cây có giá trị Y đó là nút lá, nút có 2 nhánh con, nút có 1 nhánh con bên trái/phải, hay nút gốc. Nếu nó là nút có 2 nhánh con hãy xóa nó ra khỏi cây.

*****Lưu ý: Trước mỗi câu code sinh viên giải thích ý tưởng và các bước sẽ làm, viết code theo lập trình hướng đối tượng.**

-----Hết-----